

Tim Fingscheidt  
Hanno Gottschalk  
Sebastian Houben  
*Editors*

# Deep Neural Networks and Data for Automated Driving

Robustness, Uncertainty Quantification,  
and Insights Towards Safety

OPEN ACCESS



Springer

# Deep Neural Networks and Data for Automated Driving

Tim Fingscheidt · Hanno Gottschalk ·  
Sebastian Houben  
Editors

# Deep Neural Networks and Data for Automated Driving

Robustness, Uncertainty Quantification,  
and Insights Towards Safety



*Editors*

Tim Fingscheidt  
Institute for Communications Technology  
Technische Universität Braunschweig  
Braunschweig, Germany

Hanno Gottschalk  
Fachgruppe Mathematik und Informatik  
Bergische Universität Wuppertal  
Wuppertal, Germany

Sebastian Houben  
Schloss Birlinghoven  
Fraunhofer Institute for Intelligent Analysis  
and Information Systems IAIS  
Sankt Augustin, Germany



ISBN 978-3-031-01232-7

ISBN 978-3-031-01233-4 (eBook)

<https://doi.org/10.1007/978-3-031-01233-4>

© The Editor(s) (if applicable) and The Author(s) 2022. This book is an open access publication.

**Open Access** This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this book are included in the book's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Front page picture kindly provided by Jan-Aike Termöhlen and Andreas Bär, © Jan-Aike Termöhlen and Andreas Bär

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Foreword

I am thankful to the editors for sending me a prerelease copy of their book and for their kind invitation to write a foreword. I have greatly benefitted by reading the book and learning fresh ideas shared by distinguished experts from academia as well as professionals with serious engagement in the development of new generations of automobiles and driver assistance systems.

I have been fortunate to enjoy a long academic career pursuing a research agenda involving a number of topics discussed in the book. These include computer vision, intelligent vehicles and systems, machine learning, and human–robot interactions. For the past two decades, my team and collaborators have focused our explorations on developing human-centered, **safe** intelligent vehicles and transportation systems. Therefore, I am especially pleased to read this book presenting latest developments, findings, and insights in automated driving with *towards safety* as a highlighted term in the title and as a thread binding various chapters in the book. Emphasis on safety as an essential requirement was not always the case in the past. Reports of serious accidents involving automobiles with inadequate capabilities for automated driving are making safety an uncompromising requirement.

However, progress in the field does not occur at a consistent, predictable manner or rate. In the early phase, spanning around 1980–2000, autonomous vehicle research activities were mostly limited to a handful of universities and industrial labs. Important insights and milestones realized during this phase were novel perception and control systems for lane detection, cruise control, and carefully designed demonstrations of “autonomous” highway driving. However, interest and sponsorship of research in the field diminished and real-world deployment was unrealized.

In the next phase of autonomous vehicles research, spanning about mid-2000 to 2015, key enabling technologies, such as Global Positioning Systems and cost-effective mass production of sensors (radars, cameras) and embedded processors, not only invigorated serious activities in academic and technical communities, but also sparked interest and commitments by venture and commercial sectors.

It seemed that this time, our field was on an irreversible forward trajectory. Important insights and milestones realized during this phase include the introduction of active safety systems like dynamic stability control, and practical driver assistance

systems, such as lane keep assistance, collision avoidance, pedestrian protection, panoramic viewing, and other camera-based enhancements. We also started to take note of the great strides made in the machine learning and computer vision fields. Some of the major limitations encountered in the autonomous vehicles field, such as “brittle” object detectors trained using traditional, hand-crafted features, seemed resolvable with new developments in machine learning and computer vision.

Topics covered in this book are from the latest phase (from 2015 onwards) of autonomous vehicle research activities. Authors clearly acknowledge the important and essential roles of deep neural networks (DNN) and data-driven intelligent systems in making new contributions. They also recognize that there are unresolved challenges laying ahead. More focused research of both analytical and experimental nature is required. As a long-term proponent of *safe, human-centered* intelligent vehicles, I appreciate that the book places prominent emphasis on safety, integrated systems, robustness, performance evaluation, experimental metrics, benchmarks, and protocols. This makes the book very unique and an essential reading for students and scholars from multiple disciplines including engineering, computer/data/cognitive sciences, artificial intelligence, machine learning, and human/machine systems.

This is a very important book, as it deals with a topic surrounded with great excitement, not only in the technical communities, but also in the minds of people at large. With great excitement and wide popular interest in the autonomous driving field, often there is a temptation to overlook or ignore unresolved technical challenges. Given grave consequences of failures of complex systems deployed in the real world inhabited by humans, designers, engineers, and promoters of autonomous driving systems, we need to be most careful and attentive to the topics presented in the book. The editors and authors have done an outstanding job in presenting important themes and carefully selected topics in a systematic, insightful manner. The book identifies promising avenues for focused research and development, as well as acknowledges outstanding challenges yet to be resolved.

I congratulate Prof. Tim Fingscheidt, Prof. Hanno Gottschalk, and Prof. Sebastian Houben, for their vision and efforts in organizing an outstanding team of authors with deep academic and industrial backgrounds in developing this special book. I am confident that it will prove to be of great value in realizing a new generation of safer, smoother, trustworthy automobiles.



Mohan M. Trivedi

La Jolla, CA, USA

**Mohan M. Trivedi** Distinguished Professor of Engineering University of California, San Diego, founding director of the Computer Vision and Robotics Research (CVRR) and the Laboratory for Intelligent and Safe Automobiles (LISA). Fellow IEEE, SPIE, and IAPR.

# Preface

This book addresses readers from both academia and industry, since it is written by authors from both academia and industry. Accordingly, it takes on diverse viewing angles, but keeps a clear focus on machine-learned environment perception in autonomous vehicles. Special interest is on deep neural networks themselves, their robustness and uncertainty awareness, the data they are trained on, and, last but not least, on safety aspects.

The book is also special in its structure. While a first part is an extensive survey of literature in the field, the second part consists of 14 chapters, each detailing a particular aspect in the area of interest. This book wouldn't exist without the large-scale national flagship project KI Absicherung (safe AI for automated driving), with academia, car manufacturers, and suppliers being involved to work towards robustness and safety arguments in machine-learned environment perception for autonomous vehicles. I came up with the idea of a book project when I saw a first draft of a wonderful survey that was prepared by Sebastian Houben in collaboration with many partners from the project. My immediate thought was that this should not only be some project deliverable, it should become the core of a book (namely, Part I), appended with a number of novel contributions on specific topics in the field (namely, Part II). Beyond Sebastian Houben, I am happy that Hanno Gottschalk agreed to join the editorial team with his expertise in uncertainty and safety aspects. We thank KI Absicherung project leader (PL) Stephan Scholz (Volkswagen), Michael Mock (Scientific PL, Fraunhofer IAIS), and Fabian Hüger (Part PL, Volkswagen) very much, since without their immediate and ongoing support this book project wouldn't have come alive. We also thank the many reviewers for the time and energy they put into providing valuable feedback on the initial versions of the book chapters.

But the book reaches beyond the project. There are several contributions from outside the project enriching the diversity of views. Aiming at increasing impact, we editors decided to make the book an open-access publication. The funds for the open-access publication have been covered by the German Federal Ministry for Economic Affairs and Energy (BMWi) in the framework of the “Safe AI for Automated Driving” research consortium. Beyond that we are very grateful for the financial support both from Bergische Universität Wuppertal and Technische Universität Braunschweig.

A very special thanks goes to Andreas Bär, TU Braunschweig, for his endless efforts in running the editorial office, for any extra proofreading, formatting, and for keeping close contact to all the authors.

*Tim Fingscheidt, in the name of the editorial team, jointly with Hanno Gottschalk  
and Sebastian Houben*

Braunschweig, Germany  
Wuppertal, Germany  
Sankt Augustin, Germany  
November 2021

Tim Fingscheidt  
Hanno Gottschalk  
Sebastian Houben

# About This Book

An autonomous vehicle must be able to perceive its environment and react adequately to it. In highly automated vehicles, camera-based perception is increasingly performed by artificial intelligence (AI). One of the greatest challenges in integrating these technologies in highly automated vehicles is to ensure the functional safety of the system. Existing and established safety processes cannot simply be transferred to machine learning methods.

In the nationally funded project KI Absicherung, which is funded by the German Federal Ministry for Economic Affairs and Energy (BMWi), 24 partners from industry, academia, and research institutions collaborate to set up a stringent safety argumentation, with which AI-based function modules (AI modules) can be secured and validated for highly automated driving. The project brings together experts from the fields of artificial intelligence and machine learning, functional safety, and synthetic sensor data generation.

The topics and focus of this book have a large overlap with the field of research being tackled in subproject three of the KI Absicherung project, in which methods and measures are developed to identify and reduce inherent insufficiencies of the AI modules. We consider such methods and measures to be key elements in supporting the general safeguarding AI modules in a car.

We gratefully thank the editors of this book as well as Springer Verlag for making this research accessible to a broad audience and wish the reader interesting insights and new research ideas.

Stephan Scholz, Michael Mock, and Fabian Hüiger

# Introduction

Environment perception for driver assistance systems, highly automated driving, and autonomous driving has seen the same disruptive technology paradigm shift as many other disciplines: machine learning approaches and along with them deep neural networks (DNNs) have taken over and oftentimes replaced classical algorithms from signal processing, tracking, and control theory.

Along with this, many questions arise. Of fundamental interest is the question, how safe an artificial intelligence (AI) system actually is. Part I of this book provides an extensive literature survey introducing various relevant aspects to the reader. It not only provides guidance on how to inspect and understand deep neural networks, but also on how to overcome robustness issues or safety problems. Part I will also provide motivations and links to Part II of the book, which features novel approaches in various aspects of deep neural networks in the context of environment perception, such as their robustness and uncertainty awareness, the data they are trained on, and, last but not least, again, it will be about safety.

It all starts from choosing the right *data* from the right set of sensors, and, very important, the right amount of training data. Chapter “[Does Redundancy in AI Perception Systems Help to Test for Super-Human Automated Driving Performance?](#)” (p. 81) will provide insights into the problem of obtaining statistically sufficient test data and will discuss this under the assumption of redundant and independent (sensor) systems. The reader will learn about the curse of correlation in error occurrences of redundant sub-systems. A second aspect of dataset optimization is covered in Chapter “[Analysis and Comparison of Datasets by Leveraging Data Distributions in Latent Spaces](#)” (p. 107). Investigating variational autoencoder latent space data distributions, it will present a new technique to assess the dissimilarity of the domains of different datasets. Knowledge about this will help in mixing diverse datasets in DNN training, an important aspect to keep both training efficiency and generalization at a high level. When training or inferring on real data, outlier detection is important and potentially also to learn so far unknown objects. Chapter “[Detecting and Learning the Unknown in Semantic Segmentation](#)” (p. 277) presents a method to detect anomalous objects by statistically analyzing entropy responses in semantic segmentation to suggest new semantic categories. Such method can also be

used to pre-select diverse datasets for training. Chapter “[Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation](#)” (p. 127) follows the path of training with synthetic data, which has high potential, simply because rich ground truth is available for supervised learning even on sequence data. The authors are concerned with intelligent strategies of augmentation and use an appropriate metric to better match synthetic and real data, thereby increasing performance of synthetically trained networks.

A second focus area of this book is *robustness*. Robustness comprises the ability of a DNN to keep high performance also under adverse conditions such as varying weather and light conditions, dirt on the sensors, any kind of domain shift, or even under adversarial attacks. In Chapter “[Improved DNN Robustness by Multi-task Training with an Auxiliary Self-Supervised Task](#)” (p. 149), it is shown how intelligent multi-task training can be employed to strengthen a semantic segmentation DNN against various noise types and against adversarial attacks. An important aspect is that the supporting task of depth estimation does not require extra ground truth labels. Among the adversarial attacks, the universal ones are particularly dangerous as they can be applied to any current sensor input. Chapter “[Improving Transferability of Generated Universal Adversarial Perturbations for Image Classification and Segmentation](#)” (p. 171) addresses a way to generate these dangerous attacks both for semantic segmentation and for classification tasks, and should not be missed in any test suite for environment perception functions. Obtaining compact and efficient networks by compression techniques typically comes along with a degradation in performance. Not so in Chapter “[Joint Optimization for DNN Model Compression and Corruption Robustness](#)” (p. 405): The authors present a joint pruning and quantization framework which yields more robust networks against various corruptions. Mixture-of expert architectures for DNN aggregation can be a suitable means for improved robustness but also interpretability. Chapter “[Evaluating Mixture-of-Experts Architectures for Network Aggregation](#)” (p. 315) also shows that the models deal favorably with out-of-distribution (OoD) objects.

*Uncertainty* and interpretability are third focus areas of the book. Both concepts are fundamental towards self-awareness of environment perception for autonomous driving. In Chapter “[Invertible Neural Networks for Understanding Semantics of Invariances of CNN Representations](#)” (p. 197), invertible neural networks are deployed to recover invariant representations present in a trained network. They allow for a mapping to semantic concepts and thereby interpretation (and manipulation) of inner model representations. Confidence calibration, on the other hand, is an important means to obtain reliable confidences for further processing. Chapter “[Confidence Calibration for Object Detection and Segmentation](#)” (p. 225) proposes a multivariate calibration approach not only suitable for classification but also for semantic segmentation and object detection. Also aiming at improved object detection calibration, the authors of Chapter “[Uncertainty Quantification for Object Detection: Output- and Gradient-Based Approaches](#)” (p. 251) show that output-based and learning-gradient-based uncertainty metrics are uncorrelated. This allows advantageous combination of both paradigms leading to better overall detection accuracy and to better detection uncertainty estimates.

Part II started with safety aspects related to datasets (Chapter “[Does Redundancy in AI Perception Systems Help to Test for Super-Human Automated Driving Performance?](#)”, p. 81). In this fourth focus area, further *safety-related aspects* are taken up, including validation of networks. To validate environment perception functions, Chapter “[A Variational Deep Synthesis Approach for Perception Validation](#)” (p. 359) introduces a novel data generation framework for DNN validation purposes. The framework allows for effectively defining and testing common traffic scenes. The authors of Chapter “[The Good and the Bad: Using Neuron Coverage as a DNN Validation Technique](#)” (p. 383) present a useful metric to assess the training regimen and final quality of a DNN. Different forms of neuron coverage are discussed along with their roots in traditional verification and validation (V&V) methods. Finally, in Chapter “[Safety Assurance of Machine Learning for Perception Functions](#)” (p. 335), an assurance case is constructed and acceptance criteria for DNN models are proposed. Various use cases are covered.

Tim Fingscheidt  
Hanno Gottschalk  
Sebastian Houben

# Contents

## Part I Safe AI—An Overview

<b>Inspect, Understand, Overcome: A Survey of Practical Methods for AI Safety</b> .....	3
---	---

Sebastian Houben, Stephanie Abrecht, Maram Akila,  
Andreas Bär, Felix Brockherde, Patrick Feifel, Tim Fingscheidt,  
Sujan Sai Gannamaneni, Seyed Eghbal Ghobadi, Ahmed Hammam,  
Anselm Haselhoff, Felix Hauser, Christian Heinzeemann,  
Marco Hoffmann, Nikhil Kapoor, Falk Kappel, Marvin Klingner,  
Jan Kronenberger, Fabian Küppers, Jonas Löhdefink,  
Michael Mlynarski, Michael Mock, Firas Mualla,  
Svetlana Pavlitskaya, Maximilian Poretschkin, Alexander Pohl,  
Varun Ravi-Kumar, Julia Rosenzweig, Matthias Rottmann,  
Stefan Rüping, Timo Sämann, Jan David Schneider, Elena Schulz,  
Gesina Schwalbe, Joachim Sicking, Toshika Srivastava,  
Serin Varghese, Michael Weber, Sebastian Wirkert, Tim Wirtz,  
and Matthias Woehrle

## Part II Recent Advances in Safe AI for Automated Driving

<b>Does Redundancy in AI Perception Systems Help to Test for Super-Human Automated Driving Performance?</b> .....	81
---	----

Hanno Gottschalk, Matthias Rottmann, and Maida Saltagic

<b>Analysis and Comparison of Datasets by Leveraging Data Distributions in Latent Spaces</b> .....	107
--	-----

Hanno Stage, Lennart Ries, Jacob Langner, Stefan Otten, and Eric Sax

<b>Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation</b> .....	127
---	-----

Korbinian Hagn and Oliver Grau

<b>Improved DNN Robustness by Multi-task Training with an Auxiliary Self-Supervised Task .....</b>	149
Marvin Klingner and Tim Fingscheidt	
<b>Improving Transferability of Generated Universal Adversarial Perturbations for Image Classification and Segmentation .....</b>	171
Atiye Sadat Hasemi, Andreas Bär, Saeed Mozaffari, and Tim Fingscheidt	
<b>Invertible Neural Networks for Understanding Semantics of Invariances of CNN Representations .....</b>	197
Robin Rombach, Patrick Esser, Andreas Blattmann, and Björn Ommer	
<b>Confidence Calibration for Object Detection and Segmentation .....</b>	225
Fabian Küppers, Anselm Haselhoff, Jan Kronenberger, and Jonas Schneider	
<b>Uncertainty Quantification for Object Detection: Output- and Gradient-Based Approaches .....</b>	251
Tobias Riedlinger, Marius Schubert, Karsten Kahl, and Matthias Rottmann	
<b>Detecting and Learning the Unknown in Semantic Segmentation .....</b>	277
Robin Chan, Svenja Uhlemeyer, Matthias Rottmann, and Hanno Gottschalk	
<b>Evaluating Mixture-of-Experts Architectures for Network Aggregation .....</b>	315
Svetlana Pavlitskaya, Christian Hubschneider, and Michael Weber	
<b>Safety Assurance of Machine Learning for Perception Functions .....</b>	335
Simon Burton, Christian Hellert, Fabian Hüger, Michael Mock, and Andreas Rohatschek	
<b>A Variational Deep Synthesis Approach for Perception Validation .....</b>	359
Oliver Grau, Korbinian Hagn, and Qutub Syed Sha	
<b>The Good and the Bad: Using Neuron Coverage as a DNN Validation Technique .....</b>	383
Sujan Sai Gannamaneni, Maram Akila, Christian Heinzemann, and Matthias Woehrle	
<b>Joint Optimization for DNN Model Compression and Corruption Robustness .....</b>	405
Serin Varghese, Christoph Hügger, Andreas Bär, Fabian Hüger, and Tim Fingscheidt	

# About the Editors

**Tim Fingscheidt** Chair of Signal Processing and Machine Learning, Institute for Communications Technology, Technische Universität Braunschweig, Braunschweig, Germany

Tim Fingscheidt received the Dipl.-Ing. degree in Electrical Engineering in 1993 and the Ph.D. degree in 1998 from RWTH Aachen University, Germany, both with distinction. He joined AT&T Labs, Florham Park, NJ, USA, for a Postdoc in 1998, and Siemens AG, Information and Communication, Munich, Germany, in 1999, heading a team in digital signal processing. After a stay with Siemens Corporate Technology, Munich, Germany, from 2005 to 2006, he became Full Professor with the Institute for Communications Technology, Technische Universität (TU) Braunschweig, Germany, holding the Chair of “Signal Processing and Machine Learning”. His research interests are machine learning methods for robust environment perception, information fusion, signal enhancement, and coding, with applications both in speech technology and computer vision. He is founder of the TU Braunschweig Deep Learning Lab (TUBS.dll), a graduate student research think tank. Many of his projects have been dealing with automotive applications, such as autonomous driving. In recent years, he has been actively involved in the large-scale national research projects AI Validation, AI Delta Learning, and AI Data Tooling, contributing research in robust semantic segmentation, monocular depth estimation, domain adaptation, corner case detection, and learned image coding. He received numerous national and international awards for his publications, among these three CVPR Workshop Best Paper Awards in 2019, 2020, and 2021.

**Hanno Gottschalk** Professor for Stochastics at the University of Wuppertal, Germany

Hanno Gottschalk received diploma degrees of Physics and Mathematics from the Ruhr University of Bochum in 1995 and 1997, respectively. There he also received his Ph.D. in Mathematical Physics under the supervision of Sergio Albeverio in 1999. As a fellow of the German Academic Exchange Service (DAAD) he thereafter spent a Postdoc year at Università “La Sapienza” in Rome. Returning to the University of Bonn, he led a research project funded by the research council (DFG)

in 2000–2003, before completing his habilitation in 2003. Acquiring a C2 lecturer position at the University of Bonn in 2007, he left academia for a 3-year experience with Siemens Energy, where he held the position of a core competency owner in probabilistic design in the field of gas turbine engineering. Returning to academia in 2011, since then he holds a position as a Professor of Stochastics at the University of Wuppertal. In 2016, he became spokesperson of the Institute of Mathematical Modeling, Analysis and Computational Mathematics (IMACM). In 2018, he founded the Interdisciplinary Center of Machine Learning and Data Analytics (IZMD) at the University of Wuppertal as the spokesperson jointly with Anton Kummert and in 2020 he became the elected spokesperson of the institute. In the field of computer vision, his research is centered around the online detection and reduction of the errors of artificial intelligence.

**Sebastian Houben** Senior Data Scientist, Project Lead KI Absicherung, Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), St. Augustin, Germany

Sebastian Houben studied Mathematics and Computer Science in Aachen and Hagen working as a scientific staff member at the Institute for Bio and Nanosystems at the Helmholtz Center in Jülich, Germany. He received his Ph.D. degree in 2015 from the Ruhr University of Bochum, Germany, after which he joined the Autonomous Intelligent Systems Group at the University of Bonn as a Postdoctoral Researcher and was appointed Junior Professor at the University of Bochum in 2017. In early 2020, he took over the project lead KI Absicherung within the Fraunhofer Institute for Intelligent Analysis and Information Systems. He has a strong research interest in environment perception for autonomous robots, in particular, in automated vehicles, and the ramifications from today's state-of-the-art data-driven approaches. During his Ph.D. studies he has been involved in multiple pre-development projects for omnidirectional cameras deployed in advanced driver assistance systems. He participated in several flight robotics challenges, such as the European Robotics Challenge and the Mohamed Bin Zayed International Robotics Challenge, which he, as part of the University Bonn team NimbRo, won in 2017. As of October 2021, he has been appointed Full Professor with specialization in Autonomous Systems and Machine Learning at the University of Applied Sciences Bonn-Rhein-Sieg.

# **Safe AI—An Overview**

# Inspect, Understand, Overcome: A Survey of Practical Methods for AI Safety



**Sebastian Houben, Stephanie Abrecht, Maram Akila, Andreas Bär,  
Felix Brockherde, Patrick Feifel, Tim Fingscheidt, Sujan Sai Gannamaneni,  
Seyed Eghbal Ghobadi, Ahmed Hammam, Anselm Haselhoff, Felix Hauser,  
Christian Heinzeemann, Marco Hoffmann, Nikhil Kapoor, Falk Kappel,  
Marvin Klingner, Jan Kronenberger, Fabian Küppers, Jonas Löhdefink,  
Michael Mlynarski, Michael Mock, Firas Mualla, Svetlana Pavlitskaya,  
Maximilian Poretschkin, Alexander Pohl, Varun Ravi-Kumar,  
Julia Rosenzweig, Matthias Rottmann, Stefan Rüping, Timo Sämann,  
Jan David Schneider, Elena Schulz, Gesina Schwalbe, Joachim Sicking,  
Toshika Srivastava, Serin Varghese, Michael Weber, Sebastian Wirkert,  
Tim Wirtz, and Matthias Woehrle**

---

S. Houben (✉) · M. Akila · S. S. Gannamaneni · M. Mock · M. Poretschkin · J. Rosenzweig ·

S. Rüping · E. Schulz · J. Sicking · T. Wirtz

Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Sankt Augustin,  
Germany

e-mail: [sebastian.houben@h-brs.de](mailto:sebastian.houben@h-brs.de)

M. Akila

e-mail: [maram.akila@iais.fraunhofer.de](mailto:maram.akila@iais.fraunhofer.de)

S. S. Gannamaneni

e-mail: [sujan.sai.gannamaneni@iais.fraunhofer.de](mailto:sujan.sai.gannamaneni@iais.fraunhofer.de)

M. Mock

e-mail: [michael.mock@iais.fraunhofer.de](mailto:michael.mock@iais.fraunhofer.de)

M. Poretschkin

e-mail: [maximilian.poretschkin@iais.fraunhofer.de](mailto:maximilian.poretschkin@iais.fraunhofer.de)

J. Rosenzweig

e-mail: [julia.rosenzweig@iais.fraunhofer.de](mailto:julia.rosenzweig@iais.fraunhofer.de)

S. Rüping

e-mail: [stefan.rueping@iais.fraunhofer.de](mailto:stefan.rueping@iais.fraunhofer.de)

E. Schulz

e-mail: [elena.schulz@iais.fraunhofer.de](mailto:elena.schulz@iais.fraunhofer.de)

J. Sicking

e-mail: [joachim.sicking@iais.fraunhofer.de](mailto:joachim.sicking@iais.fraunhofer.de)

T. Wirtz

e-mail: [tim.wirtz@iais.fraunhofer.de](mailto:tim.wirtz@iais.fraunhofer.de)

S. Abrecht · C. Heinzeemann · M. Woehrle

Robert Bosch GmbH, Gerlingen, Germany

e-mail: [stephanie.abrecht@de.bosch.com](mailto:stephanie.abrecht@de.bosch.com)

**Abstract** Deployment of modern data-driven machine learning methods, most often realized by deep neural networks (DNNs), in safety-critical applications such as health care, industrial plant control, or autonomous driving is highly challenging due to numerous model-inherent shortcomings. These shortcomings are diverse and range from a lack of generalization over insufficient interpretability and implausible predictions to directed attacks by means of malicious inputs. Cyber-physical systems employing DNNs are therefore likely to suffer from so-called safety concerns, properties that preclude their deployment as no argument or experimental setup can help to assess the remaining risk. In recent years, an abundance of state-of-the-art techniques aiming to address these safety concerns has emerged. This chapter provides a structured and broad overview of them. We first identify categories of insufficiencies to then describe research activities aiming at their detection, quantification, or mitigation. Our work addresses machine learning experts and safety engineers alike: The former ones might profit from the broad range of machine learning topics covered and discussions on limitations of recent methods. The latter ones might gain insights into the specifics of modern machine learning methods. We hope that this contribution fuels discussions on desiderata for machine learning systems and strategies on how to help to advance existing approaches accordingly.

---

C. Heinzeemann

e-mail: [christian.heinzemann@de.bosch.com](mailto:christian.heinzemann@de.bosch.com)

M. Woehrle

e-mail: [matthias.woehrle@de.bosch.com](mailto:matthias.woehrle@de.bosch.com)

G. Schwalbe

Continental AG, Hanover, Germany

e-mail: [gesina.schwalbe@continental.com](mailto:gesina.schwalbe@continental.com)

V. Ravi-Kumar · T. Sämann

Valeo S.A., Paris, France

e-mail: [varun-ravi.kumar@valeo.com](mailto:varun-ravi.kumar@valeo.com)

T. Sämann

e-mail: [timo.saemann@valeo.com](mailto:timo.saemann@valeo.com)

M. Rottmann

Interdisciplinary Center for Machine Learning and Data Analytics, University of Wuppertal,  
Wuppertal, Germany

e-mail: [rottmann@uni-wuppertal.de](mailto:rottmann@uni-wuppertal.de)

S. Wirkert

Bayerische Motorenwerke AG, Munich, Germany

e-mail: [sebastian.wa.wirkert@bmwgroup.com](mailto:sebastian.wa.wirkert@bmwgroup.com)

N. Kapoor · J. D. Schneider · S. Varghese

Volkswagen AG, Wolfsburg, Germany

e-mail: [nikhil.kapoor@volkswagen.de](mailto:nikhil.kapoor@volkswagen.de)

J. D. Schneider

e-mail: [jan.david.schneider@volkswagen.de](mailto:jan.david.schneider@volkswagen.de)

S. Varghese

e-mail: [john.serin.varghese@volkswagen.de](mailto:john.serin.varghese@volkswagen.de)

## 1 Introduction

In barely a decade, deep neural networks (DNNs) have revolutionized the field of machine learning by reaching unprecedented, sometimes superhuman, performances on a growing variety of tasks. Many of these neural models have found their way into consumer applications like smart speakers, machine translation engines, or content feeds. However, in safety-critical systems, where human life might be at risk, the use of recent DNNs is challenging as various model-immanent insufficiencies are yet difficult to address.

This work summarizes the promising lines of research in how to identify, address, and at least partly mitigate these DNN insufficiencies. While some of the reviewed works are theoretically grounded and foster the overall understanding of training and predictive power of DNNs, others provide practical tools to adapt their development, training, or predictions. We refer to any such method as a *safety mechanism* if it addresses one or several safety concerns in a feasible manner. Their effectiveness in mitigating safety concerns is assessed by *safety metrics* [CNH+18, OOAG19, BGS+19, SS20a]. As most safety mechanisms target only a particular

P. Feifel · S. E. Ghobadi · A. Hammam  
 Opel Automobile GmbH, Rüsselsheim, Germany  
 e-mail: [patrick.feifel@opel-vauxhall.com](mailto:patrick.feifel@opel-vauxhall.com)

S. E. Ghobadi  
 e-mail: [seyed.eghbali.ghobadi@opel-vauxhall.com](mailto:seyed.eghbali.ghobadi@opel-vauxhall.com)

A. Hammam  
 e-mail: [ahmedmostafa.hammam@opel-vauxhall.com](mailto:ahmedmostafa.hammam@opel-vauxhall.com)

A. Haselhoff · J. Kronenberger · F. Küppers  
 Institute of Computer Science, Hochschule Ruhr West, Mülheim, Germany  
 e-mail: [anselm.haselhoff@hs-ruhrwest.de](mailto:anselm.haselhoff@hs-ruhrwest.de)

J. Kronenberger  
 e-mail: [jan.kronenberger@hs-ruhrwest.de](mailto:jan.kronenberger@hs-ruhrwest.de)

F. Küppers  
 e-mail: [fabian.kueppers@hs-ruhrwest.de](mailto:fabian.kueppers@hs-ruhrwest.de)

F. Brockherde  
 umlaut AG, Aachen, Germany  
 e-mail: [felix.brockherde@umlaut.com](mailto:felix.brockherde@umlaut.com)

F. Hauser  
 Karlsruhe Institute of Technology, Karlsruhe, Germany  
 e-mail: [felix.hauser@kit.edu](mailto:felix.hauser@kit.edu)

T. Srivastava  
 Audi AG, Ingolstadt, Germany  
 e-mail: [toshika.srivastava@audi.de](mailto:toshika.srivastava@audi.de)

F. Kappel · F. Mualla  
 ZF Friedrichshafen AG, Friedrichshafen, Germany  
 e-mail: [falk.kappel@zf.com](mailto:falk.kappel@zf.com)

insufficiency, we conclude that a *holistic safety argumentation* [BGS+19, SSH20, SS20a, WSRA20] for complex DNN-based systems will in many cases rely on a variety of safety mechanisms.

We structure our review of these mechanisms as follows: Sect. 2 focuses on *dataset optimization* for network training and evaluation. It is motivated by the well-known fact that, in comparison to humans, DNNs perform poorly on data that is structurally different from training data. Apart from insufficient generalization capabilities of these models, the data acquisition process and distributional data shifts over time play vital roles. We survey potential counter-measures, e.g., augmentation strategies and outlier detection techniques.

Mechanisms that improve on *robustness* are described in Sects. 3 and 4, respectively. They deserve attention as DNNs are generally not resilient to common perturbations and adversarial attacks.

Section 5 addresses incomprehensible network behavior and reviews mechanisms that aim at *explainability*, e.g., a more transparent functioning of DNNs. This is particularly important from a safety perspective as interpretability might allow for tracing back model failure cases thus facilitating purposeful improvements.

Moreover, DNNs tend to overestimate their prediction confidence, especially on unseen data. Straightforward ways to estimate prediction confidence yield mostly

F. Mualla  
e-mail: [firas.mualla@zf.com](mailto:firas.mualla@zf.com)

S. Pavlitskaya · M. Weber  
FZI Research Center for Information Technology, Karlsruhe, Germany  
e-mail: [pavlitskaya@fzi.de](mailto:pavlitskaya@fzi.de)

M. Weber  
e-mail: [weber@fzi.de](mailto:weber@fzi.de)

A. Bär · T. Fingscheidt · M. Klingner · J. Löhdefink  
Institute for Communications Technology (IfN), Technische Universität Braunschweig,  
Braunschweig, Germany  
e-mail: [andreas.baer@tu-bs.de](mailto:andreas.baer@tu-bs.de)

T. Fingscheidt  
e-mail: [t.fingscheidt@tu-bs.de](mailto:t.fingscheidt@tu-bs.de)

M. Klingner  
e-mail: [m.klingner@tu-bs.de](mailto:m.klingner@tu-bs.de)

J. Löhdefink  
e-mail: [j.loehdefink@tu-bs.de](mailto:j.loehdefink@tu-bs.de)

M. Hoffmann · M. Mlynarski · A. Pohl  
QualityMinds GmbH, München, Germany  
e-mail: [marco.hoffmann@qualityminds.de](mailto:marco.hoffmann@qualityminds.de)

M. Mlynarski  
e-mail: [michael.mlynarski@qualityminds.de](mailto:michael.mlynarski@qualityminds.de)

A. Pohl  
e-mail: [alexander.pohl@qualityminds.de](mailto:alexander.pohl@qualityminds.de)

unsatisfying results. Among others, this observation fueled research on more sophisticated *uncertainty estimations* (see Sect. 6), *redundancy mechanisms* (see Sect. 7), and attempts to reach *formal verification* as addressed in Sect. 8.

At last, many safety-critical applications require not only accurate but also near real-time decisions. This is covered by mechanisms on the DNN *architectural level* (see Sect. 9) and furthermore by *compression* and *quantization* methods (see Sect. 10).

We conclude this review of mechanism categories with an outlook on the steps to transfer a carefully arranged combination of safety mechanisms into an actual holistic safety argumentation.

## 2 Dataset Optimization

The performance of a trained model inherently relies on the nature of the underlying dataset. For instance, a dataset with poor variability will hardly result in a model ready for real-world applications. In order to approach the latter, data selection processes, such as corner case selection and active learning, are of utmost importance. These approaches can help to design datasets that contain the most important information, while preventing the so-much-desired information from getting lost in an ocean of data. For a given dataset and active learning setup, data augmentation techniques are very common aiming at extracting as much model performance out of the dataset as possible.

On the other hand, safety arguments also require the analysis of how a model behaves on out-of-distribution data, data that contains concepts the model has not encountered during training. This is quite likely to happen as our world is under constant change, in other words, exposed to a constantly growing domain shift. Therefore, these fields are lately gaining interest, also with respect to perception in automated driving.

In this section, we provide an overview over anomaly and outlier detection, active learning, domain shift, augmentation, and corner case detection. The highly relevant problem of obtaining statistically sufficient test data, even under the assumption of redundant and independent systems, will then be discussed in Chapter “[Does Redundancy in AI Perception Systems Help to Test for Super-Human Automated Driving Performance?](#)” [GRS22]. There it will be shown that neural networks trained on the same computer vision task show high correlation in error cases, even if training data and other design choices are kept independent. Using different sensor modalities, however, diminishes the problem to some extent.

## 2.1 Outlier/Anomaly Detection

The terms anomaly, outlier, and *out-of-distribution* (OoD) data detection are often used interchangeably in literature and refer to task of identifying data samples that are not representative of training data distribution. Uncertainty evaluation (cf. Sect. 6) is closely tied to this field as self-evaluation of models is one of the active areas of research for OoD detection. In particular, for image classification problems it has been reported that neural networks often produce high confidence predictions on OoD data [NYC15, HG17]. The detection of such OoD inputs can either be tackled by post-processing techniques that adjust the estimated confidence [LLS18, DT18] or by enforcing low confidence on OoD samples during training [HAB19, HMD19]. Even guarantees that neural networks produce low confidence predictions for OoD samples can be provided under specific assumptions (cf. [MH20b]). More precisely, this work utilizes Gaussian mixture models that, however, may suffer from high-dimensional data and require strong assumptions on the distribution parameters. Some approaches use generative models like *GANs* [SSW+17, AAAB18] and *autoencoders* [ZP17] for outlier detection. The models are trained to learn in-distribution data manifolds and will produce higher reconstruction loss for outliers.

For OoD detection in semantic segmentation, only a few works have been presented so far. Angus et al. [ACS19] present a comparative study of common OoD detection methods, which mostly deal with image-level classification. In addition, they provide a novel setup of relevant OoD datasets for this task. Another work trains a fully convolutional binary classifier that distinguishes image patches from a known set of classes from image patches stemming from an unknown class [BKOŠ18]. The classifier output applied at every pixel will give the per-pixel confidence value for an OoD object. Both of these works perform at pixel level and without any sophisticated feature generation methods specifically tailored for the detection of entire OoD instances. Up to now, outlier detection has not been studied extensively for object detection tasks. In [GBA+19], two CNNs are used to perform object detection and binary classification (benign or anomaly) in a sequential fashion, where the second CNN takes the localized object within the image as input. From a safety standpoint, detecting outliers or OoD samples is extremely important and beneficial as training data cannot realistically be large enough to capture all situations. Research in this area is heavily entwined with progress in uncertainty estimation (cf. Sect. 6) and domain adaptation (cf. Sect. 2.3). Extending research works to segmentation and object detection tasks would be particularly significant for leveraging automated driving research. In addition to safety, OoD detection can be beneficial in other aspects, e.g., when using local expert models. Here, an expert model for segmentation of urban driving scenes and another expert model for segmentation of highway driving scenes can be deployed in parallel, where an additional OoD detector could act as a trigger on which models can be switched. We extend this discussion in Chapter “[Detecting and Learning the Unknown in Semantic Segmentation](#)” [CURG22], where we investigate the handling of unknown objects in semantic segmentation. In the scope of semantic

segmentation, we detect anomalous objects via high-entropy responses and perform a statistical analysis over these detections to suggest new semantic categories.

With respect to the approaches presented above, uncertainty-based and generative-model-based OoD detection methods are currently promising directions of research. However, it remains an open question whether they can unfold their potential well on segmentation and object detection tasks.

## 2.2 Active Learning

It is widely known that, as a rule of thumb, for the training of any kind of artificial neural network, an increase of training data leads to increased performance. Obtaining labeled training data, however, is often very costly and time-consuming. *Active learning* provides one possible remedy to this problem: Instead of labeling every data point, active learning utilizes a *query strategy* to request labels from a teacher/an oracle which leverage the model performance most. The survey paper by Settles [Set10] provides a broad overview regarding query strategies for active learning methods. However, except for *uncertainty sampling* and *query by committee*, most of them seem to be infeasible in deep learning applications up to now. Hence, most of the research activities in active deep learning focus on these two query strategies, as we outline in the following.

It has been shown for image classification [GIG17, RKG18] that labels corresponding to uncertain samples can leverage the network’s performance significantly and that a combination with semi-supervised learning is promising. In both works, uncertainty of unlabeled samples is estimated via Monte Carlo (MC) dropout inference. MC dropout inference and a chosen number of training epochs are executed alternatingly, after performing MC dropout inference, the unlabeled samples’ uncertainties are assessed by means of sample-wise dispersion measures. Samples for which the DNN model is very uncertain about its prediction are presented to an oracle and labeled.

With respect to object detection, a moderate number of active learning methods has been introduced [KLSL18, RUN18, DCG+19, BKD19]. These approaches include uncertainty sampling [KLSL18, BKD19] and query-by-committee methods [RUN18]. In [KLSL18, DCG+19], additional algorithmic features specifically tailored for object detection networks are presented, i.e., separate treatment of the localization and classification loss [KLSL18], as well as weak and strong supervision schemes [DCG+19]. For semantic segmentation, an uncertainty-sampling-based approach has been presented [MLG+18], which queries polygon masks for image sections of a fixed size ( $128 \times 128$ ). Queries are performed by means of accumulated entropy in combination with a cost estimation for each candidate image section. Recently, new methods for estimating the quality of a prediction [DT18, RCH+20] as well as new uncertainty quantification approaches, e.g., gradient-based ones [ORG18], have been proposed. It remains an open question whether they are suitable for active learning. Since most of the conducted studies are rather of academic

nature, also their applicability to real-life data acquisition is not yet demonstrated sufficiently. In particular, it is not clear whether the proposed active learning schemes, including the label acquisition, for instance, in semantic segmentation, is suitable to be performed by human labelers. Therefore, labeling acquisition with a common understanding of the labelers' convenience and suitability for active learning are a promising direction for research and development.

### 2.3 Domains

The classical assumption in machine learning is that the training and testing datasets are drawn from the same distribution, implying that the model is deployed under the same conditions as it was trained under. However, as it is mentioned in [MTRA+12, JDCR12], for real-world applications this assumption is often violated in the sense that the training and the testing set stems from different domains having different distributions. This poses difficulties for statistical models and the performance will mostly degrade when they are deployed on a domain  $\mathcal{D}^{\text{test}}$ , having a different distribution than the training dataset (i.e., generalizing from the training to the testing domain is not possible). This makes the study of domains not only relevant from the machine learning perspective, but also from a safety point of view.

More formally, there are differing notions of a “domain” in literature. For [Csu17, MD18], a domain  $\mathcal{D} = \{\mathcal{X}, P(\mathbf{x})\}$  consists of a feature space  $\mathcal{X} \subset \mathbb{R}^d$  together with a marginal probability distribution  $P(\mathbf{x})$  with  $\mathbf{x} \in \mathcal{X}$ . In [BCK+07, BDBC+10], a domain is a pair consisting of a distribution over the inputs together with a labeling function. However, instead of a sharp labeling function, it is also widely accepted to define a (training) domain  $\mathcal{D}^{\text{train}} = \{(\mathbf{x}_i, \bar{y}_i)\}_{i=1}^n$  to consist of  $n$  (labeled) samples that are sampled from a joint distribution  $P(\mathbf{x}, \bar{y})$  (cf. [LCWJ18]).

The reasons for *distributional shift* are diverse—as are the names to indicate a shift. For example, if the rate of (class) images of interest is different between training and testing set this can lead to a domain gap and, result in differing overall error rates. Moreover, as Chen et al. [CLS+18] mention, changing weather conditions and camera setups in cars lead to a domain mismatch in applications of autonomous driving. In biomedical image analysis, different imaging protocols and diverse anatomical structures can hinder generalization of trained models (cf. [KBL+17, DCO+19]). Common terms to indicate distributional shift are *domain shift*, *dataset shift*, *covariate shift*, *concept drift*, *domain divergence*, *data fracture*, *changing environments*, or *dataset bias*. References [Sto08, MTRA+12] provide an overview. Methods and measures to overcome the problem of domain mismatch between one or more (cf. [ZZW+18]) source domains and target domain(s) and the resulting poor model performance are studied in the field of transfer learning and, in particular, its subtopic domain adaptation (cf. [MD18]). For instance, adapting a model that is trained on synthetically generated data to work on real data is one of the core challenges, as can be seen [CLS+18, LZG+19, VJB+19]. Furthermore, detecting when samples are out-of-domain or out-of-distribution is an active field of research (cf. [LLS18]).

and Sect. 2.1 as well as Sect. 8.2 for further reference). This is particularly relevant for machine learning models that operate in the real world: if an automated vehicle encounters some situation that deviates strongly from what was seen during training (e.g., due to some special event like a biking competition, carnival, etc.) this can lead to wrong predictions and thereby potential safety issues if not detected in time.

In Chapter “[Analysis and Comparison of Datasets by Leveraging Data Distributions in Latent Spaces](#)” [SRL+22], a new technique to automatically assess the discrepancy of the domains of different datasets is proposed, including domain shift within one target application. The aptitude of encodings generated by different machine-learned models on a variety of automotive datasets is considered. In particular, loss variants of the variational autoencoder that enforce disentangled latent space representations yield promising results in this respect.

## 2.4 Augmentation

Given the need for big amounts of data to train neural networks, one often runs into a situation where data is lacking. This can lead to insufficient generalization and an overfitting to the training data. An overview over different techniques to tackle this challenge can be found in [KGC17]. One approach to try and overcome this issue is the augmentation of data. It aims at optimizing available data and increasing its amount, curating a dataset that represents a wide variety of possible inputs during deployment. Augmentation can as well be of help when having to work with a heavily unbalanced dataset by creating more samples of underrepresented classes. A broad survey on data augmentation is provided by [SK19]. They distinguish between two general approaches to data augmentation with the first one being data warping augmentations that focus on taking existing data and transforming it in a way that does not affect labels. The other options are oversampling augmentations, which create synthetic data that can be used to increase the size of the dataset.

Examples of some of the most basic augmentations are flipping, cropping, rotating, translating, shearing, and zooming. These are affecting the geometric properties of the image and are easily implemented [SK19]. The machine learning toolkit Keras, for example, provides an easy way of applying them to data using their `ImageDataGenerator` class [C+15]. Other simple methods include adaptations in color space that affect properties such as lighting, contrast, and tints, which are common variations within image data. Filters can be used to control increased blur or sharpness [SK19]. In [ZZK+20], random erasing is introduced as a method with similar effect as cropping, aiming at gaining robustness against occlusions. An example for mixing images together as an augmentation technique can be found in [Ino18].

The abovementioned methods have in common that they work on the input data but there are different approaches that make use of deep learning for augmentation. An example for making augmentations in feature space using autoencoders can be found in [DT17]. They use the representation generated by the encoder and generate new samples by interpolation and extrapolation between existing samples of a

class. The lack of interpretability of augmentations in feature space in combination with the tendency to perform worse than augmentations in image space presents open challenges for those types of augmentations [WGSM17, SK19]. Adversarial training is another method that can be used for augmentation. The goal of adversarial training is to discover cases that would lead to wrong predictions. That means the augmented images won't necessarily represent samples that could occur during deployment but that can help in achieving more robust decision boundaries [SK19]. An example of such an approach can be found in [LCPB18]. Generative modeling can be used to generate synthetic samples that enlarge the dataset in a useful way with GANs, variational autoencoders and the combination of both are important tools in this area [SK19]. Examples for data augmentation in medical context using a CycleGAN [ZPIE17] can be found in [SYPS19] and using a progressively growing GAN [KALL18] in [BCG+18]. Next to neural style transfer [GEB15] that can be used to change the style of an image to a target style, AutoAugment [CZM+19] and population-based augmentation [HLS+19] are two more interesting publications. In both, the idea is to search a predefined search space of augmentations to gather the best selection.

The field of augmenting datasets with purely synthetic images, including related work, is addressed in Chapter “[Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation](#)” [HG22], where a novel approach to apply realistic sensor artifacts to given synthetic data is proposed. The better overall quality is demonstrated via established per-image metrics and a domain distance measure comparing entire datasets. Exploiting this metric as optimization criterion leads to an increase in performance for the DeeplabV3+ model as demonstrated on the Cityscapes dataset.

## 2.5 Corner Case Detection

Ensuring that AI-based applications behave correctly and predictably even in unexpected or rare situations is a major concern that gains importance especially in safety-critical applications such as autonomous driving. In the pursuit of more robust AI corner cases play an important role. The meaning of the term corner case varies in the literature. Some consider mere erroneous or incorrect behavior as corner cases [PCYJ19, TPJR18, ZHML20]. For example, in [BBLFs19] corner cases are referred to as situations in which an object detector fails to detect relevant objects at relevant locations. Others characterize corner cases mainly as rare combinations of input parameter values [KKB19, HDHH20]. This project adopts the first definition: inputs that result in unexpected or incorrect behavior of the AI function are defined as corner cases. Contingent on the hardware, the AI architecture and the training data, the search space of corner cases quickly becomes incomprehensibly large. While manual creation of corner cases (e.g., constructing or re-enacting scenarios) might be more controllable, approaches that scale better and allow for a broader and more systematic search for corner cases require extensive automation.

One approach to automatic corner case detection is based on transforming the input data. The *DeepTest* framework [TPJR18] uses three types of image transformations: linear, affine, and convolutional transformations. In addition to these transformations, metamorphic relations help detect undesirable behaviors of deep learning systems. They allow changing the input while asserting some characteristics of the result [XHM+11]. For example, changing the contrast of input frames should not affect the steering angle of a car [TPJR18]. Input–output pairs that violate those metamorphic relations can be considered as corner cases.

Among other things, the white-box testing framework *DeepXplore* [PCYJ19] applies a method called *gradient ascent* to find corner cases (cf. Sect. 8.1). In the experimental evaluation of the framework, three variants of deep learning architectures were used to classify the same input image. The input image was then changed according to the gradient ascent of an objective function that reflected the difference in the resulting class probabilities of the three model variants. When the changed (now artificial) input resulted in different class label predictions by the model variants, the input was considered as a corner case.

In [BBLFs19], corner cases are detected on video sequences by comparing predicted frames with actual frames. The detector has three components: the first component, semantic segmentation, is used to detect and locate objects in the input frame. As the second component, an image predictor trained on frame sequences predicts the actual frame based on the sequence preceding that frame. An error is determined by comparing the actual with the predicted (i.e., expected) frame, following the idea that only situations that are unexpected for AI-based perception functions may be potentially dangerous and therefore a corner case. Both the segmentation and the prediction error are then fed into the third component of the detector, which determines a corner case score that reflects the extent to which unexpected relevant objects are at relevant locations.

In [HDHH20], a corner case detector based on simulations in a CARLA environment [DRC+17] is presented. In the simulated world, AI agents control the vehicles. During simulations, state information of both the environment and the AI agents are fed into the corner case detector. While the environment provides the real vehicle states, the AI agents provide estimated and perceived state information. Both sources are then compared to detect conflicts (e.g., collisions). These conflicts are recorded for analysis. Several ways of automatically generating and detecting corner cases exist. However, corner case detection is a task with challenges of its own: depending on the operational design domain including its boundaries, the space of possible inputs can be very large. Also, some types of corner cases are specific to the AI architecture, e.g., the network type or the network layout used. Thus, corner case detection has to assume a holistic point of view on both model and input, adding further complexity and reducing transferability of previous insights.

Although it can be argued that rarity does not necessarily characterize corner cases, rare input data might have the potential of challenging the AI functionality (cf. Sect. 2.1). Another research direction could investigate whether structuring the input space in a way suitable for the AI functionality supports the detection of corner cases. Provided that the operational design domain is conceptualized as an ontol-

ogy, ontology-based testing [BMM18] may support automatic detection. A properly adapted generator may specifically select promising combinations of extreme parameter values and, thus, provide valuable input for synthetic test data generation.

### 3 Robust Training

Recent works [FF15, RSFD16, BRW18, AW19, HD19, ETTS19, BHFs19] have shown that state-of-the-art deep neural networks (DNNs) performing a wide variety of computer vision tasks, such as image classification [KSH12, HZRS15, MGR+18], object detection [Gir15, RDGF15, HGDG17], and semantic segmentation [CPSA17, ZSR+19, WSC+20, LBS+19], are not robust to small changes in the input.

Robustness of neural networks is an active and open research field that can be considered highly relevant for achieving safety in automated driving. Currently, most of the research is directed toward either improving adversarial robustness [SZS+14] (robustness against carefully designed perturbations that aim at causing misclassifications with high confidence) or improving corruption robustness [HD19] (robustness against commonly occurring augmentations such as weather changes, addition of Gaussian noise, photometric changes, etc.). While adversarial robustness might be more of a security issue than a safety issue, corruption robustness, on the other hand, can be considered highly safety-relevant.

Equipped with these definitions, we broadly term *robust training* here as methods or mechanisms that aim at improving either adversarial or corruption robustness of a DNN, by incorporating modifications into the architecture or into the training mechanism itself.

In this section, we cover three widespread techniques for fostering robustness during model training: hyperparameter optimization, modification of loss, and domain generalization. Additionally, in Chapter “[Improved DNN Robustness by Multi-Task Training With an Auxiliary Self-Supervised Task](#)” [KFs22], an approach for robustification via multi-task training is presented. Semantic segmentation is combined with the additional target of depth estimation and is proven to show increased robustness on the Cityscapes and KITTI datasets.

A useful metric to assess the training regimen and final quality of a neural network is presented in Chapter “[The Good and the Bad: Using Neuron Coverage as a DNN Validation Technique](#)” [GAHW22]. The use of different forms of neuron coverage is discussed and juxtaposed with pair-wise coverage on a tractable example that is being developed.

### 3.1 Hyperparameter Optimization

The final performance of a neural network highly depends on the learning process. The process includes the actual optimization and may additionally introduce training methods such as dropout, regularization, or parametrization of a multi-task loss.

These methods adapt their behavior for predefined parameters. Hence, their optimal configuration is *a priori* unknown. We refer to them as *hyperparameters*. Important hyperparameters comprise, for instance, the initial learning rate, steps for learning-rate reduction, learning-rate decay, momentum, batch size, dropout rate, and number of iterations. Their configuration has to be determined according to the architecture and task of the CNN [FH19]. The search of an optimal hyperparameter configuration is called hyperparameter optimization (HO).

HO is usually described as an optimization problem [FH19]. Thereby, the combined configuration space is defined as  $\Lambda = \lambda_1 \times \lambda_2 \times \dots \times \lambda_N$ , according to each domain  $\lambda_n$ . Their individual spaces can be continuous, discrete, categorical, or binary.

Hence, one aims to find an optimal hyperparameter configuration  $\lambda^*$  by minimizing an objective function  $\mathcal{O}(\cdot)$ , which evaluates a trained model  $\mathbf{F}$  having parameters  $\theta$  on the validation dataset  $\mathcal{D}^{\text{val}}$  with the loss  $J$ :

$$\lambda^* = \arg \min_{\lambda \in \Lambda} \mathcal{O}(J, \mathbf{F}, \theta, \mathcal{D}^{\text{train}}, \mathcal{D}^{\text{val}}). \quad (1)$$

This problem statement is widely regarded in traditional machine learning and primarily based on Bayesian optimization (BO) in combination with Gaussian processes. However, a straightforward application to deep neural networks encounters problems due to a *lack of scalability, flexibility, and robustness* [FKH18, ZCY+19]. To exploit the benefits of BO, many authors proposed different combinations with other approaches. Hyperband [LJD+18] in combination with BO (BOHB) [FKH18] frames the optimization as “[...] a pure exploration non-stochastic infinite-armed bandit problem [...]”. The method of BO for iterative learning (BOIL) [NSO20] iteratively internalizes collected information about the learning curve and the learning algorithm itself. The authors of [WTPFW19] introduce the trace-aware knowledge gradient (taKG) as an acquisition function for BO (BO-taKG) which “leverages both trace information and multiple fidelity controls”. Thereby BOIL and BO-taKG achieve state-of-the-art performance regarding CNNs outperforming hyperband.

Other approaches such as the orthogonal array tuning method (OATM) [ZCY+19] or HO by reinforcement learning (Hyp-RL) [JGST19] turn away from the Bayesian approaches and offer new research directions.

Finally, the insight that many authors include kernel sizes and number of kernels and layers in their hyperparameter configuration should be emphasized. More work should be spent on the distinct integration of HO in the performance estimation strategy of neural architecture search (cf. Sect. 9.3).

### 3.2 Modification of Loss

There exist many approaches that aim at directly modifying the loss function with an objective of improving either adversarial or corruption robustness [PDZ18, KS18, LYZZ19, XCKW19, SLC19, WSC+20, SR20]. One of the earliest approaches for improving corruption robustness was introduced by Zheng et al. [ZSLG16] and is called *stability training*, where they introduce a regularization term that penalizes the network prediction to a clean and an augmented image. However, their approach does not scale to many augmentations at the same time. Janocha et al. [JC17] then introduced a detailed analysis on the influence of multiple loss functions to model performance as well as robustness and suggested that expectation-based losses tend to work better with noisy data and squared-hinge losses tend to work better for clean data. Other well-known approaches are mainly based on variations of data augmentation [CZM+19, CZSL19, ZCG+19, LYP+19], which can be computationally quite expensive.

In contrast to corruption robustness, there exist many more approaches based on adversarial examples. We highlight some of the most interesting and relevant ones here. Mustafa et al. [CWL+19] propose to add a loss term that maximally separates class-wise feature map representations, hence increasing the distance from data points to the corresponding decision boundaries. Similarly, Pang et al. [PXD+20] proposed the Max-Mahalanobis center (MMC) loss to learn more structured representations and induce high-density regions in the feature space. Chen et al. [CBLR18] proposed a variation of the well-known cross-entropy (CE) loss that not only maximizes the model probabilities of the correct class, but, in addition, also minimizes model probabilities of incorrect classes. Cisse et al. [CBG+17] constraint the Lipschitz constant of different layers to be less than one which restricts the error propagation introduced by adversarial perturbations to a DNN. Dezfouli et al. [MDFUF19] proposed to minimize the curvature of the loss surface locally around data points. They emphasize that there exists a strong correlation between locally small curvature and correspondingly high adversarial robustness.

All of these methods highlighted above are evaluated mostly for image classification tasks on smaller datasets, namely, CIFAR-10 [Kri09], CIFAR-100 [Kri09], SVHN [NWC+11], and only sometimes on ImageNet [KSH12]. Very few approaches have been tested rigorously on complex safety-relevant tasks, such as *object detection*, *semantic segmentation*, etc. Moreover, methods that improve adversarial robustness are only tested on a small subset of attack types under differing attack specifications. This makes comparing multiple methods difficult.

In addition, methods that improve corruption robustness are evaluated over a standard dataset of various corruption types which may or may not be relevant to its application domain. In order to assess multiple methods for their effect on safety-related aspects, a thorough robustness evaluation methodology is needed, which is largely missing in the current literature. This evaluation would need to take into account relevant disturbances/corruption types present in the real world (application domain) and have to assess robustness toward such changes in a rigorous manner.

Without such an evaluation, we run into the risk of being overconfident in our network, thereby harming safety.

### 3.3 Domain Generalization

Domain generalization (DG) can be seen as an extreme case of *domain adaptation* (DA). The latter is a type of transfer learning, where the source and target tasks are the same (e.g., shared class labels) but the source and target domains are different (e.g., another image acquisition protocol or a different background) [Csu17, WYKN20]. The DA can be either supervised (SDA), where there is little available labeled data in the target domain, or unsupervised (UDA), where data in the target domain is not labeled. The DG goes one step further by assuming that the target domain is entirely unknown. Thus, it seeks to solve the train-test domain shift in general. While DA is already an established line of research in the machine learning community, DG is relatively new [MBS13], though with an extensive list of papers in the last few years.

Probably, the first intuitive solution that one may think of to implement DG is neutralizing the domain-specific features. It was shown in [WHLX19] that the gray-level co-occurrence matrices (GLCM) tend to perform poorly in semantic classification (e.g., digit recognition) but yield good accuracy in textural classification compared to other feature sets, such as speeded up robust features (SURF) and local binary patterns (LBP). DG was thus implemented by decorrelating the model’s decision from the GLCM features of the input image even without the need of domain labels.

Besides the aforementioned intensity-based statistics of an input image, it is known that characterizing image style can be done based on the correlations between the filter responses of a DNN layer [GEB16] (neural style transfer). In [SMK20], the training images are enriched with stylized versions, where a style is defined either by an external style (e.g., cartoon or art) or by an image from another domain. Here, DG is addressed as a *data augmentation* problem.

Some approaches [LTG+18, MH20a] try to learn generalizable latent representations by a kind of adversarial training. This is done by a generator or an encoder, which is trained to generate a hidden feature space that maximizes the error of a domain discriminator but at the same time minimizes the classification error of the task of concern. Another variant of adversarial training can be seen in [LPWK18], where an adversarial autoencoder [MSJ+16] is trained to generate features, which a discriminator cannot distinguish from random samples drawn from a prior Laplace distribution. This regularization prevents the hidden space from overfitting to the source domains, in a similar spirit to how variational autoencoders do not leave gaps in the latent space. In [MH20a], it is argued that the domain labels needed in such approaches are not always well-defined or easily available. Therefore, they assume unknown latent domains which are learned by clustering in a space similar to the style-transfer features mentioned above. The pseudo-labels resulting from clustering are then used in the adversarial training.

Autoencoders have been employed for DG not only in an adversarial setup, but also in the sense of *multi-task learning* nets [Car97], where the classification task in such nets is replaced by a reconstruction one. In [GKZB15], an autoencoder is trained to reconstruct not only the input image but also the corresponding images in the other domains.

In the core of both DA and DG, we are confronted with a distribution matching problem. However, estimating the probability density in high-dimensional spaces is intractable. The density-based metrics, such as Kullback–Leibler divergence, are thus not directly applicable. In statistics, the so-called *two-sample tests* are usually employed to measure the distance between two distributions in a point-wise manner, i.e., without density estimation. For deep learning applications, these metrics need not only to be point-wise but also differentiable. The two-sample tests were approached in the machine learning literature using (differentiable) K-NNs [DK17], classifier two-sample tests (C2ST) [LO17], or based on the theory of kernel methods [SGSS07]. More specifically, the *maximum mean discrepancy* (MMD) [GBR+06, GBR+12], which belongs to the kernel methods, is widely used for DA [GKZ14, LZWJ17, YDL+17, YLW+20] but also for DG [LPWK18]. Using the MMD, the distance between two samples is estimated based on pair-wise kernel evaluations, e.g., the radial basis function (RBF) kernel.

While the DG approaches generalize to domains from which zero shots are available, the so-called *zero-shot learning* (ZSL) approaches generalize to tasks (e.g., new classes in the same source domains) for which zero shots are available. Typically, the input in ZSL is mapped to a semantic vector per class instead of a simple class label. This can be, for instance, a vector of visual attributes [LNH14] or a word embedding of the class name [KXG17]. A task (with zero shots at training time) can be then described by a vector in this space. In [MARC20], there is an attempt to combine ZSL and DG in the same framework in order to generalize to new domains as well as new tasks, which is also referred to as *heterogeneous domain generalization*.

Note that most discussed approaches for DG require non-standard handling, i.e., modifications to models, data, and/or the optimization procedure. This issue poses a serious challenge as it limits the practical applicability of these approaches. There is a line of research which tries to address this point by linking DG to other machine learning paradigms, especially the model-agnostic meta-learning (MAML) [FAL17] algorithm, in an attempt to apply DG in a model-agnostic way. Loosely speaking, a model can be exposed to simulated train-test domain shift by training on a small *support set* to minimize the classification error on a small *validation set*. This can be seen as an instance of a *few-shot learning* (FSL) problem [WYKN20]. Moreover, the procedure can be repeated on other (but related) FSL tasks (e.g., different classes) in what is known as *episodic training*. The model transfers its knowledge from one task to another task and learns how to learn fast for new tasks. Thus, this can be seen as a *meta-learning* objective [HAMS20] (in a FSL setup). Since the goal of DG is to adapt to new domains rather than new tasks, several model-agnostic approaches [LYSH18, BSC18, LZY+19, DdCKG19] try to recast this procedure in a DG setup.

## 4 Adversarial Attacks

Over the last few years, deep neural networks (DNNs) consistently showed state-of-the-art performance across several vision-related tasks. While their superior performance on clean data is indisputable, they show a lack of robustness to certain input patterns, denoted as *adversarial examples* [SZS+14]. In general, an algorithm for creating adversarial examples is referred to as an *adversarial attack* and aims at fooling an underlying DNN, such that the output changes in a desired and malicious way. This can be carried out without any knowledge about the DNN to be attacked (black-box attack) [MDFF16, PMG+17], or with full knowledge about the parameters, architecture, or even training data of the respective DNN (white-box attack) [GSS15, CW17a, MMS+18]. While initially being applied on simple classification tasks, some approaches aim at finding more realistic attacks [TVRG19, JLS+20], which particularly pose a threat to safety-critical applications, such as DNN-based environment perception systems in autonomous vehicles. Altogether, this motivated the research in finding ways of defending against such adversarial attacks [GSS15, GRCvdM18, MDFUF19, XZZ+19]. In this section, we introduce the current state of research regarding adversarial attacks in general, more realistic adversarial attacks closely related to the task of environment perception for autonomous driving, and strategies for detecting or defending adversarial attacks. We conclude each section by clarifying current challenges and research directions.

### 4.1 Adversarial Attacks and Defenses

The term *adversarial example* was first introduced by Szegedy et al. [SZS+14]. From there on, many researchers tried to find new ways of crafting adversarial examples more effectively. Here, the fast gradient sign method (FGSM) [GSS15], DeepFool [MDFF16], least-likely class method (LLCM) [KGB17a, KGB17b], C&W [CW17b], momentum iterative fast gradient sign method (MI-FGSM) [DLP+18], and projected gradient descent (PGD) [MMS+18] are a few of the most famous attacks so far. In general, these attacks can be executed in an iterative fashion, where the underlying adversarial perturbation is usually bounded by some norm and is following additional optimization criteria, e.g., minimizing the number of changed pixels.

The mentioned attacks can be further categorized as image-specific attacks, where for each image a new perturbation needs to be computed. On the other hand, image-agnostic attacks aim at finding a perturbation, which is able to fool an underlying DNN on a set of images. Such a perturbation is also referred to as a *universal adversarial perturbation* (UAP). Here, the respective algorithm UAP [MDFFF17], fast feature fool (FFF) [MGB17], and prior driven uncertainty approximation (PD-UA) [LJL+19] are a few honorable mentions. Although the creation process of a universal adversarial perturbation typically relies on a white-box setting, they show a high

*transferability* across models [HAMFs22]. This allows for black-box attacks, where one model is used to create a universal adversarial perturbation, and another model is being attacked with the beforehand-created perturbation. Universal adversarial perturbations are investigated in more detail in Chapter “[Improving Transferability of Generated Universal Adversarial Perturbations for Image Classification and Segmentation](#)” [HAMFs22], where a way to construct these perturbations on the task of image classification and segmentation is presented. In detail, the low-level feature extraction layer is attacked via an additional loss term to generate universal adversarial perturbations more effectively. Another way of designing black-box attacks is to create a surrogate DNN, which mimics the respective DNN to be attacked and thus can be used in the process of adversarial example creation [PMG+17]. On the contrary, some research has been done to create completely incoherent images (based on evolutionary algorithms or gradient ascent) to fool an underlying DNN [NYC15]. Different from that, another line of work has been proposed to alter only some pixels in images to attack a respective model. Here [NK17] has used optimization approaches to perturb some pixels in images to produce targeted attacks, aiming at a specific class output, or non-targeted attacks, aiming at outputting a class different from the network output or the ground truth. This can be extended up to finding one pixel in the image to be exclusively perturbed to generate adversarial images [NK17, SVS19]. The authors of [BF17, SBMC17, PKGB18] proposed to train generative models to generate adversarial examples. Given an input image and the target label, a generative model is trained to produce adversarial examples for DNNs. However, while the produced adversarial examples look rather unrealistic to a human, they are able to completely deceive a DNN.

The existence of adversarial examples not only motivated research in finding new attacks, but also in finding *defense strategies* to effectively defend these attacks. Especially for safety-critical applications, such as DNN-based environment perception for autonomous driving, the existence of adversarial examples needs to be handled accordingly. Similar to adversarial attacks, one can categorize defense strategies into two types: *model-specific* defense strategies and *model-agnostic* defense strategies. The former refers to defense strategies, where the model of interest is modified in certain ways. The modification can be done on the architecture, training procedure, training data, or model weights. On the other hand, model-agnostic defense strategies consider the model to be a black box. Here, only the input or the output is accessible. Some well-known model-specific defense strategies include adversarial training [GSS15, MMS+18], the inclusion of robustness-oriented loss functions during training [CLC+19, MDFUF19, KYL+20], removing adversarial patterns in features by denoising layers [HRF19, MKH+19, XWvdM+19], and redundant teacher-student frameworks [BHSFs19, BKV+20]. Majority of model-agnostic defense strategies primarily focus on various kinds of (gradient masking) pre-processing strategies [GRCvdM18, BFW+19, GR19, JWCF19, LLL+19, RSFM19, TCBZ19]. The idea is to remove the adversary from the respective image, such that the image is transformed from the adversarial space back into the clean space.

Nonetheless, Athalye et al. [ACW18] showed that gradient masking alone is not a sufficient criterion for a reliable defense strategy. In addition, detection and out-of-

distribution techniques have also been proposed as model-agnostic defense strategies against adversarial attacks. Here, the Mahalanobis distance [LLLS18] or area under the receiver operating characteristic curve (AUROC) and area under the precision-recall curve (AUPR) [HG17] are used to detect adversarial examples. The authors of [HG17, LLLS17, MCKBF17], on the other hand, proposed to train networks to detect, whether the input image is out-of-distribution or not.

Moreover, Feinman et al. [FCSG17] proved that adversarial attacks usually produce high uncertainty on the output of the DNN. As a consequence, they proposed to use the dropout technique to estimate uncertainty on the output to identify a possible adversarial attack. Regarding adversarial attacks, majority of the listed attacks are designed for image classification. Only a few adversarial attacks consider tasks that are closely related to autonomous driving, such as bounding box detection, semantic segmentation, instance segmentation, or even panoptic segmentation. Also, the majority of the adversarial attacks rely on a white-box setting, which is usually not present for a potential attacker. Especially universal adversarial perturbations have to be considered as a real threat due to their high model transferability. Generally speaking, the existence of adversarial examples has not been thoroughly studied yet. An analytical interpretation is still missing, but could help in designing more mature defense strategies.

Regarding defense strategies, adversarial training is still considered as one of the most effective ways of increasing the robustness of a DNN. Nonetheless, while adversarial training is indeed effective, it is rather inefficient in terms of training time. In addition, model-agnostic defenses should be favored as once being designed, they can be easily transferred to different models. Moreover, as most model-agnostic defense strategies rely on gradient-masking and it has been shown that gradient-masking is not a sufficient property for a defense strategy, new ways of designing model-agnostic defenses should be taken into account. Furthermore, out-of-distribution and adversarial attacks detection or even correction methods have been a new trend for identifying attacks. However, as the environment perception system of an autonomous driving vehicle could rely on various information sources, including LiDAR, optical flow, or depth from a stereo camera, techniques of information fusion should be further investigated to mitigate or even eliminate the effect of adversarial examples.

## 4.2 More Realistic Attacks

We consider the following two categories of realistic adversarial attacks: (1) image-level attacks, which not only fool a neural network but also pose a provable threat to autonomous vehicles and (2) attacks which have been applied in a real world or in a simulation environment, such as car learning to act (CARLA) [DRC+17].

Some notable examples in the first category of attacks include attacks on semantic segmentation [MCKBF17] or person detection [TVRG19].

In the second group of approaches, the attacks are specifically designed to survive real-world distortions, including different distances, weather and lighting conditions,

as well as camera angles. For this, adversarial perturbations are usually concentrated in a specific image area, called *adversarial patch*. Crafting an adversarial patch involves specifying a patch region in each training image, applying transformations to the patch, and iteratively changing the pixel values within this region to maximize the network prediction error. The latter step typically relies on an algorithm, proposed for standard adversarial attacks, which aim at crafting invisible perturbations while misleading neural networks, e.g., C&W [CW17b], Jacobian-based saliency map attack (JSMA) [PMG+17], and PGD [MMS+18].

The first printable adversarial patch for image classification was described by Brown et al. [BMR+17]. Expectation over transformations (EOT) [AEIK18] is one of the influential updates to the original algorithm—it permits to robustify patch-based attacks to distortions and affine transformations. Localized and visible adversarial noise (LaVAN) [KZG18] is a further method to generate much smaller patches (up to 2% of the pixels in the image). In general, fooling image classification with a patch is a comparatively simple task, because adversarial noise can mimic an instance of another class and thus lower the prediction probability for a true class.

Recently, patch-based attacks for a more tricky task of object detection have been described [LYL+19, TVRG19]. Also, Lee and Kolter [LK19] generate a patch using PGD [MMS+18], followed by EOT applied to the patch. With this approach, all detections in an image can be successfully suppressed, even without any overlap of a patch with bounding boxes. Furthermore, several approaches for generating an adversarial T-shirt have been proposed, including [XZL+20, WLDG20].

DeepBillboard [ZLZ+20] is the first attempt to attack end-to-end driving models with adversarial patches. The authors propose to generate a single patch for a sequence of input images to mislead four steering models, including DAVE-2 in a drive-by scenario.

Apart from physical feasibility, inconspicuousness is crucial for a realistic attack. Whereas adversarial patches usually look like regions of noise, several works have explored attacks with an inconspicuous patch. In particular, Eykholt et al. [EEF+18] demonstrate the vulnerability of road sign classification to the adversarial perturbations in the form of only black and white stickers. In [BHG+19], an end-to-end driving model is attacked in CARLA by painting of black lines on the road. Also, Kong and Liu [KGLL20] use a generative adversarial network to get a realistic billboard to attack an end-to-end driving model in a drive-by scenario. In [DMW+20], a method to hide visible adversarial perturbations with customized styles is proposed, which leads to adversarial traffic signs that look unsuspicious to a human. Current research mostly focuses on attacking image-based perception of an autonomous vehicle. Adversarial vulnerability of further components of an autonomous vehicle, e.g., LiDAR-based perception, optical flow, and depth estimation, has only recently gained attention. Furthermore, most attacks consider only a single component of an autonomous driving pipeline, the question whether the existing attacks are able to propagate to further pipeline stages has not been studied yet. The first work in this direction [JLS+20] describes an attack on object detection and tracking. The evaluation is, however, limited to a few clips, where no experiments in the real world

have been performed. Overall, the research on realistic adversarial attacks, especially combined with physical tests, is currently in the starting phase.

## 5 Interpretability

Neural networks are, by their nature, black boxes and therefore intrinsically hard to interpret [Tay06]. Due to their unrivaled performance, they still remain first choice for advanced systems even in many safety-critical areas, such as level 4 automated driving. This is why the research community has invested considerable effort to unhinge the black-box character and make deep neural networks more transparent.

We can observe three strategies that provide different viewpoints toward this goal in the state of the art. First is the most direct approach of opening up the black box and looking at intermediate representations. Being able to interpret individual layers of the system facilitates interpretation of the whole. The second approach tries to provide interpretability by explaining the network's decisions with pixel attributions. Aggregated explanations of decisions can then lead to interpretability of the system itself. Third is the idea of approximating the network with interpretable proxies to benefit from the deep neural networks performance while allowing interpretation via surrogate models. Underlying all aspects here is the area of visual analytics.

There exists earlier research in the medical domain to help human experts understand and convince them of machine learning decisions [CLG+15]. Legal requirements in the finance industry gave rise to interpretable systems that can justify their decisions. An additional driver for interpretability research was the concern for Clever Hans predictors [LWB+19].

### 5.1 Visual Analytics

Traditional data science has developed a huge tool set of automated analysis processes conducted by computers, which are applied to problems that are well defined in the sense that the dimensionality of input and output as well as the size of the dataset they rely on is manageable. For those problems that in comparison are more complex, the automation of the analysis process is limited and/or might not lead to the desired outcome. This is especially the case with unstructured data like image or video data in which the underlying information cannot directly be expressed by numbers. Rather, it needs to be transformed to some structured form to enable computers to perform some task of analysis. Additionally, with an ever-increasing amount of various types of data being collected, this “information overload” cannot solely be analyzed by automatic methods [KAF+08, KMT09].

*Visual analytics* addresses this challenge as “the science of analytical reasoning facilitated by interactive visual interfaces” [TC05]. Visual analytics therefore does not only focus on either computationally processing data or visualizing results but

coupling both tightly with interactive techniques. Thus, it enables an integration of the human expert into the iterative visual analytics process: through visual understanding and human reasoning, the knowledge of the human expert can be incorporated to effectively refine the analysis. This is of particular importance, where a stringent safety argumentation for complex models is required. With the help of visual analytics, the line of argumentation can be built upon arguments that are understandable for humans. To include the human analyst efficiently into this process, a possible guideline is the *visual analytics mantra* by Keim: “Analyze first, show the important, zoom, filter and analyze further, details on demand” [KAF+08].<sup>1</sup>

The core concepts of visual analytics therefore rely on well-designed interactive visualizations, which support the analyst in the tasks of, e.g., reviewing, understanding, comparing, and inferring not only the initial phenomenon or data but also the computational model and its results itself with the goal of enhancing the analytical process. Driven by various fields of application, visual analytics is a multidisciplinary field with a wide variety of task-oriented development and research. Recent work has been done in several areas: depending on the task, there exist different pipeline approaches to create whole *visual analytics systems* [WZM+16]; the injection of human expert knowledge into the process of determining trends and patterns from data is the focus of *predictive visual analytics* [LCM+17, LGH+17]; enabling the human to explore *high-dimensional data* [LMW+17] interactively and visually (e.g., via dimensionality reduction [SZS+17]) is a major technique to enhance the understandability of complex models (e.g., neural networks); the iterative improvement and the understanding of machine learning models is addressed by using interactive visualizations in the field of *general machine learning* [LWLZ17], or the other way round: using machine learning to improve visualizations and guidance based on user interactions [ERT+17]. Even more focused on the loop of simultaneously developing and refining machine learning models is the area of *interactive machine learning*, where the topics of *interface design* [DK18] and the *importance of users* [ACKK14, SSZ+17] are discussed. One of the current research directions is using visual analytics in the area of *deep learning* [GTC+18, HKPC18, CL18]. However, due to the interdisciplinarity of visual analytics, there are still open directions and ongoing research opportunities.

Especially in the domain of neural networks and deep learning, visual analytics is a relatively new approach in tackling the challenge of *explainability* and *interpretability* of those often called *black boxes*. To enable the human to better interact with the models, research is done in enhancing the *understandability* of complex deep learning models and their outputs with the use of proper visualizations. Other research directions attempt to achieve improving the *trustability* of the models, giving the opportunity to inspect, diagnose, and refine the model. Further, possible areas for research are *online training processes* and the development of *interactive systems* covering the whole process of training, enhancing, and monitoring machine learning models. Here, the approach of *mixed guidance*, where system-initiated guidance is

---

<sup>1</sup> Extending the original *visualization mantra* by Shneiderman “Overview first, filter and zoom, details on demand” [Shn96].

combined with user-initiated guidance, is discussed among the visual analytics community as well. Another challenge and open question is creating ways of *comparing models* to examine which model yields a better performance, given specific situations and selecting or combining the best models with the goal of increasing performance and overall safety.

## 5.2 Intermediate Representations

In general, representation learning [BCV13] aims to extract lower dimensional features in latent space from higher dimensional inputs. These features are then used as an effective representation for regression, classification, object detection, and other machine learning tasks. Preferably, latent features should be disentangled, meaning that they represent separate factors found in the data that are statistically independent. Due to their importance in machine learning, finding meaningful intermediate representations has long been a primary research goal. Disentangled representations can be interpreted more easily by humans and can, for example, be used to explain the reasoning of neural networks [HDR18].

Among the longer known methods for extracting disentangled representations are principal component analysis (PCA) [FP78, JC16], independent component analysis [HO00], and nonnegative matrix factorization [BBL+07]. PCA is highly sensitive to outliers and noise in the data. Therefore, more robust algorithms were proposed. In [SBS12] already a small neural network was used as an encoder and the algorithm proposed in [FXY12] can deal with high-dimensional data. Some robust PCA algorithms are provided with analytical performance guarantees [XCS10, RA17, RL19].

A popular method for representation learning with deep networks is the variational autoencoder (VAE) [KW14]. An important generalization of the method is the  $\beta$ -VAE variant [HMP+17], which improved the disentanglement capability [FAA18]. Later analysis added to the theoretical understanding of  $\beta$ -VAE [BHP+18, SZYP19, KP20]. Compared to standard autoencoders, VAEs map inputs to a distribution, instead of mapping them to a fixed vector. This allows for additional regularization of the training to avoid overfitting and ensure good representations. In  $\beta$ -VAEs, the trade-off between reconstruction quality and disentanglement can be fine-tuned by the hyperparameter  $\beta$ .

Different regularization schemes have been suggested to improve the VAE method. Among them are Wasserstein autoencoders [TBGS19, XW19], attribute regularization [PL20], and relational regularization [XLH+20]. Recently, a connection between VAEs and nonlinear independent component analysis was established [KKMH20] and then expanded [SRK20].

Besides VAEs, deep generative adversarial networks can be used to construct latent features [SLY15, CDH+16, MSJ+16]. Other works suggest centroid encoders [GK20] or conditional learning of Gaussian distributions [SYZ+21] as alternatives to VAEs. In [KWG+18], concept activation vectors are defined as being orthogonal

to the decision boundary of a classifier. Apart from deep learning, entirely new architectures, such as capsule networks [SFH17], might be used to disassemble inputs.

While many different approaches for disentangling exist, the feasibility of the task is not clear yet and a better theoretical understanding is needed. The disentangling performance is hard to quantify, which is only feasible with information about the latent ground truth [EW18]. Models that overly rely on single directions, single neurons in fully connected networks, or single feature maps in CNNs have the tendency to overfit [MBRB18]. According to [LBL+19], unsupervised learning does not produce good disentangling and even small latent spaces do not reduce the sample complexity for simple tasks. This is in direct contrast to newer findings that show a decreased sample complexity for more complex visual downstream tasks [vLSB20]. So far, it is unclear if disentangling improves the performance of machine learning tasks.

In order to be interpretable, latent disentangled representations need to be aligned with human understandable concepts. In [EIS+19], training with adversarial examples was used and the learned representations were shown to be more aligned with human perception. For explainable AI, disentangling alone might not be enough to generate interpretable output, and additional regularization could be needed.

In Chapter “[Invertible Neural Networks for Understanding Semantics of Invariances of CNN Representations](#)” [REBO22], a new approach is presented, which aims at extracting invariant representations that are present in a trained network. To this end, invertible neural networks are deployed to recover these invariances and to map them to accessible semantic concepts. These concepts allow for both interpretation of an inner model representation and means to carefully manipulate them and examine the effect on the prediction.

### 5.3 Pixel Attribution

The non-linearity and complexity of DNNs allow them to solve perception problems, such as detecting a pedestrian, that cannot be specified in detail. At the same time, the automatic extraction of features given in an input image and the mapping to the respective prediction is counterintuitive and incomprehensible for humans, which makes it hard to argue safety for a neural-network-based perception task. Feature importance techniques are currently predominantly used to diagnose the causes of incorrect model behaviors [BXS+20]. So-called *attribution maps* are a visual technique to express the relationship between relevant pixels in the input image and the network’s prediction. Regions in an image that contain relevant features are highlighted accordingly. Attribution approaches mostly map to one of three categories.

Gradient-based and activation-based approaches (such as [SVZ14, SDBR14, BBM+15, MLB+17, SCD+20, STK+17, SGK19] among others) rely on the gradient of the prediction with respect to the input. Regions that were most relevant for the prediction are highlighted. Activation-based approaches relate the feature maps of the last convolutional layer to output classes.

Perturbation-based approaches [ZF14, FV17, ZCAW17, HMK+19] suggest manipulating the input. If the prediction changes significantly, the input may hold a possible explanation at least.

While gradient-based approaches are oftentimes faster in computation, perturbation-based approaches are much easier to interpret. As many studies have shown [STY17, AGM+20], there is still a lot of research to be done before attribution methods are able to robustly provide explanations for model predictions, in particular, erroneous behavior. One key difficulty is the lack of an agreed-upon definition of a good attribution map including important properties. Even between humans, it is hard to agree on what a good explanation is due to its subjective nature. This lack of ground truth makes it hard or even impossible to quantitatively evaluate an explanation method. Instead, this evaluation is done only implicitly. One typical way to do this is the axiomatic approach. Here a set of desiderata of an attribution method are defined, on which different attribution methods are then evaluated. Alternatively, different attribution methods may be compared by perturbing the input features starting with the ones deemed most important and measuring the drop in accuracy of the perturbed models. The best method will result into the greatest overall loss in accuracy as the number of inputs are omitted [ACÖG17]. Moreover, for gradient-based methods it is hard to assess if an unexpected attribution is caused by a poorly performing network or a poorly performing attribution method [FV17]. How to cope with negative evidence, i.e., the object was predicted because a contrary clue in the input image was missing, is an open research question. Additionally, most methods were shown on classification tasks until now. It remains to be seen how they can be transferred to object detection and semantic segmentation tasks. In the case of perturbation-based methods, the high computation time and single-image analysis inhibit widespread application.

## 5.4 Interpretable Proxies

Neural networks are capable of capturing complicated logical (cor)relations. However, this knowledge is encoded on a *sub-symbolic* level in the form of learned weights and biases, meaning that the reasoning behind the processing chain cannot be directly read out or interpreted by humans [CCO98]. To explain the sub-symbolic processing, one can either use attribution methods (cf. Sect. 5.3), or lift this sub-symbolic representation to a *symbolic* one [GBY+18], meaning a more interpretable one. Interpretable proxies or surrogate models try to achieve the latter: the DNN behavior is approximated by a model that uses symbolic knowledge representations. Symbolic representations can be linear models like local interpretable model-agnostic explanations (LIME) [RSG16] (proportionality), decision trees (if-then chains) [GBY+18], or loose sets of logical rules. Logical connectors can simply be AND and OR but also more general ones like at-least-M-of-N [CCO98]. The *expressiveness* of an approach refers to the logic that is used: Boolean-only versus first-order logic, and binary versus fuzzy logic truth values [TAGD98]. Other than attribution methods (cf.

Sect. 5.3), these representations can capture combinations of features and (spatial) relations of objects and attributes. As an example, consider “eyes are closed” as explanation for “person asleep”: attribution methods only could mark the location of the eyes, dismissing the relations of the attributes [RSS18]. All mentioned surrogate model types (linear, set of rules) require interpretable input features in order to be interpretable themselves. These features must either be directly obtained from the DNN input or (intermediate) output, or automatically be extracted from the DNN representation. Examples for extraction are the super-pixeling used in LIME for input feature detection, or concept activation vectors [KWG+18] for DNN representation decoding.

Quality criteria and goals for interpretable proxies are [TAGD98]: *accuracy* of the standalone surrogate model on unseen examples, *fidelity* of the approximation by the proxy, *consistency* with respect to different training sessions, and *comprehensibility* measured by the complexity of the rule set (number of rules, number of hierarchical dependencies). The criteria are usually in conflict and need to be balanced: Better accuracy may require a more complex, thus less expressive set of rules.

Approaches for interpretable proxies differ in the validity range of the representations: Some aim for surrogates that are only valid *locally* around specific samples, like in LIME [RSG16] or in [RSS18] via inductive logic programming. Other approaches try to be more *globally* approximate aspects of the model behavior. Another categorization is defined by whether full access (*white box*), some access (*gray box*), or no access (*black box*) to the DNN internals is needed. One can further differentiate between *post hoc* approaches that are applied to a trained model, and approaches that try to integrate or *enforce symbolic representations* during training. Post hoc methods cover the wide field of rule extraction techniques for DNNs. The interested reader may refer to [AK12, Hai16]. Most white- and gray-box methods try to turn the DNN connections into if-then rules that are then simplified, like done in DeepRED [ZLMJ16]. A black-box example is validity interval analysis [Thr95], which refines or generalizes rules on input intervals, either starting from one sample or a general set of rules. Enforcement of symbolic representations can be achieved by enforcing an output structure that provides insights to the decision logic, such as textual explanations or a rich output structure allowing investigation of correlations [XLZ+18]. An older discipline for enforcing symbolic representations is the field of neural-symbolic learning [SSZ19]. The idea is based on a hybrid learning cycle in which a symbolic learner and a DNN iteratively update each other via rule insertion and extraction. The comprehensibility of global surrogate models suffers from the complexity and size of concurrent DNNs. Thus, stronger rule simplification methods are required [Hai16]. The alternative direction of local approximations mostly concentrates on linear models instead of more expressive rules [Thr95, RSS18]. Furthermore, balancing of the quality objectives is hard since available indicators for interpretability may not be ideal. And lastly, applicability is heavily infringed by the requirement of interpretable input features. These are usually not readily available from input (often pixel level) or DNN output. Supervised extraction approaches vary in their fidelity, and unsupervised ones do not guarantee to yield meaningful or interpretable results, respectively, such as the super-pixel clusters of LIME.

## 6 Uncertainty

Uncertainty refers to the view that a neural network is not conceived as a deterministic function but as a probabilistic function or estimator, delivering a random distribution for each input point. Ideally, the mean value of the distribution should be as close as possible to the ground-truth value of the function being approximated by the neural network and the uncertainty of the neural network refers to its variance when being considered as a random variable, thus allowing to derive a confidence with respect to the mean value. Regarding safety, the variance may lead to estimations about the confidence associated with a specific network output and opens the option for discarding network outputs with insufficient confidence.

There are roughly two broad approaches for training neural networks as probabilistic functions: parametric approaches [KG17] and Bayesian neural networks on the one hand, such as [BCKW15], where the transitions along the network edges are modeled as probability distributions, and ensemble-based approaches on the other hand [LPB17, SOF+19], where multiple networks are trained and considered as samples of a common output distribution. Apart from training as probabilistic function, uncertainty measures have been derived from single, standard neural networks by post-processing on the trained network logits, leading, for example, to calibration measures (cf. e.g., [SOF+19]).

### 6.1 Generative Models

*Generative models* belong to the class of unsupervised machine learning models. From a theoretical perspective, these are particularly interesting, because they offer a way to analyze and model the density of data. Given a finite dataset  $\mathcal{D}$  independently distributed according to some distribution  $p(\mathbf{x})$ ,  $\mathbf{x} \in \mathcal{D}$ , generative models aim to estimate or enable sampling from the underlying density  $p(\mathbf{x})$  in a model  $\mathbf{F}(\mathbf{x}, \theta)$ . The resulting model can be used for data indexing [Wes04], data retrieval [ML11], for visual recognition [KSH12], speech recognition and generation [HDY+12], language processing [KM02, CE17], and robotics [Thr02]. Following [OE18], we can group generative models into two main classes:

- Cost function-based models, such as autoencoder [KW14, Doe16], deep belief networks [Hin09], and generative adversarial networks [GPAM+14, RMC15, Goo17].
- Energy-based models [LCH+07, SH09], where the joint probability density is modeled by an energy function.

Besides these *deep learning* approaches, generative models have been studied in machine learning in general for quite some time (cf. [Fry77, Wer78, Sil86, JMS96, She04, Sco15, Gra18]). A very prominent example of generative networks is Gaussian process [WR96, Mac98, WB98, Ras03] and their deep learning extensions [DL13, BHLHL+16] as generative models.

An example of a generative model being employed for image segmentation uncertainty estimation is the probabilistic U-Net [KRPM+18]. Here a variational autoencoder (VAE) conditioned on the image is trained to model uncertainties. Samples from the VAE are fed into a segmentation U-Net which can thus give different results for the same image. This was tested in context of medical images, where inter-rater disagreements lead to uncertain segmentation results and Cityscapes segmentation. For the Cityscapes segmentation, the investigated use case was label ambiguity (e.g., is a BMW X7 a car or a van) using artificially created, controlled ambiguities. Results showed that the probabilistic U-Net could reproduce the segmentation ambiguity modes more reliably than competing methods, such as a dropout U-Net which is based on techniques elaborated in the next section.

## 6.2 Monte Carlo Dropout

A widely used technique to estimate model uncertainty is Monte Carlo (MC) dropout [GG16] that offers a Bayesian motivation, conceptual simplicity, and scalability to application-size networks. This combination distinguishes MC dropout from competing Bayesian neural network (BNN) approximations (e.g., [RBB18, BCKW15], see Sect. 6.3). However, these approaches and MC dropout share the same goal: to equip neural networks with a *self-assessment* mechanism that detects unknown input concepts and thus potential model insufficiencies.

On a technical level, MC dropout assumes prior distributions on network activations, usually independent and identically distributed (i.i.d.) Bernoulli distributions. Model training with iteratively drawn Bernoulli samples, the so-called *dropout masks*, then yields a data-conditioned posterior distribution within the chosen parametric family. It is interesting to note that this training scheme was used earlier—*independent from an uncertainty context*—for better model generalization [SHK+14]. At inference, sampling provides estimates of the input-dependent output distributions. The spread of these distributions is then interpreted as the prediction uncertainty that originates from limited knowledge of model parameters. Borrowing “frequentist” terms, MC dropout can be considered as an implicit network ensemble, i.e., as a set of networks that share (most of) their parameters.

In practice, MC dropout requires only a minor modification of the optimization objective during training and multiple, trivially parallelizable forward passes during inference. The loss modification is largely agnostic to network architecture and does not cause substantial overhead. This is in contrast to the sampling-based inference that increases the computational effort massively—by estimated factors of 20–100 compared to networks without MC dropout. A common practice is therefore the use of *last-layer dropout* [SOF+19] that reduces computational overhead to estimated factors of 2–10. Alternatively, analytical moment propagation allows sampling-free MC dropout inference at the price of additional approximations (e.g., [PFC+19]). Further extensions of MC dropout target the integration of data-inherent (aleatoric)

uncertainty [KG17] and tuned performance by learning layer-specific dropout rate using concrete relaxations [GHK17].

The quality of MC dropout uncertainties is typically evaluated using negative log-likelihood (NLL), expected calibration error (ECE) and its variants (cf. Sect. 6.6) and by considering correlations between uncertainty estimates and model errors (e.g., AUSE [ICG+18]). Moreover, it is common to study how useful uncertainty estimates are for solving auxiliary tasks like out-of-distribution classification [LPB17] or robustness w.r.t. adversarial attacks.

MC dropout is a working horse of safe ML, being used with various networks and for a multitude of applications (e.g., [BFS18]). However, several authors pointed out shortcomings and limitations of the method: MC dropout bears the risk of overconfident false predictions ([Osb16]), offers less diverse uncertainty estimates compared to (the equally simple and scalable) deep ensembles ([LPB17], see Sect. 7.1), and provides only rudimentary approximations of true posteriors.

Relaxing these modeling assumptions and strengthening the Bayesian motivation of MC dropout is therefore an important research avenue. Further directions for future work are the development of *semantic uncertainty mechanisms* (e.g., [KRPM+18]), improved local uncertainty calibrations, and a better understanding of the outlined sampling-free schemes to uncertainty estimation.

### 6.3 Bayesian Neural Networks

As the name suggests, Bayesian neural networks (BNNs) are inspired by a Bayesian interpretation of probability (for an introduction cf. [Mac03]). In essence, it rests on Bayes' theorem,

$$p(a|b)p(b) = p(a, b) = p(b|a)p(a) \Rightarrow p(a|b) = \frac{p(b|a)p(a)}{p(b)}, \quad (2)$$

stating that the conditional probability density function (PDF)  $p(a|b)$  for  $a$  given  $b$  may be expressed in terms of the inverted conditional PDF  $p(b|a)$ . For machine learning, where one intends to make predictions  $y$  for unknown  $\mathbf{x}$  given some training data  $\mathcal{D}$ , this can be reformulated into

$$y = \text{NN}(\mathbf{x}|\boldsymbol{\theta}) \quad \text{with} \quad p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}. \quad (3)$$

Therein NN denotes a conventional (deep) neural network (DNN) with model parameters  $\boldsymbol{\theta}$ , e.g., the set of weights and biases. In contrast to a regular DNN, the weights are given in terms of a probability distribution  $p(\boldsymbol{\theta}|\mathcal{D})$  turning also the output  $y$  of a BNN into a distribution. This allows to study the mean  $\mu = \langle y^1 \rangle$  of the DNN for a given  $\mathbf{x}$  as well as higher moments of the distribution, typically the resulting variance  $\sigma^2 = \langle (y - \mu)^2 \rangle$  is of interest, where

$$\langle y^k \rangle = \int \text{NN}(\mathbf{x}|\boldsymbol{\theta})^k p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}. \quad (4)$$

While  $\mu$  yields the output of the network, the variance  $\sigma^2$  is a measure for the uncertainty of the model for the prediction at the given point. Central to this approach is the probability of the data given the model, here denoted by  $p(\mathcal{D}|\boldsymbol{\theta})$ , as it is the key component connecting model and training data. Typically, the prior distribution  $p(\boldsymbol{\theta})$  is “ignored” as it only appears as a normalization constant within the averages, see (4). In the cases where the data  $\mathcal{D}$  is itself a distribution due to inherent uncertainty, i.e., presence of aleatoric risk [KG17], such a concept seems natural. However, Bayesian approaches are also applicable for all other cases. In those, loosely speaking, the likelihood of  $\boldsymbol{\theta}$  is determined via the chosen loss function (for the connection between the two concepts cf. [Bis06]).

On this general level, Bayesian approaches are broadly accepted and also find use for many other model classes besides neural networks. However, the loss surfaces of DNNs are known for their high dimensionality and strong non-convexity. Typically, there are abundant parameter combinations  $\boldsymbol{\theta}$  that lead to (almost) equally good approximations to the training data  $\mathcal{D}$  with respect to a chosen loss. This makes an evaluation of  $p(\boldsymbol{\theta}|\mathcal{D})$  for DNNs close to impossible in full generality. At least no (exact) solutions for this case exist at the moment. Finding suitable approximations to the posterior distribution  $p(\boldsymbol{\theta}|\mathcal{D})$  is an ongoing challenge for the construction of BNNs. At this point we only summarize two major research directions in the field. One approach is to assume that the distribution factorizes. While the full solution would be a joint distribution implying correlations between different weights (etc.), possibly even across layers, this approximation takes each element of  $\boldsymbol{\theta}$  to be independent from the others. Although this is a strong assumption, it is often made, in this case, parameters for the respective distributions of each element can be learned via training (cf. [BCKW15]). The second class of approaches focuses on the region of the loss surface around the minimum chosen for the DNN. As discussed, the loss relates to the likelihood and quantities such as the curvature at the minimum, therefore directly connected to the distribution of  $\boldsymbol{\theta}$ . Unfortunately, already using this type of quantities requires further approximations [RBB18]. Alternatively, the convergence of the training process may be altered to sample networks close to the minimum [WT11]. While this approach contains information about correlations among the weights, it is usually restricted to a specific minimum. For a non-Bayesian approach taking into account several minima, see deep ensembles in Sect. 7.1. BNNs also touch other concepts such as MC dropout (cf. [GG16] and Sect. 7.1), or prior networks, which are based on a Bayesian interpretation but use conventional DNNs with an additional (learned)  $\sigma$  output [MG18].

## 6.4 Uncertainty Metrics for DNNs in Frequentist Inference

Classical uncertainty quantification methods in frequentist inference are mostly based on the outputs of statistical models. Their uncertainty is quantified and assessed, for instance, via dispersion measures in classification (such as entropy, probability margin, or variation ratio), or confidence intervals in regression. However, the nature of DNN architectures [RDGF15, CPK+18] and the cutting-edge applications tackled by those (e.g., semantic segmentation, cf. [COR+16]) open the way toward more elaborate uncertainty quantification methods. Besides the mentioned classical approaches, intermediate feature representations within a DNN (cf. [CZYS19, OSM19]) or gradients according to self-affirmation that represent re-learning stress (see [ORG18]) reveal additional information. In addition, in case of semantic segmentation, the geometry of a prediction may give access to further information, cf. [MRG20, RS19, RCH+20]. By the computation of statistics of those quantities as well as low-dimensional representations thereof, we obtain more elaborate uncertainty quantification methods specifically designed for DNNs that can help us to detect misclassifications and out-of-distribution objects (cf. [HG17]). Features gripped during a forward pass of a data point  $\mathbf{x}$  through a DNN  $\mathbf{F}$  can be considered layer-wise, i.e.,  $\mathbf{F}_\ell(\mathbf{x})$  after the  $\ell$ -th layer. These can be translated into a handful of quantities per layer [OSM19] or further processed by another DNN that aims at detecting errors [CZYS19]. While, in particular, [OSM19] presents a proof of concept on small-scale classification problems, their applicability to large-scale datasets and problems, such as semantic segmentation and object detection, remains open.

The development for gradient-based uncertainty quantification methods [ORG18] is guided by one central question: if the present prediction was true, how much re-learning would this require. The corresponding hypothesis is that wrong predictions would be more in conflict with the knowledge encoded in the deep neural network than correct ones, therefore causing increased re-learning stress. Given a predicted class

$$\hat{y} = \arg \max_{s \in S} \mathbf{F}(\mathbf{x}) \quad (5)$$

we compute the gradient of layer  $\ell$  corresponding to the predicted label. That is, given a loss function  $J$ , we compute

$$\nabla_\ell J(\hat{y}, \mathbf{x}, \theta) \quad (6)$$

via backpropagation. The obtained quantities can be treated similarly to the case of forward pass features. While this concept seems to be prohibitively expensive for semantic segmentation (at least when calculating gradients for each pixel of the multidimensional output  $\hat{\mathbf{y}}$ ), its applicability to object detection might be feasible, in particular, with respect to offline applications. Gradients are also of special interest in active learning with query by *expected model change* (cf. [Set10]).

In the context of semantic segmentation, geometrical information on segments' shapes as well as neighborhood relations of predicted segments can be taken into

account alongside with dispersion measures. It has been demonstrated [RS19, MRG20, RCH+20] that the detection of errors in an in-distribution setting strongly benefits from geometrical information. Recently, this has also been considered in scenarios under moderate domain shift [ORF20]. However, its applicability to out-of-distribution examples and to other sensors than the camera remains subject to further research.

The problem of realistically quantifying uncertainty measures will be taken up in Chapter “[Uncertainty Quantification for Object Detection: Output- and Gradient-based Approaches](#)” [RSKR22]. Here output-based and learning-gradient-based uncertainty metrics for object detection will be examined, showing that they are non-correlated. Thus, a combination of both paradigms leads, as is demonstrated, to a better object detection uncertainty estimate and by extension to a better overall detection accuracy.

## 6.5 Markov Random Fields

Although deep neural networks are currently the state of the art for almost all computer vision tasks, Markov random fields (MRF) remain one of the fundamental techniques used for many computer vision tasks, specifically image segmentation [LWZ09, KK11]. MRFs hold its power in the essence of being able to model dependencies between pixels in an image. With the use of energy functions, MRFs integrate pixels into models relating between unary and pair-wise pixels together [WKP13]. Given the model, MRFs are used to infer the optimal configuration yielding the lowest energy using mainly maximum a posteriori (MAP) techniques. Several MAP inference approaches are used to yield the optimal configuration such as graph cuts [KRBTO8] and belief propagation algorithms [FZ10]. However, as with neural networks, MAP inference techniques result in deterministic point estimates of the optimal configuration without any sense of uncertainty in the output. To obtain uncertainties on results from MRFs, most of the work is directed toward modeling MRFs with Gaussian distributions. Getting uncertainties from MRFs with Gaussian distributions is possible by two typical methods: either approximate models are inferred to the posterior, from which sampling is easy or the variances can be estimated analytically, or approximate sampling from the posterior is used. Approximate models include those inferred using variational Bayesian (VB) methods, like mean-field approximations, and using Gaussian process (GP) models enforcing a simplified prior model [Bis06, LUAD16]. Examples of approximate sampling methods include traditional Markov chain Monte Carlo (MCMC) methods like Gibbs sampling [GG84]. Some recent theoretical advances propose the perturb-and-MAP framework and a Gumbel perturbation model (GPM) [PY11, HMJ13] to exactly sample from MRF distributions. Another line of work has also been proposed, where MAP inference techniques are used to estimate the probability of the network output. With the use of graph cuts, [KT08] try to estimate uncertainty using the min-marginals associated with the label assignments of a random field. Here, the work by Kohli and Torr [KT08]

was extended to show how this approach can be extended to techniques other than graph cuts [TA12] or compute uncertainties on multi-label marginal distributions [STP17]. A current research direction is the incorporation of MRFs with deep neural networks, along with providing uncertainties on the output [SU15, CPK+18]. This can also be extended to other forms of neural networks such as recurrent neural networks to provide uncertainties on segmentation of streams of videos with extending dependencies of pixels to previous frames [ZJRP+15, LLL+17].

## 6.6 Confidence Calibration

Neural network classifiers output a label  $\hat{Y} \in \mathcal{Y}$  on a given input  $\mathbf{x} \in \mathcal{D}$  with an associated confidence  $\hat{P}$ . This confidence can be interpreted as a probability of correctness that the predicted label matches the ground-truth label  $\bar{Y} \in \mathcal{Y}$ . Therefore, these probabilities should reflect the “self-confidence” of the system. If the empirical accuracy for any confidence level matches the predicted confidence, a model is called *well calibrated*. Therefore, a *classification* model is perfectly calibrated if

$$\underbrace{P(\hat{Y} = Y | \hat{P} = p)}_{\text{accuracy given } p} = \underbrace{p}_{\text{confidence}} \quad \forall p \in [0, 1] \quad (7)$$

is fulfilled [GPSW17]. For example, assume 100 predictions with confidence values of 0.9. We call the model well calibrated if 90 out of these 100 predictions are actually correct. However, recent work has shown that modern neural networks tend to be too overconfident in their predictions [GPSW17]. The deviation of a model to the perfect calibration can be measured by the expected calibration error (ECE) [NCH15]. It is possible to recalibrate models as a post-processing step after classification. One way to get a calibration mapping is to group all predictions into several bins by their confidence. Using such a binning scheme, it is possible to compute the empirical accuracy for certain confidence levels, as it is known for a long time already in reconstructing confidence outputs for Viterbi decoding [HR90]. Common methods are histogram binning [ZE01], isotonic regression [ZE02], or more advanced methods like Bayesian binning into quantiles (BBQ) [NCH15] and ensembles of near-isotonic regression (ENIR) [NC16]. Another way to get a calibration mapping is to use scaling methods based on logistic regression like Platt scaling [Pla99], temperature scaling [GPSW17], and beta calibration [KSFF17].

In the setting of probabilistic regression, a model is calibrated if 95% of the true target values are below or equal to a credible level of 95% (so-called *quantile-calibrated regression*) [GBR07, KFE18, SDKF19]. A regression model is usually calibrated by fine-tuning its predicted CDF in a post-processing step to match the empirical frequency. Common approaches utilize isotonic regression [KFE18], logistic and beta calibration [SKF18], as well as Gaussian process models [SKF18, SDKF19] to build a calibration mapping. In contrast to quantile-calibrated regression, [SDKF19]

have recently introduced the concept of distribution calibration, where calibration is applied on a distribution level and naturally leads to calibrated quantiles. Recent work has shown that miscalibration in the scope of *object detection* also depends on the position and scale of a detected object [KKSH20]. The additional box regression output is denoted by  $\hat{\mathbf{R}}$  with  $A$  as the size of the used box encoding. Furthermore, if we have no knowledge about all anchors of a model (which is a common case in many applications), it is not possible to determine the accuracy. Therefore, Küppers et al. [KKSH20] use the precision as a surrogate for accuracy and propose that an *object detection model* is perfectly calibrated if

$$\underbrace{P(M = 1 | \hat{P} = p, \hat{Y} = y, \hat{\mathbf{R}} = \mathbf{r})}_{\text{precision given } p, y, \mathbf{r}} = \underbrace{p}_{\text{confidence}} \quad \forall p \in [0, 1], y \in \mathcal{Y}, \mathbf{r} \in \mathbb{R}^A \quad (8)$$

is fulfilled, where  $M = 1$  denotes a correct prediction that matches a ground-truth object with a chosen IoU threshold and  $M = 0$  denotes a mismatch, respectively. The authors propose the detection-expected calibration error (D-ECE) as the extension of the ECE to object detection tasks in order to measure miscalibration also by means of the position and scale of detected objects. Other approaches try to fine-tune the regression output in order to obtain more reliable object proposals [JLM+18, RTG+19] or to add a regularization term to the training objective such that training yields models that are both well performing and well calibrated [PTC+17, SSH19].

In Chapter “[Confidence Calibration for Object Detection and Segmentation](#)” [KHKSS22], an extension of the ECE for general multivariate learning problems, such as object detection or image segmentation, will be discussed. In fact, the proposed multivariate confidence calibration is designed to take additional information into account, such as bounding boxes or shape descriptors. It is shown that this extended calibration reduces the calibration error as expected but also has a positive bearing on the quality of segmentation masks.

## 7 Aggregation

From a high-level perspective, a *neural network* is based on processing inputs and coming to some output conclusion, e.g., mapping incoming image data onto class labels. Aggregation or collection of non-independent information on either the input or output side of this network function can be used as a tool to leverage its performance and reliability. Starting with the input, any additional “dimension” to add data can be of use. For example, in the context of automated vehicles, this might be input from any further sensor measuring the same scene as the original one, e.g., stereo cameras or LiDAR. Combining those sensor sets for prediction is commonly referred to as *sensor fusion* [CBSW19]. Staying with the example, the scene will be monitored consecutively providing a whole (temporally ordered) stream of input information. This may be used either by adjusting the network for this kind of input [KLX+17] or

in terms of a post-processing step, in which the predictions are aggregated by some measure of temporal consistency.

Another more implicit form of aggregation is training the neural network on several “independent” tasks, e.g., segmentation and depth regression. Although the individual task is executed on the same input, the overall performance can still benefit from the correlation among all given tasks. We refer to the discussion on *multi-task networks* in Sect. 9.2. By extension, solving the same task in multiple different ways can be beneficial for performance and provide a measure of redundancy. In this survey, we focus on single-task systems and discuss *ensemble methods* in the next section and the use of *temporal consistency* in the one thereafter.

## 7.1 Ensemble Methods

Training a neural network is optimizing its parameters to fit a given training dataset. The commonly used gradient-based optimization schemes cause convergence in a “nearby” local minimum. As the loss landscapes of neural networks are notoriously non-convex [CHM+15], various locally optimal model parameter sets exist. These local optima differ in the degree of optimality (“deepness”), qualitative characteristics (“optimal for different parts of the training data”), and their generalizability to unseen data (commonly referred to by the geometrical terms of “sharpness” and “flatness” of minima [KMN+17]).

A single trained network corresponds to one local minimum of such a loss landscape and thus captures only a small part of a potentially diverse set of solutions. *Network ensembles* are collections of models and therefore better suited to reflect this multi-modality. Various modeling choices shape a loss landscape: the selected model class and its meta-parameters (like architecture and layer width), the training data and the optimization objective. Accordingly, approaches to diversify ensemble components range from combinations of different model classes over varying training data (bagging) to methods that train and weight ensemble components to make up for the flaws of other ensemble members (boosting) [Bis06].

Given the millions of parameters of application-size networks, ensembles of NNs are resource-demanding w.r.t. computational load, storage, and runtime during training and inference. This complexity increases linearly with ensemble size for naïve ensembling. Several approaches were put forward to reduce some dimensions of this complexity: *snapshot ensembles* [HLP+17] require only one model optimization with a cyclical learning-rate schedule leading to an optimized training runtime. The resulting training trajectory passes through several local minima. The corresponding models compose the ensemble. On the contrary, *model distillation* [HVD14] tackles runtime at inference. They “squeeze” an NN ensemble into a single model that is optimized to capture the gist of the model set. However, such a compression goes along with reduced performance compared to the original ensemble.

Several hybrids of single model and model ensemble exist: multi-head networks [AJD18] share a backbone network that provides inputs to multiple prediction net-

works. Another variant is mixture-of-expert model that utilizes a gating network to assign inputs to specialized expert networks [SMM+17]. Multi-task networks (cf. Sect. 9.2) and Bayesian approximations of NNs (cf. Sects. 6.2 and 6.3) can be seen as implicit ensembles.

NN ensembles (or deep ensembles) are not only used to boost model quality. They pose the frequentist’s approach to estimating NN uncertainties and are state of the art in this regard [LPB17, SOF+19]. The emergent field of federated learning is concerned with the integration of decentrally trained ensemble components [MMR+17] and safety-relevant applications of ensembling range from automated driving [Zha12] to medical diagnostics [RRMH17]. Taking this safe-ML perspective, promising research directions comprise a more principled and efficient composition of model ensembles, by application-driven diversification as well as improved techniques to miniaturize ensembles and by gaining a better understanding of methods like model distillation. In the long run, better designed, more powerful learning systems might partially reduce the need for combining weaker models in a network ensemble.

In Chapter “[Evaluating Mixture-of-Expert Architectures for Network Aggregation](#)” [PHW22], the advantages and drawbacks of mixture-of-experts architectures with respect to robustness, interpretability, and overall test performance will be discussed. Two state-of-the-art architectures are investigated, obtaining, sometimes surpassing, baseline performance. What is more, the models exhibit improved awareness for OoD data.

## 7.2 Temporal Consistency

The focus of previous DNN development for semantic segmentation has been on single-image prediction. This means that the final and intermediate results of the DNN are discarded after each image. However, the application of a computer vision model often involves the processing of images in a sequence, i.e., there is a temporal consistency in the image content between consecutive frames (for a metric, cf., e.g., [VBB+20]). This consistency has been exploited in previous work to increase quality and reduce computing effort. Furthermore, this approach offers the potential to improve the robustness of DNN prediction by incorporating this consistency as a priori knowledge into DNN development. The relevant work in the field of video prediction can be divided into two major approaches:

1. DNNs are specially designed for video prediction. This usually requires training from scratch and the availability of training data in a sequence.
2. A transformation from single prediction DNNs to video prediction DNNs takes place. Usually no training is required, i.e., the existing weights of the model can be used unaltered.

The first set of approaches often involves *conditional random fields* (CRF) and its variants. CRFs are known for their use as post-processing step in the prediction of

semantic segmentation, in which their parameters are learned separately or jointly with the DNN [ZJRP+15]. Another way to use spatiotemporal features is to include 3D convolutions, which add an additional dimension to the conventional 2D convolutional layer. Tran et al. [TBF+15] use 3D convolution layers for video recognition tasks, such as action and object recognition. One further approach to use spatial and temporal characteristics of the input data is to integrate *long short-term memory* (LSTM) [HS97], a variant of the *recurrent neural network* (RNN). Fayyaz et al. [FSS+16] integrate LSTM layers between the encoder and decoder of their convolutional neural network for semantic segmentation. The significantly higher GPU memory requirements and computational effort are disadvantages of this method. More recently, Nilsson and Sminchisescu [NS18] deployed *gated recurrent units*, which generally require significantly less memory. An approach to improve temporal consistency of automatic speech recognition outputs is known as a posterior-in-posterior-out (PIPO) LSTM “sequence enhancer”, a postfilter which could be applicable to video processing as well [LSFs19]. A disadvantage of the described methods is that sequential data for training must be available, which may be limited or show a lack of diversity.

The second class of approaches has the advantage that it is model independent most of the time. Shelhamer et al. [SRHD16] found that the deep feature maps within the network change only slightly with temporal changes in video content. Accordingly, [GJG17] calculate the optical flow of the input images from time steps  $t = 0$  and  $t = -1$  and convert it into the so-called *transform flow* which is used to transform the feature maps of the time step  $t = -1$  so that an aligned representation to the feature map  $t = 0$  is achieved. Sämann et al. [SAMG19] use a confidence-based combination of feature maps from previous time steps based on the calculated optical flow.

## 8 Verification and Validation

Verification and validation is an integral part of the safety assurance for any safety-critical systems. As of the functional safety standard for automotive systems [ISO18], *verification* means to determine whether given requirements are met [ISO18, 3.180], such as performance goals. *Validation*, on the other side, tries to assess whether the given requirements are sufficient and adequate to guarantee safety [ISO18, 3.148], e.g., whether certain types of failures or interactions simply were overlooked. The latter is usually achieved via extensive testing in real operation conditions of the integrated product. This differs from the notion of validation used in the machine learning community in which it usually refers to simple performance tests on a selected dataset. In this section, we want to concentrate on general verification aspects for deep neural networks.

Verification as in the safety domain encompasses (manual) inspection and analysis activities, and testing. However, the contribution of single processing steps within a neural network to the final behavior can hardly be assessed manually (compared

to the problem of interpretability in Sect. 5). Therefore, we here will concentrate on different approaches to verification testing, i.e., formal testing and black-box methods.

The discussion on how to construct an assurance case and how to derive suitable acceptance criteria for ML models in the scope of autonomous driving, as well as how to determine and acquire sufficient evidences for such an assurance case is provided in Chapter “[Safety Assurance of Machine Learning for Perception Functions](#)” [BHH+22]. The authors look into several use cases on how to systematically obtain evidences and incorporate them into a structured safety argumentation.

In Chapter “[A Variational Deep Synthesis Approach for Perception Validation](#)” [GHS22], a novel data generation framework is introduced, aiming at validating perception functions, in particular, in the context of autonomous driving. The framework allows for effectively defining and testing common traffic scenes. Experiments are performed for pedestrian detection and segmentation in versatile urban scenarios.

## 8.1 Formal Testing

Formal testing refers to testing methods that include formalized and formally verifiable steps, e.g., for test data acquisition, or for verification in the local vicinity of test samples. For image data, local testing around valid samples is usually more practical than fully formal verification: (safety) properties are not expected to hold on the complete input space but only on the much smaller unknown lower dimensional manifold of real images [WGH19]. Sources of such samples can be real ones or generated ones using an input space formalization or a trained generative model.

Coverage criteria for the data samples are commonly used for two purposes: (a) deciding when to stop testing or (b) identifying missing tests. For CNNs, there are at least three different approaches toward coverage: (1) approaches that establish coverage based on a model with semantic features of the input space [GHHW20], (2) approaches trying to semantically cover the latent feature space of neural network or a proxy network (e.g., an autoencoder) [SS20a], and (3) approaches trying to cover neurons and their interactions, inspired by classical software white-box analysis [PCYJ19, SHK+19].

Typical types of properties to verify are simple test performance, local stability (robustness), a specific structure of the latent spaces like embedding of semantic concepts [SS20b], and more complex logical constraints on inputs and outputs, which can be used for testing when fuzzified [RDG18]. Most of these properties require in-depth semantic information about the DNN inner workings, which is often only available via interpreting intermediate representations [KGW+18], or interpretable proxies/surrogates (cf. Sect. 5.4), which do not guarantee fidelity.

There exist different testing and formal verification methods from classical software engineering that have already been applied to CNNs. *Differential testing* as used by DeepXplore [PCYJ19] trains  $n$  different CNNs for the same task using independent datasets and compares the individual prediction results on a test set. This

allows to identify inconsistencies between the CNNs but no common weak spots. *Data augmentation* techniques start from a given dataset and generate additional transformed data. *Generic data augmentation* for images like rotations and translation is state of the art for training but may also be used for testing. *Concolic testing* approaches incrementally grow test suites with respect to a coverage model to finally achieve completeness. Sun et al. [SWR+18] use an adversarial input model based on some norm (cf. Sect. 4.1), e.g., an  $L_p$ -norm, for generating additional images around a given image using concolic testing. *Fuzzing* generates new test data constrained by an input model and tries to identify interesting test cases, e.g., by optimizing white-box coverage mentioned above [OOAG19]. Fuzzing techniques may also be combined with the differential testing approach discussed above [GJZ+18]. In all these cases, it needs to be ensured that the image as well as its meta-data remains valid for testing after transformation. Finally, *proving methods* surveyed by Liu et al. [LAL+20] try to formally prove properties on a trained neural network, e.g., based on satisfiability modulo theories (SMT). These approaches require a formal characterization of an input space and the property to be checked, which is hard for non-trivial properties like contents of an image.

Existing formal testing approaches can be quite costly to integrate into testing workflows: differential testing and data augmentation require several inferences per initial test sample; concolic and fuzzy testing apply an optimization to each given test sample, while convergence toward the coverage goals is not guaranteed; also, the iterative approaches need tight integration into the testing workflow; and lastly, proving methods usually have to balance computational efficiency against the precision or completeness of the result [LAL+20]. Another challenge of formal testing is that machine learning applications usually solve problems for which no (formal) specification is possible. This makes it hard to find useful requirements for testing [ZHML20] and properties that can be formally verified. Even partial requirements, such as specification of useful input perturbations, specified corner cases, and valuable coverage goals, are typically difficult to identify [SS20a, BTLFs20].

## 8.2 Black-Box Methods

In machine learning literature, neural networks are often referred to as black boxes due to the fact that their internal operations and their decision-making are not completely understood [SZT17], hinting at a lack of interpretability and transparency. However, in this survey, we consider a black box to be a machine learning model to which we only have Oracle (query) access [TZJ+16, PMG+17]. That means we can query the model to get input–output pairs, but we do not have access to the specific architecture (or weights, in case of neural networks). As [OAFS18] describes, black boxes are increasingly widespread, e.g., health care, autonomous driving, or ML as a service in general, due to proprietary, privacy, or security reasons. As deploying black boxes gains popularity, so do methods that aim to extract internal information, such as architecture and parameters, or to find out, whether a sample belongs to the

training dataset. These include *model extraction attacks* [TZJ+16, KTP+20], *membership inference attacks* [SSSS17], general attempts to reverse-engineer the model [OAFS18], or to attack it adversarially [PMG+17]. Protection and counter-measures are also actively researched: Kesarwani et al. [KMAM18] propose a warning system that estimates how much information an attacker could have gained from queries. Adi et al. [ABC+18] use watermarks for models to prevent illegal re-distribution and to identify intellectual property.

Many papers in these fields make use of so-called *surrogate*, *avatar*, or *proxy models* that are trained on input–output pairs of the black box. In case the black-box output is available in soft form (e.g., logits), distillation as first proposed by [HVD14] can be applied to train the surrogate (student) model. Then, any white-box analysis can be performed on the surrogates (cf. e.g., [PMG+17]) to craft adversarial attacks targeted at the black box. More generally, (local) surrogates, as, for example, in [RSG16], can be used to (locally) explain its decision-making. Moreover, these techniques are also of interest if one wants to compare or test black-box models (cf. Sect. 8.1). This is the case, among others, in ML marketplaces, where you wish to buy a pre-trained model [ABC+18], or if you want to verify or audit that a third-party black-box model obeys regulatory rules (cf. [CH19]).

Another topic of active research is so-called *observers*. The concept of observers is to evaluate the interface of a black-box module to determine if it behaves as expected within a given set of parameters. The approaches can be divided into *model-explaining* and *anomaly-detecting* observers. First, model explanation methods answer the question of which input characteristic is responsible for changes at the output. The observer is able to alter the inputs for this purpose. If the input of the model under test evolves only slightly but the output changes drastically, this can be a signal that the neural network is misled, which is also strongly related to adversarial examples (cf. Sect. 4). Hence, the reason for changes in the classification result via the input can be very important. In order to figure out in which region of an input image the main reason for the classification is located, [FV17] “delete” information from the image by replacing regions with generated patches until the output changes. This replaced region is likely responsible for the decision of the neural network. Building upon this, [UEKH19] adapt the approach to medical images and generate “deleted” regions by a variational autoencoder (VAE). Second, anomaly-detecting observers register input and output anomalies, either examining input and output independently or as an input–output pair, and predict the black-box performance in the current situation. In contrast to model-explaining approaches, this set of approaches has high potential to be used in an online scenario since it does not need to modify the model input. The maximum mean discrepancy (MMD) [BGR+06] measures the domain gap between two data distributions independently from the application and can be used to raise a warning if input or output distributions during inference deviate too strongly from their respective training distributions. Another approach is using a GAN-based autoencoder [LFK+20] to perform a domain shift estimation where the Wasserstein distance is used as domain mismatch metric. This metric can also be evaluated by use of a casual time-variant aggregation of distributions during inference time.

## 9 Architecture

In order to solve a specific task, the architecture of a CNN and its building blocks play a significant role. Since the early days of using CNNs in image processing, when they were applied to handwriting recognition [LBD+89] and the later breakthrough in general image classification [KSH12], the architecture of the networks has changed radically. Did the term of *deep learning* for these first convolutional neural networks imply a depth of approximately four layers, their depth increased significantly during the last years and new techniques had to be developed to successfully train and utilize these networks [HZRS16]. In this context, new activation functions [RZL18] as well as new loss functions [LGG+17] have been designed and new optimization algorithms [KB15] were investigated.

With regard to the layer architecture, the initially alternating repetition of convolution and pooling layers as well as their characteristics have changed significantly. The convolution layers made the transition from a few layers with often large filters to many layers with small filters. A further trend was then the definition of entire modules, which were used repeatedly within the overall architecture as so-called *network in network* [LCY14].

In areas such as automated driving, there is also a strong interest in the simultaneous execution of different tasks within one single convolutional neural network architecture. This kind of architecture is called *multi-task learning* (MTL) [Car97] and can be utilized in order to save computational resources and at the same time to increase performance of each task [KTMFs20]. Within such multi-task networks, usually one shared feature extraction part is followed by one separate so-called head per task [TWZ+18].

In each of these architectures, manual design using expert knowledge plays a major role. The role of the expert is the crucial point here. In recent years, however, there have also been great efforts to automate the process of finding architectures for networks or, in the best case, to learn them. This is known under the name *neural architecture search* (NAS).

### 9.1 Building Blocks

Designing a convolutional neural network typically includes a number of design choices. The general architecture usually contains a number of convolutional and pooling layers which are arranged in a certain pattern. Convolutional layers are commonly followed by a nonlinear activation function. The learning process is based on a loss function which determines the current error and an optimization function that propagates the error back to the single convolution layers and its learnable parameters.

When CNNs became state of the art in computer vision [KSH12], they were usually built using a few alternating convolutional and pooling layers having a few

fully connected layers in the end. It turned out that better results are achieved with deeper networks and so the number of layers increased [SZ15] over the years. To deal with these deeper networks, new architectures had to be developed. In a first step, to reduce the number of parameters, the convolutional layers with partly large filter kernels were replaced by several layers with small  $3 \times 3$  kernels. Today, most architectures are based on the *network-in-network* principle [LCY14], where more complex modules are used repeatedly. Examples of such modules are the *inception module* from GoogleNet [SLJ+15] or the *residual block* from ResNet [HZRS16]. While the inception module consists of multiple parallel strings of layers, the residual blocks are based on the *highway network* [SGS15], which means that they can bypass the original information and the layers in between are just learning residuals. With ResNeXt [XGD+17] and Inception-ResNet [SIVA17] there already exist two networks that combine both approaches. For most tasks, it turned out that replacing the fully connected layers by convolutional layers is much more convenient making the networks fully convolutional [LSD15]. These so-called *fully convolutional networks* (FCN) are no longer bound to fixed input dimensions. Note that with the availability of convolutional long short-term memory (ConvLSTM) structures also fully convolutional recurrent neural networks (FCRNs) became available for fully scalable sequence-based tasks [SCW+15, SDF+20].

Inside the CNNs, the *rectified linear unit* (ReLU) has been the most frequently used activation function for a long time. However, since this function suffers from problems related to the mapping of all negative values to zero like the vanishing gradient problem, new functions have been introduced in recent years. Examples are the *exponential linear unit* (ELU), *swish* [RZL18], and the *non-parametric linearly scaled hyperbolic tangent* (LiSHT) [RMDC19]. In order to be able to train a network consisting of these different building blocks, the loss function is the most crucial part. This function is responsible for how and what the network ultimately learns and how exactly the training data is applied during the training process to make the network train faster or perform better. So the different classes can be weighted in a classification network with fixed values or so-called  $\alpha$ -balancing according to their probability of occurrence. Another interesting approach is weighting training examples according to their easiness for the current network [LGG+17, WFZ19]. For multi-task learning also weighting tasks based on their uncertainty [KGC18] or gradients [CBLR18] can be done as further explained in Sect. 9.2. A closer look on how a modification of the loss function might affect safety-related aspects is given in Sect. 3, Modification of Loss.

## 9.2 Multi-task Networks

*Multi-task learning* (MTL) in the context of neural networks describes the process of optimizing several tasks simultaneously by learning a unified feature representation [Car97, RBV18, GHL+20, KBFs20] and coupling the task-specific loss contributions, thereby enforcing cross-task consistency [CPMA19, LYW+19, KTMFs20].

Unified feature representation is usually implemented by sharing the parameters of the initial layers inside the encoder (also called feature extractor). It not only improves the single tasks by more generalized learned features but also reduces the demand for computational resources at inference. Not an entirely new network has to be added for each task but only a task-specific decoder head. It is essential to consider the growing amount of visual perception tasks in automated driving, e.g., depth estimation, semantic segmentation, motion segmentation, and object detection. While the parameter sharing can be soft, as in *cross stitch* [MSGH16] and *sluice networks* [RBAS18], or hard [Kok17, TWZ+18], meaning ultimately sharing the parameters, the latter is usually preferred due to its straightforward implementation and lower computational complexity during training and inference.

Compared to implicitly coupling tasks via a shared feature representation, there are often more direct ways to optimize the tasks inside cross-task losses jointly. It is only made possible as, during MTL, there are network predictions for several tasks, which can be enforced to be consistent. As an example, sharp depth edges should only be at class boundaries of semantic segmentation predictions. Often both approaches to MTL are applied simultaneously [YZS+18, CLLW19] to improve a neural network's performance as well as to reduce its computational complexity at inference.

While the theoretical expectations for MTL are quite clear, it is often challenging to find a good weighting strategy for all the different loss contributions as there is no theoretical basis on which one could choose such a weighting with early approaches either involving heuristics or extensive hyperparameter tuning. The easiest way to balance the tasks is to use uniform weight across all tasks. However, the losses from different tasks usually have different scales, and uniformly averaging them suppresses the gradient from tasks with smaller losses. Addressing these problems, Kendall et al. [KGC18] propose to weigh the loss functions by the *homoscedastic uncertainty* of each task. One does not need to tune the weighting parameters of the loss functions by hand, but they are adapted automatically during the training process. Concurrently Chen et al. [CBLR18] propose *GradNorm*, which does not explicitly weigh the loss functions of different tasks but automatically adapts the gradient magnitudes coming from the task-specific network parts on the backward pass. Liu et al. [LJD19] proposed dynamic weight average (DWA), which uses an average of task losses over time to weigh the task losses.

### 9.3 Neural Architecture Search

In the previous sections, we saw manually engineered modifications of existing CNN architectures proposed by ResNet [HZRS16] or Inception [SLJ+15]. They are results of human design and showed their ability to improve performance. ResNet introduces a *skip connection* in building blocks and Inception makes use of its specific *inception module*. Hereby, the intervention by an expert is crucial. The

approach of *neural architecture search* (NAS) aims to automate this time-consuming and manual design of neural network architectures.

NAS is closely related to hyperparameter optimization (HO), which is described in Sect. 3, hyperparameter optimization. Originally, both tasks were solved simultaneously. Consequently, the kernel size or number of filters were seen as additional hyperparameters. Nowadays, the distinction between HO and NAS should be stressed. The concatenation of complex building blocks or modules cannot be accurately described with single parameters. This simplification is no longer suitable.

To describe the NAS process, the authors of [EMH19b] define three steps: (1) definition of search space, (2) search strategy, and (3) performance estimation strategy.

Majority of search strategies take advantage of the *NASNet search space* [ZVSL18] which arranges various operations, e.g., convolution, pooling within a single cell. However, other spaces based on a chain or multi-branch structure are possible [EMH19b]. The search strategy comprises advanced methods from sequential model-based optimization (SMBO) [LZN+18], Bayesian optimization [KNS+18], evolutionary algorithms [RAHL19, EMH19a], reinforcement learning [ZVSL18, PGZ+18], and gradient descent [LSY19, SDW+19]. Finally, the performance estimation describes approximation techniques due to the impracticability of multiple evaluation runs. For a comprehensive survey regarding the NAS process, we refer to [EMH19b]. Recent research has shown that reinforcement learning approaches such as NASNet-A [ZVSL18] and ENAS [PGZ+18] are partly outperformed by evolutionary algorithms, e.g., AmoebaNet [RAHL19] and gradient-based approaches, e.g., DARTS [LSY19].

Each of these approaches focuses on different optimization aspects. Gradient-based methods are applied to a continuous search space and offer faster optimization. On the contrary, the evolutionary approach LEMONADE [EMH19a] enables multi-object optimization by considering the conjunction of resource consumption and performance as the two main objectives. Furthermore, single-path NAS [SDW+19] extends the multi-path approach of former gradient-based methods and proposes the integration of “over-parameterized superkernels”, which significantly reduces memory consumption.

The focus of NAS is on the optimized combination of humanly predefined CNN elements with respect to objectives such as resource consumption and performance. NAS offers automation, however, the realization of the objectives is strongly limited by the potential of the CNN elements.

## 10 Model Compression

Recent developments in CNNs have resulted in neural networks being the state of the art in computer vision tasks like image classification [KSH12, HZRS15, MGR+18], object detection [Gir15, RDGF15, HGDG17], and semantic segmentation [CPSA17, ZSR+19, LBS+19, WSC+20]. This is largely due to the increasing availability of

hardware computational power and an increasing amount of training data. We also observe a general upward trend of the complexities of the neural networks along with their improvement in state-of-the-art performance. These CNNs are largely trained on back-end servers with significantly higher computing capabilities. The use of these CNNs in real-time applications is inhibited due to the restrictions on hardware, model size, inference time, and energy consumption. This led to an emergence of a new field in machine learning, commonly termed as model compression. Model compression basically implies reducing the memory requirements, inference times, and model size of DNNs to eventually enable the use of neural networks on edge devices. This is tackled by different approaches, such as *network pruning* (identifying weights or filters that are not critical for network performance), *weight quantizations* (reducing the precision of the weights used in the network), *knowledge distillation* (a smaller network is trained with the knowledge gained by a bigger network), and *low-rank factorization* (decomposing a tensor into multiple smaller tensors). In this section, we introduce some of these methods for model compression and discuss in brief the current open challenges and possible research directions with respect to its use in automated driving applications.

In Chapter “[Joint Optimization for DNN Model Compression and Corruption Robustness](#)” [VHB+22], the concept of both pruning and quantization on the application of semantic segmentation in autonomous driving is elucidated, while at the same time robustifying the model against corruptions in the input data. An improved performance and robustness for a state-of-the-art model for semantic segmentation is demonstrated.

## 10.1 Pruning

Pruning has been used as a systematic tool to reduce the complexity of deep neural networks. The redundancy in DNNs may exist on various levels, such as the individual weights, filters, and even layers. All the different methods for pruning try to take advantage of these available redundancies on various levels. Two of the initial approaches for neural networks proposed weight pruning in the 1990s as a way of systematically damaging neural networks [CDS90, Ree93]. As these weight pruning approaches do not aim at changing the structure of the neural network, these approaches are called *unstructured pruning*. Although there is reduction in the size of the network when it is saved in sparse format, the acceleration depends on the availability of hardware that facilitates sparse multiplications. As pruning filters and complete layers aim at exploiting the available redundancy in the architecture or structure of neural networks, these pruning approaches are called *structured pruning*. Pruning approaches can also be broadly classified into: data-dependent and data-independent methods. Data-dependent methods [LLS+17, LWL17, HZS17] make use of the training data to identify filters to prune. Theis et al. [TKTH18] and Molchanov et al. [MTK+17] propose a greedy pruning strategy that identifies the importance of feature maps one at a time from the network and measures the effect

of removal of the filters on the training loss. This means that filters corresponding to those feature maps that have least effect on training loss are removed from the network. Within data-independent methods [LKD+17, HKD+18, YLLW18, ZQF+18], the selection of CNN filters to be pruned are based on the statistics of the filter values. Li et al. [LKD+17] proposed a straightforward method to calculate the rank of filters in a CNN. The selection of filters are based on the  $L_1$ -norm, where the filter with the lowest norm is pruned away. He et al. [HHS17] employ a least absolute shrinkage and selection operator (LASSO) regression-based selection of filters to minimize the least squares reconstruction.

Although the abovementioned approaches demonstrated that a neural network can be compressed without affecting the accuracy, the effect on robustness is largely unstudied. Dhillon et al. [DAL+18] proposed pruning a subset of activations and scaling up the survivors to show improved adversarial robustness of a network. Lin et al. [LGH19] quantize the precision of the weights after controlling the Lipschitz constant of layers. This restricts the error propagation property of adversarial perturbations within the neural network. Ye et al. [YXL+19] evaluated the relationship between adversarial robustness and model compression in detail and show that naive compression has a negative effect on robustness. Gui et al. [GWY+19] co-optimize robustness and compression constraints during the training phase and demonstrate improvement in the robustness along with reduction in the model size. However, these approaches have mostly been tested on image classification tasks and on smaller datasets only. Their effectiveness on safety-relevant automated driving tasks, such as object detection and semantic segmentation, is not studied and remains an open research challenge.

## 10.2 Quantization

Quantization of a random variable  $x$  having a probability density function  $p(x)$  is the process of dividing the range of  $x$  into intervals, each is represented using a single value (also called reconstruction value), such that the following reconstruction error is minimized:

$$\sum_{i=1}^I \int_{b_i}^{b_{i+1}} (q_i - x)^2 p(x) dx, \quad (9)$$

where  $b_i$  is the left-side border of the  $i$ -th interval,  $q_i$  is its reconstruction value, and  $I$  is the number of intervals, e.g.,  $I = 8$  for a 3-bit quantization. This definition can be extended to multiple dimensions as well.

Quantization of neural networks has been around since the 1990s [Guo18], however, with a focus in the early days on improving the hardware implementations of these networks. In the deep learning literature, a remarkable application of quantization combined with unstructured pruning can be found in the approach of *deep compression* [HMD16], where one-dimensional k-means is utilized to cluster the

weights per layer and thus finding the  $I$  cluster centers ( $q_i$  values in (9)) iteratively. This procedure conforms to an implicit assumption that  $p(x)$  has the same spread inside all clusters. Deep compression can reduce the network size needed for image classification by a factor of 35 for AlexNet and a factor of 49 for VGG-16 without any loss in accuracy. However, as pointed out in [JKC+18], these networks from the early deep learning days are over-parameterized and a less impressive compression factor is thus expected when the same technique is applied to lightweight architectures, such as MobileNet and SqueezeNet. For instance, considering SqueezeNet (50 times smaller than AlexNet), the compression factor of deep compression without accuracy loss drops to about 10.

Compared to scalar quantization used in deep compression, there were attempts to exploit the structural information by applying variants of vector quantization of the weights [GLYB14, CEL20, SJG+20]. Remarkably, in the latter (i.e., [SJG+20]), the reconstruction error of the activations (instead of the weights) is minimized in order to find an optimal codebook for the weights, as the ultimate goal of quantization is to approximate the network’s output not the network itself. This is performed in a layer-by-layer fashion (as to prevent error accumulation) using activations generated from unlabeled data.

Other techniques [MDSN17, JKC+18] apply variants of so-called “linear” *quantization*, i.e., the quantization staircase has a fixed interval size. This paradigm conforms to an implicit assumption that  $p(x)$  in (9) is uniform and is thus also called *uniform quantization*. The uniform quantization is widely applied both in specialized software packages, such as Texas Instruments Deep Learning Library (automotive boards) [MDS+18], and in general-purpose libraries, such as Tensorflow Lite. The linearity assumption enables practical implementations, as the quantization and dequantization can be implemented using a scaling factor and an intercept, whereas no codebook needs to be stored. In many situations, the intercept can be omitted by employing a symmetric quantization mapping. Moreover, for power of 2 ranges, the scaling ends up being a bit-wise shift operator, where quantization and dequantization differ only in the shift direction. It is also straightforward to apply this scheme *dynamically*, i.e., for each tensor separately using a tensor-specific multiplicative factor. This can be easily applied not only to filters (weight tensors) but also to activation tensors (see, for instance, [MDSN17]).

Unless the scale factor in the linear quantization is assumed constant by construction, it is computed based on the statistics of the relevant tensor and can be thus sensitive to outliers. This is known to result in a low-precision quantization. In order to mitigate this issue, the original range can be *clipped* and thus reduced to the most relevant part of the signal. Several approaches are proposed in the literature for finding an optimal clipping threshold: simple percentile analysis of the original range (e.g., clipping 2% of the largest magnitude values), minimizing the mean squared error between the quantized and original range in the spirit of (9) [BNS19], or minimizing the Kullback–Leibler divergence between the original and the quantized distributions [Mig17]. While the clipping methods trade off large quantization errors of outliers against small errors of inliers [WJZ+20], other methods tackle the

outliers problem using a different trade-off, see, for instance, the outlier channel splitting approach in [ZHD+19].

An essential point to consider when deciding for a quantization approach for a given problem is the allowed or intended interaction with the training procedure. The so-called *post-training quantization*, i.e., quantization of a pre-trained network, seems to be attractive from a practical point of view: no access to training data is required and the quantization and training toolsets can be independent from each other. On the other hand, the training-aware quantization methods often yield higher inference accuracy and shorter training times. The latter is a serious issue for large complicated models which may need weeks to train on modern GPU clusters. The training-aware quantization can be implemented by inserting fake quantization operators in the computational graph of the forward pass during training (simulated quantization), whereas the backward pass is done as usual in floating-point resolution [JKC+18]. Other approaches [ZWN+18, ZGY+19] go a step further by quantizing the gradients as well. This leads to much lower training time, as the time of the often computationally expensive backward pass is reduced. The gradient’s quantization, however, is not directly applicable as it requires the derivative of the quantization function (staircase-like), which is zero almost everywhere. Luckily, this issue can be handled by employing a *straight-through estimator* [BLC13] (approximating the quantization function by an identity mapping). There are also other techniques proposed recently to mitigate this problem [UMY+20, LM19].

## 11 Discussion

We have presented an extensive overview of approaches to effectively handle safety concerns accompanying deep learning: lack of generalization, robustness, explainability, plausibility, and efficiency. It has been described which lines of research we deem prevalent, important, and promising for each of the individual topics and categories into which the presented methods fall.

The reviewed methods alone will not provide safe-ML systems as such and neither will their future extensions. This is due to the limitations of quantifying complex real-world contexts. A complete and plausible safety argumentation will, thus, require more than advances in methodology and theoretical understanding of neural network properties and training processes. Apart from methodological progress, it will be necessary to gain practical experience in using the presented methods to gather evidence for overall secure behavior, using this evidence to construct a tight safety argument, and testing its validity in various situations.

In particular, each autonomously acting robotic system with state-of-the-art deep-learning-based perception and non-negligible actuation may serve as an object of study and is, in fact, in need of this kind of systematic reasoning before being transferred to widespread use or even market entry. We strongly believe that novel scientific insights, the potential market volume, and public interest will drive the arrival of reliant and trustworthy AI technology.

**Acknowledgements** The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project “Methoden und Maßnahmen zur Absicherung von KI-basierten Wahrnehmungsfunktionen für das automatisierte Fahren (KI Absicherung)”. The authors would like to thank the consortium for the successful cooperation.

## References

- [AAAB18] S. Akcay, A. Atapour-Abarghouei, T.P. Breckon, Gandomaly: semi-supervised anomaly detection via adversarial training, in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, Perth, WA, Australia (2018), pp. 622–637
- [ABC+18] Y. Adi, C. Baum, M. Cisse, B. Pinkas, J. Keshet, Turning your weakness into a strength: watermarking deep neural networks by backdooring, in *Proceedings of the USENIX Security Symposium*, Baltimore, MD, USA (2018), pp. 1615–1631
- [ACKK14] S. Amershi, M. Cakmak, W.B. Knox, T. Kulesza, Power to the people: the role of humans in interactive machine learning. *AI Mag.* **35**(4), 105–120 (2014)
- [ACÖG17] M. Ancona, E. Ceolini, C. Öztïreli, M.H. Gross, A unified view of gradient-based attribution methods for deep neural networks (2017), pp. 1–11. [arXiv:1711.06104](https://arxiv.org/abs/1711.06104)
- [ACS19] M. Angus, K. Czarnecki, R. Salay, Efficacy of pixel-level OOD detection for semantic segmentation (2019), pp. 1–13. [arXiv:1911.02897](https://arxiv.org/abs/1911.02897)
- [ACW18] A. Athalye, N. Carlini, D. Wagner, Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden (2018), pp. 274–283
- [AEIK18] A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, Synthesizing robust adversarial examples, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden (2018), pp. 284–293
- [AGM+20] J. Adebayo, J. Gilmer, M. Muell, I.J. Goodfellow, M. Hardt, B. Kim, Sanity checks for saliency maps (2020), pp. 1–30. [arXiv:1810.03292](https://arxiv.org/abs/1810.03292)
- [AJD18] S. Arik, H. Jun, G. Diamos, Fast spectrogram inversion using multi-head convolutional neural networks. *IEEE Signal Proc. Lett.* **26**(1), 94–98 (2018)
- [AK12] M.G. Augasta, T. Kathirvalavakumar, Rule extraction from neural networks – a comparative study, in *Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME)*, Salem, India (2012), pp. 404–408
- [AW19] A. Azulay, Y. Weiss, Why do deep convolutional networks generalize so poorly to small image transformations? (2019), pp. 1–25. [arXiv:1805.12177](https://arxiv.org/abs/1805.12177)
- [BBL+07] M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, R.J. Plemmons, Algorithms and applications for approximate nonnegative matrix factorization. *Comput. Stat. & Data Anal.* **52**(1), 155–173 (2007)
- [BBLFs19] J.-A. Bolte, A. Bär, D. Lipinski, T. Fingscheidt, Towards corner case detection for autonomous driving, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Paris, France (2019), pp. 438–445
- [BBM+15] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On Pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE* **10**(7), 1–46 (2015)
- [BCG+18] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R.N. Gunn, A. Hammers, D.A. Dickie, M. del C. Valdés Hernández, J.M. Wardlaw, D. Rueckert, GAN augmentation: augmenting training data using generative adversarial networks (2018), pp. 1–12. [arXiv:1810.10863](https://arxiv.org/abs/1810.10863)
- [BCK+07] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. Wortman, Learning bounds for domain adaptation, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2007), pp. 129–136

- [BCKW15] C. Blundell, J. Cornebise, K. Kavukcuoglu, D. Wierstra, Weight uncertainty in neural networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France (2015), pp. 1613–1622
- [BCV13] Y. Bengio, A. Courville, P. Vincent, R. Learning, A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **35**(8), 1798–1828 (2013)
- [BDBC+10] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. Vaughan, A theory of learning from different domains. *Mach. Learn.* **79**(1), 151–175 (2010)
- [BF17] S. Baluja, I. Fischer, Adversarial transformation networks: learning to generate adversarial examples (2017), pp. 1–13. [arXiv:1703.09387](https://arxiv.org/abs/1703.09387)
- [BFS18] A. Bhattacharyya, M. Fritz, B. Schiele, Long-term on-board prediction of people in traffic scenes under uncertainty, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 4194–4202
- [BFW+19] Y. Bai, Y. Feng, Y. Wang, T. Dai, S.-T. Xia, Y. Jiang, Hilbert-based generative defense for adversarial examples, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea (2019), pp. 4784–4793
- [BGR+06] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.-P. Kriegel, B. Schölkopf, A.J. Smola, Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* **22**(14), 49–57 (2006)
- [BGS+19] S. Burton, L. Gauerhof, B.B. Sethy, I. Habli, R. Hawkins, Confidence arguments for evidence of performance in machine learning for highly automated driving functions, in *Proceedings of the International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*, Toulouse, France (2019), pp. 365–377
- [BHG+19] A. Boloor, X. He, C. Gill, Y. Vorobeychik, X. Zhang, Simple physical adversarial examples against end-to-end autonomous driving models, in *Proceedings of the IEEE International Conference on Embedded Software and Systems (ICESS)*, Las Vegas, NV, USA (2019), pp. 1–7
- [BHH+22] S. Burton, C. Hellert, F. Hüger, M. Mock, A. Rohatschek, Safety assurance of machine learning for perception functions, in *Deep Neural Networks and Data for Automated Driving – Robustness, Uncertainty Quantification, and Insights Towards Safety*, eds. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 365–387
- [BHLHL+16] T. Bui, D. Hernández-Lobato, J. Hernandez-Lobato, Y. Li, R. Turner, Deep gaussian processes for regression using approximate expectation propagation, in *Proceedings of the International Conference on Machine Learning (ICML)*, New York, NY, USA (2016), pp. 1472–1481
- [BHP+18] C.P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding Disentangling in  $\beta$ -VAE (2018), pp. 1–11. [arXiv:1804.03599](https://arxiv.org/abs/1804.03599)
- [BHSFs19] A. Bär, F. Hüger, P. Schlücht, T. Fingscheidt, On the robustness of redundant teacher-student frameworks for semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA (2019), pp. 1380–1388
- [Bis06] C.M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Berlin, 2006)
- [BKD19] C.A. Brust, C. Käding, J. Denzler, Active learning for deep object detection, in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, Prague, Czech Republic (2019), pp. 181–190
- [BKOŠ18] P. Bevandić, I. Krešo, M. Oršić, S. Šegvić, Discriminative out-of-distribution detection for semantic segmentation (2018), pp. 1–18. [arXiv:1808.07703](https://arxiv.org/abs/1808.07703)
- [BKV+20] A. Bär, M. Klingner, S. Varghese, F. Hüger, P. Schlücht, T. Fingscheidt, Robust semantic segmentation by redundant networks with a layer-specific loss contribution and majority vote, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2020), pp. 1348–1358, Virtual Conference

- [BLC13] Y. Bengio, N. Léonard, A.C. Courville, Estimating or propagating gradients through stochastic neurons for conditional computation (2013), pp. 1–12. [arXiv:1308.3432](https://arxiv.org/abs/1308.3432)
- [BMM18] G. Bagschik, T. Menzel, M. Maurer, Ontology based scene creation for the development of automated vehicles, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China (2018), pp. 1813–1820
- [BMR+17] T.B. Brown, D. Mané, A. Roy, M. Abadi, J. Gilmer, Adversarial patch, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS) Workshops*, Long Beach, CA, USA (2017), pp. 1–6
- [BNS19] R. Banner, Y. Nahshan, D. Soudry, Post training 4-bit quantization of convolutional networks for rapid-deployment, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2019), pp. 7948–7956
- [BRW18] C. Bunne, L. Rahmann, T. Wolf, Studying invariances of trained convolutional neural networks (2018), pp. 1–7. [arXiv:1803.05963](https://arxiv.org/abs/1803.05963)
- [BSC18] Y. Balaji, S. Sankaranarayanan, R. Chellappa, MetaReg: towards domain generalization using meta-regularization, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2018), pp. 1006–1016
- [BTLFs20] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, T. Fingscheidt, Systematization of corner cases for visual perception in automated driving, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)* (2020), pp. 1257–1264. Virtual Conference
- [BXS+20] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J.M.F. Moura, P. Eckersley, Explainable machine learning in deployment, in *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT/FAT\*)*, Barcelona, Spain (2020), pp. 648–657
- [C+15] F. Chollet et al., Keras (2015). Accessed 18 Nov 2021
- [Car97] R. Caruana, Multitask learning. *Mach. Learn.* **28**(1), 41–75 (1997)
- [CBG+17] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, N. Usunier, Parseval networks: improving robustness to adversarial examples, in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia (2017), pp. 854–863
- [CBLR18] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, GradNorm: gradient normalization for adaptive loss balancing in deep multitask networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden (2018), pp. 794–803
- [CBSW19] L. Caltagirone, M. Bellone, L. Svensson, M. Wahde, LiDAR-camera fusion for road detection using fully convolutional neural networks. *Robot. Auton. Syst.* **111**, 125–131 (2019)
- [CCO98] A.I. Cristea, P.D. Cristea, T. Okamoto, N.N.K. Extraction, *Revue Roumaine des Sciences Techniques. Série Électrotechnique et Énergétique* **43**(1), 1–14 (1998)
- [CDH+16] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, InfoGAN: interpretable representation learning by information maximizing generative adversarial nets, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Barcelona, Spain (2016), pp. 2172–2180
- [CDS90] Y.L. Cun, J.S. Denker, S.A. Solla, Optimal brain damage, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Denver, CO, USA (1990), pp. 598–605
- [CE17] R. Cotterell, J. Eisner, Probabilistic typology: deep generative models of vowel inventories (2017), pp. 1–11. [arXiv:1705.01684](https://arxiv.org/abs/1705.01684)
- [CEL20] Y. Choi, M. El-Khamy, J. Lee, Universal deep neural network compression. *IEEE J. Select. Topics Signal Proc.* **14**(4), 715–726 (2020)
- [CH19] J. Clark, G.K. Hadfield, Regulatory markets for AI safety (2019), pp. 1–23. [arXiv:2001.00078](https://arxiv.org/abs/2001.00078)

- [CHM+15] A. Choromanska, M. Henaff, M. Mathieu, G.B. Arous, Y. LeCun, The loss surfaces of multilayer networks, in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, San Diego, CA, USA (2015), pp. 192–204
- [CL18] J. Choo, S. Liu, Visual analytics for explainable deep learning (2018), pp. 1–10. [arXiv:1804.02527](https://arxiv.org/abs/1804.02527)
- [CLC+19] H.-Y. Chen, J.-H. Liang, S.-C. Chang, J.-Y. Pan, Y.-T. Chen, W. Wei, D.-C. Juan, Improving adversarial robustness via guided complement entropy, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea (2019), pp. 4881–4889
- [CLG+15] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, N. Elhadad, Intelligible models for healthcare: predicting pneumonia risk and hospital 30-day readmission, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Sydney, NSW, Australia (2015), pp. 1721–1730
- [CLLW19] P.-Y. Chen, A.H. Liu, Y.-C. Liu, Y.-C.F. Wang, Towards scene understanding: unsupervised monocular depth estimation with semantic-aware representation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 2624–2632
- [CLS+18] Y. Chen, W. Li, C. Sakaridis, D. Dai, L. Van Gool, Domain adaptive faster R-CNN for object detection in the wild, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 3339–3348
- [CNH+18] C.-H. Cheng, G. Nührenberg, C.-H. Huang, H. Ruess, H. Yasuoka, Towards dependency metrics for neural networks, in *Proceedings of the ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE)*, Beijing, China (2018), pp. 43–46
- [COR+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The Cityscapes dataset for semantic urban scene understanding, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA (2016), pp. 3213–3223
- [CPK+18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **40**(4), 834–848 (2018)
- [CPMA19] V. Casser, S. Pirk, R. Mahjourian, A. Angelova, Depth prediction without the sensors: leveraging structure for unsupervised learning from monocular videos, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA (2019), pp. 8001–8008
- [CPSA17] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking Atrous convolution for semantic image segmentation (2017), pp. 1–14. [arXiv:1706.05587](https://arxiv.org/abs/1706.05587)
- [Cs17] G. Csurka, Domain adaptation for visual applications: a comprehensive survey, in *Domain Adaptation in Computer Vision Applications*, ed. by G. Csurka (Springer, Berlin, 2017), pp. 1–35
- [CURG22] R. Chan, S. Uhlemeyer, M. Rottmann, H. Gottschalk, Detecting and learning the unknown in semantic segmentation, in *Deep Neural Networks and Data for Automated Driving – Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 307–344
- [CW17a] N. Carlini, D. Wagner, Adversarial examples are not easily detected: bypassing ten detection methods, in *Proceedings of the ACM Workshop on Artificial Intelligence and Security (AISec)*, New York, NY, USA (2017), pp. 3–14
- [CW17b] N. Carlini, D.A. Wagner, Towards evaluating the robustness of neural networks, in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA (2017), pp. 39–57

- [CWL+19] H.-Y. Chen, P.-H. Wang, C.-H. Liu, S.-C. Chang, J.-Y. Pan, Y.-T. Chen, W. Wei, D.-C. Juan, Complement objective training, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA (2019), pp. 1–10
- [CZM+19] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q.V. Le, AutoAugment: learning augmentation strategies from data, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 113–123
- [CZSL19] E.D. Cubuk, B. Zoph, J. Shlens, Q.V. Le, RandAugment: practical data augmentation with no separate search (2019), pp. 1–13. [arXiv:1909.13719](https://arxiv.org/abs/1909.13719)
- [CZYS19] F. Cheng, H. Zhang, D. Yuan, M. Sun, Leveraging semantic segmentation with learning-based confidence measure. *Neurocomputing* **329**, 21–31 (2019)
- [DAL+18] G. Dhillon, K. Azizzadenesheli, Z. Lipton, J. Bernstein, J. Kossaifi, A. Khanna, A. Anandkumar, Stochastic activation pruning for robust adversarial defense, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada (2018), pp. 1–13
- [DCG+19] S.V. Desai, A.L. Chandra, W. Guo, S. Ninomiya, V.N. Balasubramanian, An adaptive supervision framework for active learning in object detection, in *Proceedings of the British Machine Vision Conference (BMVC)*, Cardiff, UK (2019), pp. 1–13
- [DCO+19] Q. Dou, C. Chen, C. Ouyang, H. Chen, P.A. Heng, Unsupervised domain adaptation of ConvNets for medical image segmentation via adversarial learning, in *Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics*, ed. by L. Le, X. Wang, G. Carneiro, L. Yang (Springer, Berlin, 2019), pp. 93–115
- [DdCKG19] Q. Dou, D.C. de Castro, K. Kamnitsas, B. Glocker, Domain generalization via model-agnostic learning of semantic features, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2019), pp. 6447–6458
- [DK17] J. Djolonga, A. Krause, Learning implicit generative models using differentiable graph tests (2017), pp. 1–16. [arXiv:1709.01006](https://arxiv.org/abs/1709.01006)
- [DK18] J.J. Dudley, P.O. Kristensson, A review of user interface design for interactive machine learning. *ACM Trans. Inter. Intell. Syst. (TIIS)* **8**(2), 1–37 (2018)
- [DL13] A. Damianou, N. Lawrence, Deep Gaussian processes, in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Scottsdale, AZ, USA (2013), pp. 207–215
- [DLP+18] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 9185–9193
- [DMW+20] R. Duan, X. Ma, Y. Wang, J. Bailey, A.K. Qin, Y. Yang, Adversarial camouflage: hiding physical-world attacks with natural styles, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual Conference (2020), pp. 1000–1008
- [Doe16] C. Doersch, Tutorial on variational autoencoders (2016), pp. 1–23. [arXiv:1606.05908](https://arxiv.org/abs/1606.05908)
- [DRC+17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: an open urban driving simulator, in *Proceedings of the Conference on Robot Learning CORL*, Mountain View, CA, USA (2017), pp. 1–16
- [DT17] T. DeVries, G.W. Taylor, Dataset augmentation in feature space, in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, Toulon, France (2017), pp. 1–12
- [DT18] T. DeVries, G.W. Taylor, Learning confidence for out-of-distribution detection in neural networks (2018), pp. 1–12. [arXiv:1802.04865](https://arxiv.org/abs/1802.04865)

- [EEF+18] K. Erykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, Robust physical-world attacks on deep learning visual classification, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 1625–1634
- [EIS+19] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, A. Madry, Adversarial robustness as a prior for learned representations (2019), pp. 1–25. [arXiv:1906.00945](https://arxiv.org/abs/1906.00945)
- [EMH19a] T. Elsken, J.H. Metzen, F. Hutter, Efficient multi-objective neural architecture search via lamarckian evolution, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA (2019), pp. 1–23
- [EMH19b] T. Elsken, J.H. Metzen, F. Hutter, Neural architecture search: a survey. *J. Mach. Learn. Res.* **20**(55), 1–21 (2019)
- [ERT+17] A. Endert, W. Ribarsky, C. Turkay, W. Wong, I. Nabney, I. Díaz Blanco, F. Rossi, The state of the art in integrating machine learning into visual analytics. *Comput. Graph. Forum* **36**(8), 458–486 (2017)
- [ETTS19] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, A rotation and a translation suffice: fooling CNNs with simple transformations, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA (2019), pp. 1–21
- [EW18] C. Eastwood, C.K.I. Williams, A framework for the quantitative evaluation of disentangled representations, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada (2018), pp. 1–15
- [FAA18] E. Fertig, A. Arbab, A.A. Alemi,  $\beta$ -VAEs can retain label information even at high compression (2018), pp. 1–6. [arXiv:1812.02682](https://arxiv.org/abs/1812.02682)
- [FAL17] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia (2017), pp. 1126–1135
- [FCSG17] R. Feinman, R.R. Curtin, S. Shintre, A.B. Gardner, Detecting adversarial samples from artifacts (2017), pp. 1–9. [arXiv:1703.00410](https://arxiv.org/abs/1703.00410)
- [FF15] A. Fawzi, P. Frossard, Manitest: are classifiers really invariant? (2015), pp. 1–13. [arXiv:1507.06535](https://arxiv.org/abs/1507.06535)
- [FH19] M. Feurer, F. Hutter, Hyperparameter optimization, in *Automated Machine Learning: Methods, Systems, Challenges* (Springer, Berlin, 2019), pp. 3–33
- [FKH18] S. Falkner, A. Klein, F. Hutter, BOHB: robust and efficient hyperparameter optimization at scale, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden (2018), pp. 1436–1445
- [FP78] D.F. Frey, R.A. Pimentel, Principal component analysis and factor analysis, in *Quantitative Ethology*, ed. by P.W. Colgan (Wiley, New York, 1978), pp. 219–245
- [Fry77] M.J. Fryer, A review of some non-parametric methods of density estimation. *J. Inst. Math. Appl.* **20**(3), 335–354 (1977)
- [FSS+16] M. Fathy, M.H. Saffar, M. Sabokrou, M. Fathy, R. Klette, F. Huang, STFCN: Spatio-temporal FCN for semantic video segmentation (2016), pp. 1–17. [arXiv:1608.05971](https://arxiv.org/abs/1608.05971)
- [FV17] R.C. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy (2017), pp. 3429–3437
- [FXY12] J. Feng, H. Xu, S. Yan, Robust PCA in high-dimension: a deterministic approach (2012), pp. 1–8. [arXiv:1206.4628](https://arxiv.org/abs/1206.4628)
- [FZ10] P.F. Felzenszwalb, R. Zabih, Dynamic programming and graph algorithms in computer vision. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **33**(4), 721–740 (2010)
- [GAHW22] S.S. Gannamaneni, M. Akila, C. Heinemann, M. Woehrle, The good and the bad: using neuron coverage as a DNN validation technique, in *Deep Neural Networks and Data for Automated Driving – Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 413–433

- [GBA+19] Y.F.A. Gaus, N. Bhowmik, S. Akçay, P.M. Guillén-Garcia, J.W. Barker, T.P. Breckon, Evaluation of a dual convolutional neural network architecture for object-wise anomaly detection in cluttered x-ray security imagery, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary (2019), pp. 1–8
- [GBR+06] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A.J. Smola, A kernel method for the two-sample-problem, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2006), pp. 513–520
- [GBR07] T. Gneiting, F. Balabdaoui, A.E. Raftery, P. Forecasts, Calibration and sharpness. *J. R. Stat. Soc.: Ser. B (Stat. Methodol.)* **69**(2), 243–268 (2007)
- [GBR+12] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A.J. Smola, A kernel two-sample test. *J. Mach. Learn. Res.* **13**(25), 723–773 (2012)
- [GBY+18] L.H. Gilpin, D. Bau, B.Z. Yuan, A. Bajwa, M. Specter, L. Kagel, Explaining explanations: an overview of interpretability of machine learning, in *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Turin, Italy (2018), pp. 80–89
- [GEB15] L.A. Gatys, A.S. Ecker, M. Bethge, A neural algorithm of artistic style (2015), pp. 1–16. [arXiv:1508.06576](https://arxiv.org/abs/1508.06576)
- [GEB16] L.A. Gatys, A.S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA (2016), pp. 2414–2423
- [GG84] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **6**(6), 721–741 (1984)
- [GG16] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: representing model uncertainty in deep learning, in *Proceedings of the International Conference on Machine Learning (ICML)*, New York, NY, USA (2016), pp. 1050–1059
- [GHHW20] C. Gladisch, C. Heinemann, M. Herrmann, M. Woehrle, Leveraging combinatorial testing for safety-critical computer vision datasets, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2020), pp. 1314–1321. Virtual Conference
- [GHK17] Y. Gal, J. Hron, A. Kendall, Concrete dropout, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Long Beach, CA, USA (2017), pp. 3581–3590
- [GHL+20] V. Guizilini, R. Hou, J. Li, R. Ambrus, A. Gaidon, Semantically-guided representation learning for self-supervised monocular depth, in *Proceedings of the International Conference on Learning Representations (ICLR)* (2020), pp. 1–14. Virtual Conference
- [GHS22] O. Grau, K. Hagn, Q.S. Sha, A variational deep synthesis approach for perception validation, in *Deep Neural Networks and Data for Automated Driving – Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 389–412
- [GIG17] Y. Gal, R. Islam, Z. Ghahramani, Deep Bayesian active learning with image data, in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia (2017), pp. 1183–1192
- [Gir15] R. Girshick, Fast R-CNN, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile (2015), pp. 1440–1448
- [GJG17] R. Gadde, V. Jampani, P.V. Gehler, Semantic video CNNs through representation warping, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy (2017), pp. 4453–4462
- [GJZ+18] J. Guo, Y. Jiang, Y. Zhao, Q. Chen, J. Sun, DLFuzz: differential fuzzing testing of deep learning systems, in *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on Foundations of Software Engineering (ESEC/FSE)*, Lake Buena Vista, FL, USA (2018), pp. 739–743

- [GK20] T. Ghosh, M. Kirby, Supervised dimensionality reduction and visualization using centroid-encoder (2020), pp. 1–25. [arXiv:2002.11934](https://arxiv.org/abs/2002.11934)
- [GKZ14] M. Ghifary, W.B. Kleijn, M. Zhang, Domain adaptive neural networks for object recognition, in *Proceedings of the Pacific Rim International Conference on Artificial Intelligence (PRICAI) – Trends in Artificial Intelligence*, Gold Coast, QLD, Australia (2014), pp. 898–904
- [GKZB15] M. Ghifary, W.B. Kleijn, M. Zhang, D. Balduzzi, Domain generalization for object recognition with multi-task autoencoders, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile (2015), pp. 2551–2559
- [GLYB14] Y. Gong, L. Liu, M. Yang, L.D. Bourdev, Compressing deep convolutional networks using vector quantization (2014), pp. 1–10. [arXiv:1412.6115](https://arxiv.org/abs/1412.6115)
- [Goo17] I. Goodfellow, NIPS 2016 tutorial: generative adversarial networks (2017), pp. 1–57. [arXiv:1701.00160](https://arxiv.org/abs/1701.00160)
- [GPAM+14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (2014), pp. 2672–2680
- [GPSW17] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia (2017), pp. 1321–1330
- [GR19] P. Gupta, E. Rahtu, CIIDefence: defeating adversarial attacks by fusing class-specific image inpainting and image denoising, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea (2019), pp. 6708–6717
- [Gra18] A. Gramacki, *Nonparametric Kernel Density Estimation and its Computational Aspects* (Springer, Berlin, 2018)
- [GRCvdM18] C. Guo, M. Rana, M. Cissé, L. van der Maaten, Countering adversarial images using input transformations, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada (2018), pp. 1–12
- [GRS22] H. Gottschalk, M. Rottmann, M. Saltagic, Does redundancy in AI perception systems help to test for super-human automated driving performance?, in *Deep Neural Networks and Data for Automated Driving—Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 103–128
- [GSS15] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA (2015), pp. 1–11
- [GTC+18] R. Garcia, A.C. Telea, B.C. da Silva, J. Tørresen, J.L. Dihl Comba, A task-and-technique centered survey on visual analytics for deep learning model engineering. *Comput. & Graph.* **77**, 30–49 (2018)
- [Guo18] Y. Guo, A survey on methods and theories of quantized neural networks (2018), pp. 1–17. [arXiv:1808.04752](https://arxiv.org/abs/1808.04752)
- [GWY+19] S. Gui, H. Wang, H. Yang, C. Yu, Z. Wang, J. Liu, Model compression with adversarial robustness: a unified optimization framework, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2019), pp. 1283–1294
- [HAB19] M. Hein, M. Andriushchenko, J. Bitterwolf, Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 41–50
- [Hai16] T. Hailesilassie, Rule extraction algorithm for deep neural networks: a review (2016), pp. 1–6. [arXiv:1610.05267](https://arxiv.org/abs/1610.05267)
- [HAMFs22] A.S. Hashemi, B. Andreas, S. Mozaffari, T. Fingscheidt, Improving transferability of generated universal adversarial perturbations for image classification and segmentation, in *Deep Neural Networks and Data for Automated Driving—Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 195–222

- [HAMS20] T.M. Hospedales, A. Antoniou, P. Micaelli, A.J. Storkey, Meta-learning in neural networks: a survey (2020), pp. 1–20. [arXiv:2004.05439](https://arxiv.org/abs/2004.05439)
- [HD19] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA (2019), pp. 1–15
- [HDHH20] J. Hanhriova, A. Debner, M. Hyppä, V. Hirvisalo, A machine learning environment for evaluating autonomous driving software (2020), pp. 1–8. [arXiv:2003.03576](https://arxiv.org/abs/2003.03576)
- [HDR18] M. Harradon, J. Druce, B. Ruttenberg, Causal learning and explanation of deep neural networks via autoencoded activations (2018), pp. 1–8. [arXiv:1802.00541](https://arxiv.org/abs/1802.00541)
- [HDY+12] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Proc. Mag.* **29**(6), 82–97 (2012)
- [HG17] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France (2017), pp. 1–12
- [HG22] K. Hagn, O. Grau, Optimized data synthesis for DNN training and validation by sensor artifact simulation, in *Deep Neural Networks and Data for Automated Driving—Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 149–170
- [HGDG17] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy (2017), pp. 2980–2988
- [Hin09] G.E. Hinton, Deep belief networks. *Scholarpedia* **4**(5), 5947 (2009)
- [HKD+18] Y. He, G. Kang, X. Dong, Y. Fu, Y. Yang, Soft filter pruning for accelerating deep convolutional neural networks, in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, Stockholm, Sweden (2018), pp. 2234–2240
- [HKPC18] F. Hohman, M. Kahng, R. Pienta, D.H. Chau, Visual analytics in deep learning: an interrogative survey for the next frontiers (2018), pp. 1–20. [arXiv:1801.06889](https://arxiv.org/abs/1801.06889)
- [HLP+17] G. Huang, Y. Li, G. Pleiss, Z. Liu, J.E. Hopcroft, K.Q. Weinberger, Snapshot ensembles: train 1, get M for free (2017), pp. 1–14. [arXiv:1704.00109](https://arxiv.org/abs/1704.00109)
- [HLS+19] D. Ho, E. Liang, I. Stoica, P. Abbeel, X. Chen, Population based augmentation: efficient learning of augmentation policy schedules, in *Proceedings of the International Conference on Machine Learning (ICML)*, Long Beach, CA, USA (2019), pp. 2731–2741
- [HMD16] S. Han, H. Mao, W.J. Dally, Deep compression: compressing deep neural network with pruning, trained quantization and huffman coding, in *Proceedings of the International Conference on Learning Representations (ICLR)* (2016), pp. 1–14
- [HMD19] D. Hendrycks, M. Mazeika, T. Dietterich, Deep anomaly detection with outlier exposure, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA (2019), pp. 1–18
- [HMJ13] T. Hazan, S. Maji, T. Jaakkola, On sampling from the Gibbs distribution with random maximum a-posteriori perturbations, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Lake Tahoe, NV, USA (2013), pp. 1268–1276
- [HMK+19] L. Hoyer, M. Muñoz, P. Katiyar, A. Khoreva, V. Fischer, Grid saliency for context explanations of semantic segmentation, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2019), pp. 6459–6470
- [HMP+17] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M.M. Botvinick, S. Mohamed, A. Lerchner,  $\beta$ -VAE: learning basic visual concepts with a constrained variational framework, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France (2017), pp. 1–22

- [HO00] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications. *Neural Netw.* **13**(4–5), 411–430 (2000). (June)
- [HR90] J. Huber, A. Rüppel, Zuverlässigkeitsschätzung für die Ausgangssymbole von Trellis-Decodern. *Archiv für Elektronik und Übertragung (AEÜ)* (in German), **44**(1), 8–21 (1990)
- [HRF19] Z. He, A.S. Rakim, D. Fan, Parametric noise injection: trainable randomness to improve deep neural network robustness against adversarial attack, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 588–597
- [HS97] S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
- [HVD14] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS) Workshops*, Montréal, QC, Canada (2014), pp. 1–9
- [HZRS15] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **37**(9), 1904–1916 (2015)
- [HZRS16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA (2016), pp. 770–778
- [HZS17] Y. He, X. Zhang, J. Sun, Channel pruning for accelerating very deep neural networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy (2017), pp. 1398–1406
- [ICG+18] E. Ilg, O. Cicek, S. Galessio, A. Klein, O. Makansi, F. Hutter, T. Brox, Uncertainty estimates and multi-hypotheses networks for optical flow, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany (2018), pp. 652–667
- [Ino18] H. Inoue, Data augmentation by pairing samples for images classification (2018), pp. 1–8. [arXiv:1801.02929](https://arxiv.org/abs/1801.02929)
- [ISO18] ISO/TC 22/SC 32. ISO 26262-1:Road Vehicles – Functional Safety – Part 1: Vocabulary. International Organization for Standardization (ISO) (2018)
- [JC16] I.T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments. *Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci.* **374**(2065), 1–16 (2016)
- [JC17] K. Janocha, W.M. Czarnecki, On loss functions for deep neural networks in classification. *Schedae Informaticae* **25**(9), 49–49 (2017)
- [JDCR12] M. Joshi, M. Dredze, W.W. Cohen, C.P. Rosé, Multi-domain learning: when do domains matter?, in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Jeju Island, Korea (2012), pp. 1302–1312
- [JGST19] H.S. Jomaa, J. Grabocka, L. Schmidt-Thieme, Hyp-RL: hyperparameter optimization by reinforcement learning (2019), pp. 1–17. [arXiv:1906.11527](https://arxiv.org/abs/1906.11527)
- [JKC+18] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A.G. Howard, H. Adam, D. Kalenichenko, Quantization and training of neural networks for efficient integer-arithmetic-only inference, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 2704–2713
- [JLM+18] B. Jiang, R. Luo, J. Mao, T. Xiao, Y. Jiang, Acquisition of localization confidence for accurate object detection, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany (2018), pp. 784–799
- [JLS+20] Y. Jia, Y. Lu, J. Shen, Q.A. Chen, H. Chen, Z. Zhong, T. Wei, Fooling detection alone is not enough: adversarial attack against multiple object tracking, in *Proceedings of the International Conference on Learning Representations (ICLR)* (2020), pp. 1–11. Virtual Conference

- [JMS96] M.C. Jones, J.S. Marron, S.J. Sheather, A brief survey of bandwidth selection for density estimation. *J. Amer. Stat. Assoc.* **91**(433), 401–407 (1996)
- [JWCF19] X. Jia, X. Wei, X. Cao, H. Foroosh, ComDefend: an efficient image compression model to defend adversarial examples, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 6084–6092
- [KAF+08] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, G. Melançon, Visual analytics: definition, process, and challenges, in *Information Visualization: Human-Centered Issues and Perspectives* (Springer, Berlin, 2008), pp. 154–175
- [KALL18] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of GANs for improved quality, stability, and variation, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada (2018), pp. 1–26
- [KB15] D.P. Kingma, J. Ba, ADAM: a method for stochastic optimization, in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA (2015), pp. 1–15
- [KBFs20] M. Klingner, A. Bär, T. Fingscheidt, Improved noise and attack robustness for semantic segmentation by using multi-task training with self-supervised depth estimation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2020), pp. 1299–1309. Virtual Conference
- [KBL+17] K. Kamnitsas, C. Baumgartner, C. Ledig, V. Newcombe, J. Simpson, A. Kane, D. Menon, A. Nori, A. Criminisi, D. Rueckert, B. Glocker, Unsupervised domain adaptation in brain lesion segmentation with adversarial networks, in *Proceedings of the International Conference on Information Processing in Medical Imaging (IPMI)*, Boone, NC, USA (2017), pp. 597–609
- [KFE18] V. Kuleshov, N. Fenner, S. Ermon, Accurate uncertainties for deep learning using calibrated regression, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden (2018), pp. 2801–2809
- [KFs22] M. Klingner, T. Fingscheidt, Improved DNN robustness by multi-task training with an auxiliary self-supervised task, in *Deep Neural Networks and Data for Automated Driving—Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 171–194
- [KG17] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Long Beach, CA, USA, 2017), pp. 5574–5584
- [KGB17a] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops* (Toulon, France 2017), pp. 1–14
- [KGB17b] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Toulon, France, 2017), pp. 1–17
- [KGC17] J. Kukačka, V. Golkov, D. Cremers, Regularization for deep learning: a taxonomy (2017), pp. 1–23. [arXiv:1710.10686](https://arxiv.org/abs/1710.10686)
- [KGC18] A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 7482–7491
- [KGLL20] Z. Kong, J. Guo, A. Li, C. Liu, PhysGAN: generating physical-world-resilient adversarial examples for autonomous driving, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 14254–14263. Virtual Conference
- [KHKS22] F. Küppers, A. Haselhoff, J. Kronenberger, J. Schneider, Confidence calibration for object detection and segmentation, in *Deep Neural Networks and Data for Automated Driving—Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 255–282

- [KK11] P. Krähenbühl, V. Koltun, Efficient inference in fully connected CRFs with Gaussian edge potentials, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Granada, Spain (2011), pp. 109–117
- [KKB19] P. Koopman, A. Kane, J. Black, Credible autonomy safety argumentation, in *Proceedings of the Safety-Critical Systems Symposium (SSS)*, Bristol, UK (2019), pp. 1–27
- [KKMH20] I. Khemakhem, D.P. Kingma, R.P. Monti, A. Hyvärinen, Variational autoencoders and nonlinear ICA: a unifying framework (2020), pp. 1–27. [arXiv:1907.04809](https://arxiv.org/abs/1907.04809)
- [KKSH20] F. Küppers, J. Kronenberger, A. Shantia, A. Haselhoff, Multivariate confidence calibration for object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2020), pp. 1322–1330. Virtual Conference
- [KLSL18] C.C. Kao, T.Y. Lee, P. Sen, M.Y. Liu, Localization-aware active learning for object detection (2018), pp. 1–35. [arXiv:1801.05124](https://arxiv.org/abs/1801.05124)
- [KLX+17] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, X. Wang, Object detection in videos with tubelet proposal networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA (2017), pp. 727–735
- [KM02] D. Klein, C.D. Manning, Fast exact inference with a factored model for natural language parsing, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2002), pp. 3–10
- [KMAM18] M. Kesarwani, B. Mukhoty, V. Arya, S. Mehta, Model extraction warning in MLaaS paradigm, in *Proceedings of the Annual Computer Security Applications Conference* (2018), pp. 371–380
- [KMN+17] N.S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P.T.P. Tang, On large-batch training for deep learning: generalization gap and sharp minima (2017), pp. 1–16. [arXiv:1609.04836](https://arxiv.org/abs/1609.04836)
- [KMT09] D.A. Keim, F. Mansmann, J. Thomas, Visual analytics: how much visualization and how much analytics? *ACM SIGKDD Explor. Newslett.* **11**(2), 5–8 (2009)
- [KNS+18] K. Kandasamy, W. Neiswanger, J. Schneider, B. Póczos, E.P. Xing, Neural architecture search with Bayesian optimisation and optimal transport, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2018), pp. 2020–2029
- [Kok17] I. Kokkinos, Ubernet: training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA (2017), pp. 5454–5463
- [KP20] A. Kumar, B. Poole, On implicit regularization in  $\beta$ -VAEs, in *Proceedings of the International Conference on Machine Learning (ICML)* (2020), pp. 5480–5490
- [KRBT08] P. Kohli, J. Rihan, M. Bray, P.H.S. Torr, Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *Int. J. Comput. Vis. (IJCV)* **79**(3), 285–298 (2008)
- [Kri09] A. Krizhevsky, Object classification experiments. Technical report, Canadian Institute for Advanced Research (2009)
- [KRPM+18] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J.R. Ledsam, K. Maier-Hein, S.M.A. Eslami, D. Jimenez Rezende, O. Ronneberger, A probabilistic U-Net for segmentation of ambiguous images, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2018), pp. 6965–6975
- [KS18] H. Kumar, P.S. Sastry, Robust loss functions for learning multi-class classifiers, in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan (2018), pp. 1–6
- [KSFF17] M. Kull, T. Silva Filho, P. Flach, Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers, in *Proceedings*

- of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, USA (2017), pp. 623–631
- [KSH12] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Lake Tahoe, NV, USA (2012), pp. 1106–1114
- [KT08] P. Kohli, P.H.S. Torr, Measuring uncertainty in graph cut solutions. *Comput. Vis. Image Underst.* **112**(1), 30–38 (2008)
- [KTMFs20] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, T. Fingscheidt, Self-supervised monocular depth estimation: solving the dynamic object problem by semantic guidance, in *Proceedings of the European Conference on Computer Vision (ECCV)* (2020), pp. 582–600. Virtual Conference
- [KTP+20] K. Krishna, G.S. Tomar, A.P. Parikh, N. Papernot, M. Iyyer, Thieves on sesame street! Model extraction of BERT-based APIs, in *Proceedings of the International Conference on Learning Representations (ICLR)* (2020), pp. 1–19. Virtual Conference
- [KW14] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, AB, Canada (2014), pp. 1–14
- [KGW+18] B. Kim, M. Wattberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, R. Sayres, Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV), in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden (2018), pp. 2668–2677
- [KXG17] E. Kodirov, T. Xiang, S. Gong, Semantic autoencoder for zero-shot learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA (2017), pp. 4447–4456
- [KYL+20] N. Kapoor, C. Yuan, J. Löhdefink, R. Zimmermann, S. Varghese, F. Hüger, N. Schmidt, P. Schlicht, T. Fingscheidt, A self-supervised feature map augmentation (FMA) loss and combined augmentations finetuning to efficiently improve the robustness of CNNs, in *Proceedings of the ACM Computer Science in Cars Symposium (CSCS)* (2020), pp. 1–8. Virtual Conference
- [KZG18] D. Karmon, D. Zoran, Y. Goldberg, Lavan: localized and visible adversarial noise, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden (2018), pp. 2507–2515
- [LAL+20] C. Liu, T. Arnon, C. Lazarus, C. Barrett, M.J. Kochenderfer, Algorithms for verifying deep neural networks (2020), pp. 1–126. [arXiv:1903.06758](https://arxiv.org/abs/1903.06758)
- [LBD+89] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition. *Neural Comput.* **1**(4), 541–551 (1989)
- [LBL+19] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, O. Bachem, Challenging common assumptions in the unsupervised learning of disentangled representations, in *Proceedings of the International Conference on Machine Learning (ICML)*, Long Beach, CA, USA (2019), pp. 4114–4124
- [LBS+19] J. Löhdefink, A. Bär, N.M. Schmidt, F. Hüger, P. Schlicht, T. Fingscheidt, On low-bitrate image compression for distributed automotive perception: higher peak SNR does not mean better semantic segmentation, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Paris, France (2019), pp. 352–359
- [LCH+07] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F.J. Huang, Energy-based models, in *Predicting Structured Data*, ed. by G. Baklr, T. Hofmann, B. Schölkopf, A.J. Smola, B. Taskar, S.V.N. Vishwanathan (MIT Press, 2007), pp. 191–246
- [LCM+17] L. Junhua, W. Chen, Y. Ma, J. Ke, Z. Li, F. Zhang, R. Maciejewski, Recent progress and trends in predictive visual analytics. *Front. Comput. Sci.* **11**(2), 192–207 (2017)
- [LCPB18] S. Li, Y. Chen, Y. Peng, L. Bai, Learning more robust features with adversarial training (2018), pp. 1–7. [arXiv:1804.07757](https://arxiv.org/abs/1804.07757)

- [LCWJ18] M. Long, Z. Cao, J. Wang, M.I. Jordan, Conditional adversarial domain adaptation, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2018), pp. 1640–1650
- [LCY14] M. Lin, Q. Chen, S. Yan, Network in network, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, AB, Canada (2014), pp. 1–10
- [LFK+20] J. Löhdorfink, J. Fehrling, M. Klingner, F. Hüger, P. Schlicht, N.M. Schmidt, T. Fingscheidt, Self-supervised domain mismatch estimation for autonomous perception, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2020), pp. 1359–1368. Virtual Conference
- [LGG+17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy (2017), pp. 2980–2988
- [LGH+17] L. Yafeng, R. Garcia, B. Hansen, M. Gleicher, R. Maciejewski, The state-of-the-art in predictive visual analytics. *Comput. Graph. Forum* **36**(3), 539–562 (2017)
- [LGH19] J. Lin, C. Gan, S. Han, Defensive quantization: when efficiency meets robustness, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA (2019), pp. 1–15
- [LJD+18] L. Li, K.G. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: a novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.* **18**(185), 1–52 (2018)
- [LJD19] S. Liu, E. Johns, A.J. Davison, End-to-end multi-task learning with attention, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 1871–1880
- [LJL+19] H. Liu, R. Ji, J. Li, B. Zhang, Y. Gao, Y. Wu, F. Huang, Universal adversarial perturbation via prior driven uncertainty approximation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea (2019), pp. 2941–2949
- [LK19] M. Lee, Z. Kolter, On physical adversarial patches for object detection (2019), pp. 1–5. [arXiv:1906.11897](https://arxiv.org/abs/1906.11897)
- [LKD+17] H. Li, A. Kadav, I. Durdanovic, H. Samet, H.P. Graf, Pruning filters for efficient ConvNets, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France (2017), pp. 1–13
- [LLL+17] Z. Liu, X. Li, P. Luo, C.C. Loy, X. Tang, Deep learning markov random field for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **40**(8), 1814–1828 (2017)
- [LLL+19] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, W. Wen, Feature distillation: DNN-oriented JPEG compression against adversarial examples, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 860–868
- [LLLS17] K. Lee, H. Lee, K. Lee, J. Shin, Training confidence-calibrated classifiers for detecting out-of-distribution samples (2017), pp. 1–16. [arXiv:1711.09325](https://arxiv.org/abs/1711.09325)
- [LLLS18] K. Lee, K. Lee, H. Lee, J. Shin, A simple unified framework for detecting out-of-distribution samples and adversarial attacks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2018), pp. 7167–7177
- [LLS+17] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, Learning efficient convolutional networks through network slimming, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy (2017), pp. 2755–2763
- [LLS18] S. Liang, Y. Li, R. Srikant, Enhancing the reliability of out-of-distribution image detection in neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada (2018), pp. 1–15
- [LM19] Z.G. Liu, M. Mattina, Learning low-precision neural networks without straight-through estimator (STE), in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA (2019), pp. 3066–3072

- [LMW+17] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, V. Pascucci, Visualizing high-dimensional data: advances in the past decade. *IEEE Trans. Vis. Comput. Graph.* **23**(3), 1249–1268 (2017)
- [LNH14] C.H. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **36**(3), 453–465 (2014)
- [LO17] D. Lopez-Paz, M. Oquab, Revisiting classifier two-sample tests, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France (2017), pp. 1–15
- [LPB17] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Long Beach, CA, USA (2017), pp. 6402–6413
- [LPWK18] H. Li, S.J. Pan, S. Wang, A.C. Kot, Domain generalization with adversarial feature learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 5400–5409
- [LSD15] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA (2015), pp. 3431–3440
- [LSFs19] T. Lohrenz, M. Strake, T. Fingscheidt, On temporal context information for hybrid BLSTM-based phoneme recognition, in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Singapore (2019), pp. 516–523
- [LSY19] H. Liu, K. Simonyan, Y. Yang, DARTS: differentiable architecture search, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA (2019), pp. 1–13
- [LTG+18] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, D. Tao, Deep domain generalization via conditional invariant adversarial networks, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany (2018), pp. 647–663
- [LUAD16] M. Lê, J. Unkelbach, N. Ayache, H. Delingette, Sampling image segmentations for uncertainty quantification. *Med. Image Anal.* **34**, 42–51 (2016)
- [LWB+19] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking clever hans predictors and assessing what machines really learn. *Nat. Commun.* **10**(1096), 1–8 (2019)
- [LWL17] J.-H. Luo, J. Wu, W. Lin, ThiNet: a filter level pruning method for deep neural network compression, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy (2017), pp. 5068–5076
- [LWLZ17] S. Liu, X. Wang, M. Liu, J. Zhu, Towards better analysis of machine learning models: a visual analytics perspective. *Vis. Inf.* **1**(1), 48–56 (2017)
- [LWZ09] M. Li, W. Yan, Q. Zhang, SAR image segmentation based on mixture context and wavelet hidden-class-label Markov random field. *Comput. & Math. Appl.* **57**(6), 961–969 (2009)
- [LYL+19] X. Liu, H. Yang, Z. Liu, L. Song, H. Li, Y. Chen, DPATCH: an adversarial patch attack on object detectors, in *Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI)*, Honolulu, HI, USA (2019), pp. 1–8
- [LYP+19] R.G. Lopes, D. Yin, B. Poole, J. Gilmer, E.D. Cubuk, Improving robustness without sacrificing accuracy with patch Gaussian augmentation (2019), pp. 1–18. [arXiv:1906.02611](https://arxiv.org/abs/1906.02611)
- [LYSH18] D. Li, Y. Yang, Y.-Z. Song, T.M. Hospedales, Learning to generalize: meta-learning for domain generalization, in *Proceedings of the AAAI Conference on Artificial Intelligence*, New Orleans, LA, USA (2018), pp. 3490–3497
- [LYW+19] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, A. Yuille, Every pixel counts ++: joint learning of geometry and motion with 3D holistic understanding (2019), pp. 1–17. [arXiv:1810.06125](https://arxiv.org/abs/1810.06125)

- [LYZZ19] P. Li, J. Yi, B. Zhou, L. Zhang, Improving the robustness of deep neural networks via adversarial training with triplet loss, in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, Macau, China (2019), pp. 2909–2915
- [LZG+19] Y. Luo, L. Zheng, T. Guan, J. Yu, Y. Yang, Taking a closer look at domain shift: category-level adversaries for semantics consistent domain adaptation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 2507–2516
- [LZN+18] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, K. Murphy, Progressive neural architecture search, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany (2018), pp. 19–35
- [LZWJ17] M. Long, H. Zhu, J. Wang, M.I. Jordan, Deep transfer learning with joint adaptation networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia (2017), pp. 2208–2217
- [LZY+19] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, T.M. Hospedales, Episodic training for domain generalization, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea (2019), pp. 1446–1455
- [Mac98] D.J.C. MacKay, Introduction to Gaussian processes, in *Neural Networks and Machine Learning*, ed. by C.M. Bishop (Springer, Berlin, 1998), pp. 133–166
- [Mac03] D.J.C. MacKay, *Information Theory, Inference, and Learning Algorithms* (Cambridge University Press, Cambridge, 2003)
- [MARC20] M. Mancini, Z. Akata, E. Ricci, B. Caputo, Towards recognizing unseen categories in unseen domains, in *Proceedings of the European Conference on Computer Vision (ECCV)* (2020), pp. 466–483. Virtual Conference
- [MBRB18] A.S. Morcos, D.G.T. Barrett, N.C. Rabinowitz, M. Botvinick, On the importance of single directions for generalization (2018), pp. 1–15. [arXiv:1803.06959](https://arxiv.org/abs/1803.06959)
- [MBS13] K. Muandet, D. Balduzzi, B. Schölkopf, Domain generalization via invariant feature representation, in *Proceedings of the International Conference on Machine Learning (ICML)*, Atlanta, GA, USA (2013), pp. 10–18
- [MCKBF17] J.H. Metzen, M. Chaithanya Kumar, T. Brox, V. Fischer, Universal adversarial perturbations against semantic image segmentation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy (2017), pp. 2774–2783
- [MD18] W. Mei, W. Deng, D.V.D. Adaptation, A Survey. *Neurocomputing* **312**, 135–153 (2018)
- [MDFF16] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, DeepFool: a simple and accurate method to fool deep neural networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA (2016), pp. 2574–2582
- [MDFFF17] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA (2017), pp. 1765–1773
- [MDFUF19] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, P. Frossard, Robustness via curvature regularization, and vice versa, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 9078–9086
- [MDS+18] M. Mathew, K. Desappan, P.K. Swami, S. Nagori, B.M. Gopinath, Embedded low-power deep learning with TIDL. Technical Report, Texas Instruments (2018)
- [MDSN17] M. Mathew, K. Desappan, P.K. Swami, S. Nagori, Sparse, quantized, full frame CNN for low power embedded devices, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Honolulu, HI, USA (2017), pp. 328–336

- [MG18] A. Malinin, M. Gales, Predictive uncertainty estimation via prior networks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2018), pp. 7047–7058
- [MGB17] K.R. Mopuri, U. Garg, R.V. Babu, Fast feature fool: a data independent approach to universal adversarial perturbations, in *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK (2017), pp. 1–12
- [MGR+18] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, L. van der Maaten, Exploring the limits of weakly supervised pretraining, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany (2018), pp. 185–201
- [MH20a] T. Matsuurra, T. Harada, Domain generalization using a mixture of multiple latent domains, in *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, USA (2020), pp. 11749–11756
- [MH20b] A. Meinke, M. Hein, Towards neural networks that provably know when they don’t know, in *Proceedings of the International Conference on Learning Representations (ICLR)* (2020), pp. 1–18. Virtual Conference
- [Mig17] S. Migacz, 8-Bit Inference With TensorRT (2017)
- [MKH+19] A. Mustafa, S. Khan, M. Hayat, R. Goecke, J. Shen, L. Shao, Adversarial defense by restricting the hidden space of deep neural networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea (2019), pp. 3385–3394
- [ML11] R. Miotto, G. Lanckriet, A generative context model for semantic music annotation and retrieval. *IEEE/ACM Trans. Audio Speech Lang. Proc.* **20**(4), 1096–1108 (2011)
- [MLB+17] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.-R. Müller, Explaining non-linear classification decisions with deep Taylor decomposition. *Pattern Recogn.* **65**(5), 211–222 (2017)
- [MLG+18] R. Mackowiak, P. Lenz, O. Ghori, F. Diego, O. Lange, C. Rother, CEREALS – cost-effective region-based active learning for semantic segmentation, in *Proceedings of the British Machine Vision Conference (BMVC)*, Newcastle, UK (2018), pp. 1–21
- [MMR+17] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, Communication-efficient learning of deep networks from decentralized data (2017), pp. 1–11. [arXiv:1602.05629](https://arxiv.org/abs/1602.05629)
- [MMS+18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada (2018), pp. 1–10
- [MRG20] K. Maag, M. Rottmann, H. Gottschalk, Time-dynamic estimates of the reliability of deep semantic segmentation networks, in *Proceedings of the IEEE International Conference on Tools With Artificial Intelligence (ICTAI)* (2020), pp. 502–509. Virtual Conference
- [MSGH16] I. Misra, A. Shrivastava, A. Gupta, M. Hebert, Cross-stitch networks for multi-task learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA (2016), pp. 3994–4003
- [MSJ+16] A. Makhzani, J. Shlens, N. Jaitly, I.J. Goodfellow, B. Frey, Adversarial autoencoders, in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, San Juan, Puerto Rico (2016), pp. 1–16
- [MTK+17] P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz, Pruning convolutional neural networks for resource efficient inference, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France (2017), pp. 1–17
- [MTRA+12] J. Moreno-Torres, T. Raeder, R. Alaiz, N. Chawla, F. Herrera, A unifying view on dataset shift in classification. *Pattern Recogn.* **45**(1), 521–530 (2012)
- [NC16] M. Naeini, G. Cooper, Binary classifier calibration using an ensemble of near isotonic regression models, in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, Barcelona, Spain (2016), pp. 360–369

- [NCH15] M.P. Naeini, G. Cooper, M. Hauskrecht, Obtaining well calibrated probabilities using bayesian binning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Austin, TX, USA (2015), pp. 2901–2907
- [NK17] N. Narodytska, S. Kasiviswanathan, Simple black-box adversarial attacks on deep neural networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Honolulu, HI, USA (2017), pp. 1310–1318
- [NS18] D. Nilsson, C. Sminchisescu, Semantic video segmentation by gated recurrent flow propagation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 6819–6828
- [NSO20] V. Nguyen, S. Schulze, M.A. Osborne, Bayesian optimization for iterative learning, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (2020), pp. 9361–9371. Virtual Conference
- [NWC+11] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS) Workshops*, Granada, Spain (2011), pp. 1–9
- [NYC15] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: high confidence predictions for unrecognizable images, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA (2015), pp. 427–436
- [OAFS18] S.J. Oh, M. Augustin, M. Fritz, B. Schiele, Towards reverse-engineering black-box neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada (2018), pp. 1–20
- [OE18] A. Oussidi, A. Elhassouny, Deep generative models: survey, in *Proceedings of the IEEE International Conference on Intelligent Systems and Computer Vision (ISCV)*, Fez, Morocco (2018), pp. 1–8
- [OOAG19] A. Odena, C. Olsson, D. Andersen, I. Goodfellow, TensorFuzz: debugging neural networks with coverage-guided fuzzing, in *Proceedings of the International Conference on Machine Learning (ICML)*, Long Beach, CA, USA (2019), pp. 4901–4911
- [ORF20] P. Oberdiek, M. Rottmann, G.A. Fink, Detection and retrieval of out-of-distribution objects in semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 1331–1340. Virtual Conference
- [ORG18] P. Oberdiek, M. Rottmann, H. Gottschalk, Classification uncertainty of deep neural networks based on gradient information, in *Proceedings of the IAPR TC3 Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR)*, Siena, Italy (2018), pp. 113–125
- [Osb16] I. Osband, Risk versus uncertainty in deep learning: bayes, bootstrap and the dangers of dropout, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS) Workshops*, Barcelona, Spain (2016), pp. 1–5
- [OSM19] H. Okamoto, M. Suzuki, Y. Matsuo, Out-of-distribution detection using layerwise uncertainty in deep neural networks (2019). Accessed 18 Nov 2021
- [PCYJ19] K. Pei, Y. Cao, J. Yang, S. Jana, DeepXplore: automated whitebox testing of deep learning systems. *Commun. ACM* **62**(11), 137–145 (2019)
- [PDZ18] T. Pang, C. Du, J. Zhu, Max-Mahalanobis linear discriminant analysis networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden (2018), pp. 4016–4025
- [PFC+19] J. Postels, F. Ferroni, H. Coskun, N. Navab, F. Tombari, Sampling-free epistemic uncertainty estimation using approximated variance propagation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea (2019), pp. 2931–2940
- [PGZ+18] H. Pham, M.Y. Guan, B. Zoph, Q.V. Le, J. Dean, Efficient neural architecture search via parameter sharing, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden (2018), pp. 4092–4101

- [PHW22] S. Pavlitskaya, C. Hubschneider, M. Weber, Evaluating mixture-of-expert architectures for network aggregation, in *Deep Neural Networks and Data for Automated Driving - Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 345–364
- [PKGB18] O. Poursaeed, I. Katsman, B. Gao, S. Belongie, Generative adversarial perturbations, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 4422–4431
- [PL20] A. Pati, A. Lerch, Attribute-based regularization of VAE latent spaces (2020), pp. 1–15. [arXiv:2004.05485](https://arxiv.org/abs/2004.05485)
- [Pla99] John Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in *Advances in Large Margin Classifiers*. ed. by A.J. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (MIT Press, Cambridge, 1999), pp. 61–74
- [PMG+17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z.B. Celik, A. Swami, Practical black-box attacks against machine learning, in *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACSS)*, Abu Dhabi, United Arab Emirates (2017), pp. 506–519
- [PTC+17] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, G. Hinton, Regularizing neural networks by penalizing confident output distributions (2017), pp. 1–12. [arXiv:1701.06548](https://arxiv.org/abs/1701.06548)
- [PXD+20] T. Pang, K. Xu, Y. Dong, C. Du, N. Chen, J. Zhu, Rethinking softmax cross-entropy loss for adversarial robustness, in *Proceedings of the International Conference on Learning Representations (ICLR)* (2020), pp. 1–19. Virtual Conference
- [PY11] G. Papandreou, A.L. Yuille, Perturb-and-map random fields: using discrete optimization to learn and sample from energy models, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain (2011), pp. 193–200
- [RA17] M. Rahmani, G. Atia, C. Pursuit, Fast, simple, and robust principal component analysis. *IEEE Trans. Signal Proc.* **65**(23), 6260–6275 (2017)
- [RAHL19] E. Real, A. Aggarwal, Y. Huang, Q.V. Le, Regularized evolution for image classifier architecture search, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA (2019), pp. 4780–4789
- [Ras03] C.E. Rasmussen, Gaussian processes in machine learning, in *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*, ed. by O. Bousquet, U. von Luxburg, G. Rätsch (Springer, Berlin, 2003), pp. 63–71
- [RBAS18] S. Ruder, J. Bingel, I. Augenstein, A. Søgaard, Latent multi-task architecture learning (2018), pp. 1–8. [arXiv:1705.08142](https://arxiv.org/abs/1705.08142)
- [RBB18] H. Ritter, A. Botev, D. Barber, A scalable laplace approximation for neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada (2018), pp. 1–15
- [RBV18] S.-A. Rebuffi, H. Bilen, A. Vedaldi, Efficient parametrization of multi-domain deep neural networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 8119–8127
- [RCH+20] M. Rottmann, P. Colling, T.-P. Hack, R. Chan, F. Hüger, P. Schlicht, H. Gottschalk, Prediction error meta classification in semantic segmentation: detection via aggregated dispersion measures of softmax probabilities, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* (2020), pp. 1–9. Virtual Conference
- [RDG18] S. Roychowdhury, M. Diligenti, M. Gori, Image classification using deep learning and prior knowledge, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, London, UK (2018), pp. 336–343

- [RDGF15] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA (2015), pp. 779–788
- [REBO22] R. Rombach, P. Esser, A. Blattmann, B. Ommer, Invertible neural networks for understanding semantics of invariances of CNN representations, in *Deep Neural Networks and Data for Automated Driving – Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 223–253
- [Ree93] R. Reed, Pruning algorithms - a survey. *IEEE Trans. Neural Netw. (TNN)* **4**(5), 740–747 (1993)
- [RKG18] M. Rottmann, K. Kahl, H. Gottschalk, Deep Bayesian active semi-supervised learning, in *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, FL, USA (2018), pp. 158–164
- [RL19] M. Rahmani, P. Li, Outlier detection and data clustering via innovation search (2019), pp. 1–18. [arXiv:1912.12988](https://arxiv.org/abs/1912.12988)
- [RMC15] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks (2015), pp. 1–16. [arXiv:1511.06434](https://arxiv.org/abs/1511.06434)
- [RMDC19] S.K. Roy, S. Manna, S.R. Dubey, B.B. Chaudhuri, LiSHT: non-parametric linearly scaled hyperbolic tangent activation function for neural networks (2019), pp. 1–11. [arXiv:1901.05894](https://arxiv.org/abs/1901.05894)
- [RRMH17] R. Rasti, H. Rabbani, A. Mehridehnavi, F. Hajizadeh, Macular OCT classification using a multi-scale convolutional neural network ensemble. *IEEE Trans. Med. Imag.* **37**(4), 1024–1034 (2017)
- [RS19] M. Rottmann, M. Schubert, Uncertainty measures and prediction quality rating for the semantic segmentation of nested multi resolution street scene images, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA (2019), pp. 1361–1369
- [RSFD16] E. Rodner, M. Simon, R.B. Fisher, J. Denzler, Fine-grained recognition in the noisy wild: sensitivity analysis of convolutional neural networks approaches, in *Proceedings of the British Machine Vision Conference (BMVC)*, York, UK (2016), pp. 1–13
- [RSFM19] E. Raff, J. Sylvester, S. Forsyth, M. McLean, Barrage of random transforms for adversarially robust defense, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 6528–6537
- [RSG16] M.T. Ribeiro, S. Singh, C. Guestrin, “Why should I trust you?”: explaining the predictions of any classifier, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, New York, NY, USA (2016), pp. 1135–1144
- [RSKR22] T. Riedlinger, M. Schubert, K. Kahl, M. Rottmann, Uncertainty quantification for object detection: output- and gradient-based approaches, in *Deep Neural Networks and Data for Automated Driving – Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 283–306
- [RSS18] J. Rabold, M. Siebers, U. Schmid, Explaining black-box classifiers with ILP – empowering LIME with aleph to approximate non-linear decisions with relational rules, in *Proceedings of the International Conference on Inductive Logic Programming (ILP)*, Ferrara, Italy (2018), pp. 105–117
- [RTG+19] H. Rezatofighi, N. Tsoi, J.Y. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: a metric and a loss for bounding box regression, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 658–666
- [RUN18] S. Roy, A. Unmesh, V.P. Namboodiri, Deep active learning for object detection, in *Proceedings of the British Machine Vision Conference (BMVC)*, Newcastle, UK (2018), pp. 1–12

- [RZL18] P. Ramachandran, B. Zoph, Q.V. Le, Searching for activation functions, in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, Vancouver, BC, Canada (2018), pp. 1–13
- [SAMG19] T. Sämann, K. Amende, S. Milz, H.-M. Groß, Robust semantic video segmentation through confidence-based feature map warping, in *Proceedings of the ACM Computer Science in Cars Symposium (CSCS)*, Kaiserslautern, Germany (2019), pp. 1–9
- [SBMC17] S. Sarkar, A. Bansal, U. Mahbub, R. Chellappa, UPSET and ANGRI: breaking high performance image classifiers (2017), pp. 1–9. [arXiv:1707.01159](https://arxiv.org/abs/1707.01159)
- [SBS12] P. Sprechmann, A.M. Bronstein, G. Sapiro, Learning robust low-rank representations (2012), pp. 1–15. [arXiv:1209.6393](https://arxiv.org/abs/1209.6393)
- [SCD+20] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis. (IJCV)* **128**, 336–359 (2020)
- [Sco15] D.W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization* (Wiley, New York, 2015)
- [SCW+15] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.K. Wong, W.C. Woo, Convolutional LSTM network: a machine learning approach for precipitation nowcasting, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2015), pp. 802–810
- [SDBR14] J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: the all convolutional net (2014), pp. 1–14. [arXiv:1412.6806](https://arxiv.org/abs/1412.6806)
- [SDF+20] M. Strake, B. Defraene, K. Fluyt, W. Tirry, T. Fingscheidt, Fully convolutional recurrent networks for speech enhancement, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), pp. 6674–6678. Virtual Conference
- [SDKF19] H. Song, T. Diethe, M. Kull, P. Flach, Distribution calibration for regression, in *Proceedings of the International Conference on Machine Learning (ICML)*, Long Beach, CA, USA (2019), pp. 5897–5906
- [SDW+19] D. Stamoulis, R. Ding, D. Wang, D. Lymberopoulos, B. Priyantha, J. Liu, D. Marculescu, Single-path NAS: designing hardware-efficient ConvNets in less than 4 hours, in *Proceedings of the Joint European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, Würzburg, Germany (2019), pp. 481–497
- [Set10] B. Settles, Active learning literature survey. Technical Report, University of Wisconsin-Madison (2010)
- [SFH17] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Long Beach, CA, USA (2017), pp. 3856–3866
- [SGK19] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences (2019), pp. 1–9. [arXiv:1704.02685](https://arxiv.org/abs/1704.02685)
- [SGS15] R.K. Srivastava, K. Greff, J. Schmidhuber, Training very deep networks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2015), pp. 2377–2385
- [SGSS07] A.J. Smola, A. Gretton, L. Song, B. Schölkopf, A Hilbert space embedding for distributions, in *Proceedings of the International Conference Algorithmic Learning Theory (ALT)*, Sendai, Japan (2007), pp. 13–31
- [SH09] R. Salakhutdinov, G. Hinton, Deep Boltzmann machines, in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Clearwater Beach, FL, USA (2009), pp. 448–455
- [She04] S.J. Sheather, Density estimation. *Stat. Sci.* **19**(4), 588–597 (2004)
- [SHK+14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(56), 1929–1958 (2014)

- [SHK+19] Y. Sun, X. Huang, D. Kroening, J. Sharp, M. Hill, R. Ashmore, Structural test coverage criteria for deep neural networks. *ACM Trans. Embed. Comput. Syst. (TECS)* **18**(5s), 1–23 (2019)
- [Shn96] B. Shneiderman, The eyes have it: a task by data type taxonomy for information visualizations, in *Proceedings of the IEEE Symposium on Visual Languages*, Boulder, CO, USA (1996), pp. 336–343
- [Sil86] B.W. Silverman, *Density Estimation for Statistics and Data Analysis* (Chapman and Hall/CRC Press, London, 1986)
- [SIVA17] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, Inception-ResNet and the impact of residual connections on learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA (2017), pp. 4278–4284
- [SJG+20] P. Stock, A. Joulin, R. Gribonval, B. Graham, H. Jégou, And the bit goes down: revisiting the quantization of neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)* (2020), pp. 1–11. Virtual Conference
- [SK19] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning. *J. Big Data* **60**(6), 1–48 (2019). (July)
- [SKF18] H. Song, M. Kull, P. Flach, Non-parametric calibration of probabilistic regression (2018), pp. 1–17. [arXiv:1806.07690](https://arxiv.org/abs/1806.07690)
- [SLC19] I. Seck, G. Loosli, S. Canu, L1-norm double backpropagation adversarial defense, in *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Bruges, Belgium (2019), pp. 1–6
- [SLJ+15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA (2015), pp. 1–9
- [SLY15] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2015), pp. 3483–3491
- [SMK20] N. Somavarapu, C.-Y. Ma, Z. Kira, Frustratingly simple domain generalization via image stylization (2020), pp. 1–15. [arXiv:2006.11207](https://arxiv.org/abs/2006.11207)
- [SMM+17] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, J. Dean, Outrageously large neural networks: the sparsely-gated mixture-of-experts layer (2017), pp. 1–19. [arXiv:1701.06538](https://arxiv.org/abs/1701.06538)
- [SOF+19] J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. Dillon, J. Ren, Z. Nado, Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2019), pp. 13969–13980
- [SR20] S. Saito, S. Roy, Effects of loss functions and target representations on adversarial robustness, in *Proceedings of the Conference on Machine Learning and Systems (MLSys) Workshops*, Austin, TX, USA (2020), pp. 1–10
- [SRHD16] E. Shelhamer, K. Rakelly, J. Hoffman, T. Darrell, Clockwork convnets for video segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, Amsterdam, The Netherlands (2016), pp. 852–868
- [SRK20] P. Sorrenson, C. Rother, U. Köthe, Disentanglement by nonlinear ICA with general incompressible-flow networks (GIN) (2020), pp. 1–23. [arXiv:2001.04872](https://arxiv.org/abs/2001.04872)
- [SRL+22] H. Stage, L. Ries, J. Langner, S. Otten, E. Sax, Analysis and comparison of datasets by leveraging data distributions in latent spaces, in *Deep Neural Networks and Data for Automated Driving – Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 129–148

- [SS20a] G. Schwalbe, M. Schels, A survey on methods for the safety assurance of machine learning based systems, in *Proceedings of the European Congress on Embedded Real Time Software and Systems (ERTS)*, Toulouse, France (2020), pp. 1–10
- [SS20b] G. Schwalbe, M. Schels, Concept enforcement and modularization as methods for the ISO 26262 safety argumentation of neural networks, in *Proceedings of the European Congress on Embedded Real Time Software and Systems (ERTS)*, Toulouse, France (2020), pp. 1–10
- [SSH19] S. Seo, P.H. Seo, B. Han, Learning for single-shot confidence calibration in deep neural networks through stochastic inferences, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 9030–9038
- [SSH20] T. Sämann, P. Schlicht, F. Hüger, Strategy to increase the safety of a DNN-based perception for HAD systems (2020), pp. 1–14. [arXiv:2002.08935](https://arxiv.org/abs/2002.08935)
- [SSSS17] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA (2017), pp. 3–18
- [SSW+17] T. Schlegl, P. Seeböck, S.M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in *Proceedings of the International Conference on Information Processing in Medical Imaging (IPMI)*, Boone, NC, USA (2017), pp. 146–157
- [SSZ+17] D. Sacha, M. Sedlmair, L. Zhang, J.A. Lee, J. Peltonen, D. Weiskopf, S.C. North, D.A. Keim, W.Y.S.I.W.Y.C. Change, Human-centered machine learning by interactive visualization. *Neurocomputing* **268**, 164–175 (2017)
- [SSZ19] M. Svatos, G. Sourek, F. Zelezny, Revisiting neural-symbolic learning cycle, in *Proceedings of the International Workshop on Neural-Symbolic Learning and Reasoning (NeSy)*, Macao, China (2019), pp. 1–6
- [STK+17] D. Smilkov, N. Thorat, B. Kim, F.B. Viégas, M. Wattenberg, SmoothGrad: removing noise by adding noise (2007), pp. 1–10. [arXiv:1706.03825](https://arxiv.org/abs/1706.03825)
- [Sto08] A.J. Storkey, When training and test sets are different: characterising learning transfer, in *Dataset Shift in Machine Learning*, ed. by J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence (MIT Press, 2008), pp. 3–28
- [STP17] B. Summa, J. Tierny, V. Pascucci, Visualizing the uncertainty of graph-based 2D segmentation with min-path stability. *Comput. Graph. Forum* **36**(3), 133–143 (2017)
- [STY17] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia (2017), pp. 3319–3328
- [SU15] A.G. Schwing, R. Urtasun, Fully connected deep structured networks (2015), pp. 1–10. [arXiv:1503.02351](https://arxiv.org/abs/1503.02351)
- [SVS19] J. Su, D.V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput. (TEVC)* **23**(5), 828–841 (2019)
- [SVZ14] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, Banff, AB, Canada (2014), pp. 1–8
- [SWR+18] Y. Sun, M. Wu, W. Ruan, X. Huang, M. Kwiatkowska, D. Kroening, Concolic testing for deep neural networks, in *Proceedings of the ACM/IEEE International Conference on Automated Software Engineering (ASE)*, Montpellier, France (2018), pp. 109–119
- [SYPS19] V. Sandfort, K. Yan, P. Pickhardt, R. Summers, Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Sci. Rep.* **9**(1–9), 16884 (2019)
- [SYZ+21] X. Sun, Z. Yang, C. Zhang, G. Peng, K.-V. Ling, Conditional Gaussian distribution learning for open set recognition (2021), pp. 1–10. [arXiv:2003.08823](https://arxiv.org/abs/2003.08823)

- [SZ15] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA (2015), pp. 1–14
- [Szs+14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, AB, Canada (2014), pp. 1–10
- [Szs+17] D. Sacha, L. Zhang, M. Sedlmair, J.A. Lee, J. Peltonen, D. Weiskopf, S.C. North, D.A. Keim, Visual interaction with dimensionality reduction: a structured literature analysis. *IEEE Trans. Visual. Comput. Graph.* **23**(1), 241–250 (2017)
- [SZT17] R. Shwartz-Ziv, N. Tishby, Opening the black box of deep neural networks via information (2017), pp. 1–19. [arXiv:1703.00810](https://arxiv.org/abs/1703.00810)
- [SZYP19] H. Sikka, W. Zhong, J. Yin, C. Pehlevan, A closer look at disentangling in *beta*-VAE (2019), pp. 1–8. [arXiv:1912.05127](https://arxiv.org/abs/1912.05127)
- [TA12] D. Tarlow, R.P. Adams, Revisiting uncertainty in graph cut solutions, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA (2012), pp. 2440–2447
- [TAGD98] A.B. Tickle, R. Andrews, M. Golea, J. Diederich, The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks. *IEEE Trans. Neural Netw. (TNN)* **9**(6), 1057–1068 (1998)
- [Tay06] B.J. Taylor, *Methods and Procedures for the Verification and Validation of Artificial Neural Networks* (Springer, Berlin, 2006)
- [TBF+15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3D convolutional networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile (2015), pp. 4489–4497
- [TBGS19] I. Tolstikhin, O. Bousquet, S. Gelly, B. Schoelkopf, Wasserstein auto-encoders (2019), pp. 1–20. [arXiv:1711.01558](https://arxiv.org/abs/1711.01558)
- [TC05] J.J. Thomas, K.A. Cook, *Illuminating the Path: The Research and Development Agenda for Visual Analytics* (IEEE, 2005)
- [TCBZ19] R. Theagarajan, M. Chen, B. Bhanu, J. Zhang, ShieldNets: defending against adversarial attacks using probabilistic adversarial robustness, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 6988–6986
- [Thr95] S. Thrun, Extracting rules from artificial neural networks with distributed representations, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Denver, CO, USA 1995, pp. 505–512
- [Thr02] S. Thrun, Robotic mapping: a survey. Technical report, Carnegie Mellon University (2002)
- [TKTH18] L. Theis, I. Korshunova, A. Tejani, F. Huszár, Faster gaze prediction with dense networks and fisher pruning (2018), pp. 1–18. [arXiv:1801.05787](https://arxiv.org/abs/1801.05787)
- [TPJR18] Y. Tian, K. Pei, S. Jana, B. Ray, DeepTest: automated testing of deep-neural-network-driven autonomous cars, in *Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE)* (2018), pp. 303–314
- [TVRG19] S. Thys, W. Van Ranst, T. Goedemé, Fooling automated surveillance cameras: adversarial patches to attack person detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA (2019), pp. 49–55
- [TWZ+18] M. Teichmann, M. Weber, J.M. Zollner, R. Cipolla, R. Urtasun, MultiNet: real-time joint semantic reasoning for autonomous driving, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China (2018), pp. 1013–1020
- [TZJ+16] F. Tramèr, F. Zhang, A. Juels, M.K. Reiter, T. Ristenpart, Stealing machine learning models via prediction APIs, in *Proceedings of the USENIX Security Symposium*, Austin, TX, USA (2016), pp. 601–618

- [UEKH19] H. Uzunova, J. Ehrhardt, T. Kepp, H. Handels, Interpretable explanations of black box classifiers applied on medical images by meaningful perturbations using variational autoencoders, in *Proceedings of the SPIE Medical Imaging*, San Diego, CA, USA (2019), 1094911
- [UMY+20] S. Uhlich, L. Mauch, K. Yoshiyama, F. Cardinaux, J.A. García, S. Tiedemann, T. Kemp, A. Nakamura, Differentiable quantization of deep neural networks (2020), pp. 1–21. [arXiv:1905.11452](https://arxiv.org/abs/1905.11452)
- [VBB+20] S. Varghese, Y. Bayzidi, A. Bär, N. Kapoor, S. Lahiri, J.D. Schneider, N. Schmidt, F. Hüger, P. Schlicht, T. Fingscheidt, Unsupervised temporal consistency metric for video segmentation in highly-automated driving, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2020), pp. 1369–1378. Virtual Conference
- [VHB+22] S. Varghese, C. Hümer, A. Bär, F. Hüger, T. Fingscheidt, Joint optimization for DNN model compression and corruption robustness, in *Deep Neural Networks and Data for Automated Driving – Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben (Springer, Berlin, 2022), pp. 435–458
- [VJB+19] T.-H. Vu, H. Jain, M. Bucher, M. Cord, P. Perez, ADVENT: adversarial entropy minimization for domain adaptation in semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 2517–2526
- [vLSB20] S. van Steenkiste, F. Locatello, J. Schmidhuber, O. Bachem, Are disentangled representations helpful for abstract visual reasoning? (2020), pp. 1–14. [arXiv:1905.12506](https://arxiv.org/abs/1905.12506)
- [WB98] C.K.I. Williams, D. Barber, Bayesian classification with Gaussian processes. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **20**(12), 1342–1351 (1998)
- [Wer78] W. Wertz, *Statistical Density Estimation: A Survey* (Vandenhoeck & Ruprecht, 1978)
- [Wes04] T.H.-W. Westerveld, Using generative probabilistic models for multimedia retrieval. Dissertation, University of Twente (2004)
- [WFZ19] M. Weber, M. Fürst, J.M. Zöllner, Automated focal loss for image based object detection (2019), pp. 1–9. [arXiv:1904.09048](https://arxiv.org/abs/1904.09048)
- [WGH19] M. Woehrle, C. Gladisch, C. Heinzenmann, Open questions in testing of learned computer vision functions for automated driving, in *Proceedings of the International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*, Toulouse, France (2019), pp. 333–345
- [WGSM17] S.C. Wong, A. Gatt, V. Stamatescu, M.D. McDonnell, Understanding data augmentation for classification: when to warp?, in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Gold Coast, QLD, Australia (2017), pp. 1–6
- [WHLX19] H. Wang, Z. He, Z.C. Lipton, E.P. Xing, Learning robust representations by projecting superficial statistics out, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA (2019), pp. 1–16
- [WJZ+20] H. Wu, P. Judd, X. Zhang, M. Isaev, P. Micikevicius, Integer quantization for deep learning inference: principles and empirical evaluation (2020), pp. 1–20. [arXiv:2004.09602](https://arxiv.org/abs/2004.09602)
- [WKP13] C. Wang, N. Komodakis, N. Paragios, M.R.F. Modeling, Inference & learning in computer vision & image understanding: a survey. *Comput. Vis. Image Underst.* **117**(11), 1610–1627 (2013)
- [WLDG20] Z. Wu, S.-N. Lim, L. Davis, T. Goldstein, Making an invisibility cloak: real world adversarial attacks on object detectors, in *Proceedings of the European Conference on Computer Vision (ECCV)* (2020), pp. 1–17. Virtual Conference
- [WR96] C.K.I. Williams, C.E. Rasmussen, Gaussian processes for regression, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Denver, CO, USA (1996), pp. 514–520

- [WSC+20] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, M. Yadong, M. Tan, X. Wang et al., Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **43**(10), 3349–3364 (2020)
- [WSRA20] O. Willers, S. Sudholt, S. Raafatnia, S. Abrecht, Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks, in *Proceedings of the International Conference on Computer Safety, Reliability, and Security (SAFECOMP) Workshops* (2020), pp. 336–350
- [WT11] M. Welling, Y.W. Teh, Bayesian learning via stochastic gradient Langevin dynamics, in *Proceedings of the International Conference on Machine Learning (ICML)*, Bellevue, WA, USA (2011), pp. 1–11
- [WTPFW19] J. Wu, S. Toscano-Palmerin, P.I. Frazier, A.G. Wilson, Practical multi-fidelity bayesian optimization for hyperparameter tuning, in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, Tel Aviv, Israel (2019), pp. 1–11
- [WYKN20] Y. Wang, Q. Yao, J.T. Kwok, M.N. Lionel, Generalizing from a few examples: a survey on few-shot learning. *ACM Comput. Surv. (CSUR)* **53**(3), 1–34 (2020)
- [WZM+16] X.-M. Wang, T.-Y. Zhang, Y.-X. Ma, J. Xia, W. Chen, A survey of visual analytic pipelines. *J. Comput. Sci. Technol.* **31**(4), 787–804 (2016)
- [XCKW19] Y. Xu, P. Cao, Y. Kong, Y. Wang,  $L_{DMI}$  : a novel information-theoretic loss function for training deep nets robust to label noise, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2019), pp. 6222–6233
- [XCS10] H. Xu, C. Caramanis, S. Sanghavi, Robust PCA via outlier pursuit, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada (2010), pp. 2496–2504
- [XGD+17] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA (2017), pp. 1492–1500
- [XHM+11] X. Xie, J.W.K. Ho, C. Murphy, G. Kaiser, B. Xu, T.Y. Chen, Testing and validating machine learning classifiers by metamorphic testing. *Syst. Softw.* **84**(4), 544–558 (2011)
- [XLH+20] H. Xu, D. Luo, R. Henao, S. Shah, L. Carin, Learning autoencoders with relational regularization (2020), pp. 1–18. [arXiv:2002.02913](https://arxiv.org/abs/2002.02913)
- [XLZ+18] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, J. Sun, Unified perceptual parsing for scene understanding, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany (2018), pp. 432–448
- [XW19] Y. Xiao, W.Y. Wang, Disentangled representation learning with Wasserstein total correlation (2019), pp. 1–10. [arXiv:1912.12818](https://arxiv.org/abs/1912.12818)
- [XWvdM+19] C. Xie, Y. Wu, L. van der Maaten, A.L. Yuille, K. He, Feature denoising for improving adversarial robustness, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 501–507
- [XZL+20] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, X. Lin, Adversarial T-shirt! evading person detectors in a physical world, in *Proceedings of the European Conference on Computer Vision (ECCV)* (2020), pp. 665–681. Virtual Conference
- [XZZ+19] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, A.L. Yuille, Improving transferability of adversarial examples with input diversity, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 2730–2739
- [YDL+17] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, W. Zuo, Mind the class weight bias: weighted maximum mean discrepancy for unsupervised domain adaptation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA (2017), pp. 945–954

- [YLLW18] J. Ye, X. Lu, Z. Lin, J. Wang, Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada (2018), pp. 1–11
- [YLW+20] H. Yan, Z. Li, Q. Wang, P. Li, X. Yong, W. Zuo, Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation. *IEEE Trans. Multimedia* **22**(9), 2420–2433 (2020)
- [YXL+19] S. Ye, K. Xu, S. Liu, H. Cheng, J.H. Lambrechts, H. Zhang, A. Zhou, K. Ma, Y. Wang, X. Lin, Adversarial robustness vs. model compression, or both?, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea (2019), pp. 111–120
- [YZS+18] G. Yang, H. Zhao, J. Shi, Z. Deng, J. Jia, SegStereo: exploiting semantic information for disparity estimation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany (2018), pp. 636–651
- [ZCAW17] L.M. Zintgraf, T.S. Cohen, T. Adel, M. Welling, Visualizing deep neural network decisions: prediction difference analysis (2017), pp. 1–12. [arXiv:1702.04595](https://arxiv.org/abs/1702.04595)
- [ZCG+19] B. Zoph, E.D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, Q.V. Le, Learning data augmentation strategies for object detection (2013), pp. 1–13. [arXiv:1906.11172](https://arxiv.org/abs/1906.11172)
- [ZCY+19] X. Zhang, X. Chen, L. Yao, C. Ge, M. Dong, Deep neural network hyperparameter optimization with orthogonal array tuning, in *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, Sydney, NSW, Australia (2019), pp. 287–295
- [ZE01] B. Zadrozny, C. Elkan, Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers, in *Proceedings of the International Conference on Machine Learning (ICML)*, Williamstown, MA, USA (2001), pp. 609–616
- [ZE02] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Edmonton, AB, Canada (2002), pp. 694–699
- [ZF14] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Zurich, Switzerland (2014), pp. 818–833
- [ZGY+19] F. Zhu, R. Gong, F. Yu, X. Liu, Y. Wang, Z. Li, X. Yang, J. Yan, Towards unified INT8 training for convolutional neural network (2019), pp. 1–14. [arXiv:1912.12607](https://arxiv.org/abs/1912.12607)
- [Zha12] B. Zhang, Reliable classification of vehicle types based on cascade classifier ensembles. *IEEE Trans. Intell. Trans. Syst. (TITS)* **14**(1), 322–332 (2012)
- [ZHD+19] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, Z. Zhang, Improving neural network quantization without retraining using outlier channel splitting, in *Proceedings of the International Conference on Machine Learning (ICML)*, Long Beach, CA, USA (2019), pp. 7543–7552
- [ZHML20] J.M. Zhang, M. Harman, L. Ma, Y. Liu, Machine learning testing: survey, landscapes and horizons. *IEEE Trans. Softw. Eng.* 1–37 (2020). Early access
- [ZJRP+15] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P.H.S. Torr, Conditional random fields as recurrent neural networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile (2015), pp. 1529–1537
- [ZLMJ16] J.R. Zilke, E. Loza Mencía, F. Janssen, DeepRED – rule extraction from deep neural networks, in *Proceedings of the International Conference on Discovery Science (DS)*, Bari, Italy (2016), pp. 457–473
- [ZLZ+20] H. Zhou, W. Li, Y. Zhu, Y. Zhang, B. Yu, L. Zhang, C. Liu, Deepbillboard: systematic physical-world testing of autonomous driving systems, in *Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE)* (2020), pp. 347–358. Virtual Conference

- [ZP17] C. Zhou, R.C. Paffenroth, Anomaly detection with robust deep autoencoders, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Halifax, NS, Canada (2017), pp. 665–674
- [ZPIE17] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy (2017), pp. 2242–2251
- [ZQF+18] H. Zhuo, X. Qian, Y. Fu, H. Yang, X. Xue, SCSP: spectral clustering filter pruning with soft self-adaption manners (2018), pp. 1–14. [arXiv:1806.05320](https://arxiv.org/abs/1806.05320)
- [ZSLG16] S. Zheng, Y. Song, T. Leung, I. Goodfellow, Improving the robustness of deep neural networks via stability training, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA (2016), pp. 4480–4488
- [ZSR+19] Y. Zhu, K. Sapra, F.A. Reda, K.J. Shih, S. Newsam, A. Tao, B. Catanzaro, Improving semantic segmentation via video propagation and label relaxation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA (2019), pp. 8856–8865
- [ZVSL18] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA (2018), pp. 8697–8710
- [ZWN+18] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, Y. Zou, DoReFa-Net: training low bitwidth convolutional neural networks with low bitwidth gradients (2018), pp. 1–13. [arXiv:1606.06160](https://arxiv.org/abs/1606.06160)
- [ZZK+20] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, USA (2020), pp. 13001–13008
- [ZZW+18] H. Zhao, S. Zhang, G. Wu, J.M.F. Moura, J.P. Costeira, G.J. Gordon, Adversarial multiple source domain adaptation, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada (2018), pp. 8568–8579

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



# **Recent Advances in Safe AI for Automated Driving**

# Does Redundancy in AI Perception Systems Help to Test for Super-Human Automated Driving Performance?



Hanno Gottschalk, Matthias Rottmann, and Maida Saltagic

**Abstract** While automated driving is often advertised with better-than-human driving performance, this chapter reviews that it is nearly impossible to provide direct statistical evidence on the system level that this is actually the case. The amount of labeled data needed would exceed dimensions of present-day technical and economical capabilities. A commonly used strategy therefore is the use of redundancy along with the proof of sufficient subsystems' performances. As it is known, this strategy is efficient especially for the case of subsystems operating independently, i.e., the occurrence of errors is independent in a statistical sense. Here, we give some first considerations and experimental evidence that this strategy is not a free ride as the errors of neural networks fulfilling the same computer vision task, at least in some cases, show correlated occurrences of errors. This remains true, if training data, architecture, and training are kept separate or independence is trained using special loss functions. Using data from different sensors (realized by up to five 2D projections of the 3D MNIST dataset) in our experiments is more efficiently reducing correlations, however not to an extent that is realizing the potential of reduction of testing data that can be obtained for redundant and statistically independent subsystems.

---

H. Gottschalk and M. Rottmann have equal contribution.

H. Gottschalk (✉) · M. Rottmann · M. Saltagic  
School of Mathematics and Science and IZMD, University of Wuppertal, Gaußstr. 20,  
Wuppertal, Germany  
e-mail: [hanno.gottschalk@uni-wuppertal.de](mailto:hanno.gottschalk@uni-wuppertal.de)

M. Rottmann  
e-mail: [rottmann@uni-wuppertal.de](mailto:rottmann@uni-wuppertal.de)

M. Saltagic  
e-mail: [maida.saltagic@gmx.de](mailto:maida.saltagic@gmx.de)

## 1 Introduction

The final report of the ethics committee on automated and connected driving [FBB+17] at the German Federal Ministry of Transportation and Digital Infrastructure starts with the sentences<sup>1</sup> “Partially and fully automated traffic systems serve first and foremost to improve the safety of all road users. [...] Protecting people takes precedence over all other utilitarian considerations. The goal is to reduce harm up to complete prevention. The approval of automated systems is only justifiable if, in comparison with human driving performance, they promise at least a reduction of damage in the sense of a positive risk balance”. This pronounced statement sets the highest safety goals. In this chapter, we contemplate the feasibility of a justification based on direct empirical evidence.

If it comes to automated driving, the elephant in the room is the outrageous amount of data that is needed to empirically support the safety requirement set up by the ethics committee with direct measurement. This chapter, however, is not the elephant’s first sighting; see, e.g., [KP16], where it is shown that hundreds of millions to billions of test kilometers are required for statistically valid evidence of better-than-human driving performance by automated vehicles. While in this chapter the basic statistical facts on the measurement of the probability of rare events are revisited and adapted to a German context, we slightly extend the findings by estimating the data required for testing AI-based perception functionality using optical sensors along with an estimate of the labeling cost for a sufficient test database.

What is new in this chapter is a statistical discussion and preliminary experimental evidence on redundancy as a potential solution to the aforementioned problem. The decomposition of the system into redundant subsystems, each one capable to trigger the detection of other road users without fusion or filtering, largely reduces the amount of data needed to test each subsystem. However, this is only true if the failure of the subsystems is statistically independent of the other subsystems. This leads to the question of (a) how to measure independence and (b) whether the actual behavior of neural networks supports the independence assumption.

A study on the role of independence in ensembles of deep neural networks was presented in [LWC+19], where the goal was rather (1) to improve performance by selecting ensemble members according to different diversity scores and (2) to obtain robustness against adversarial attacks. In [WLX+20], a number of different consensus algorithms, i.e., ensemble voting algorithms, are compared according to different evaluation metrics. Also in that work, networks are trained independently and selected afterwards.

In our own studies of independence of the occurrence of error events in the prediction of neural networks, we provide experiments for classification with deep neural networks on the academic datasets EMNIST [CATvS17], CIFAR10 [Kri09], and 3D-MNIST.<sup>2</sup> We consider networks with an increasing degree of diversity with respect to training data, architecture, and weight initialization. Even the most diverse networks

---

<sup>1</sup> Translated from German.

<sup>2</sup> <https://www.kaggle.com/daavoo/3d-mnist>.

exhibit a Pearson correlation close to 0.60, which clearly contradicts the hypothesis of independence. Also, re-training committees of up to 5 networks with special loss functions to increase independence between committee members by far does not achieve the  $k$ -out-of- $n$  performance predicted for independent subsystems [Zac92, ME14]. While it is possible to bring the mean correlation down to zero by special loss functions in the training, this, at least in our preliminary experiments, at the same time deteriorates the performance of the committee members. As the main take-away, redundancy does not necessarily provide a solution to the testing problem.

In this chapter, we do not aim at presenting final results, but only to provide a contribution to an inevitable debate.

Our chapter is organized as follows: In the next section, we evaluate some numbers from the traffic by motor vehicles in the last pre-pandemic year in Germany, 2019. In Sect. 3, we recall some basic facts on the statistics of rare events of an entire system or a system of redundant subsystems. While independent redundant subsystems largely reduce the amount of data required for testing, we also consider the case of correlated subsystems for which the data requirements scale down less neatly. Also, we discuss the amount of data required to actually prove sufficiently low correlation. In Sect. 4, we test neural networks for independence or correlation for simple classification problems. Not surprisingly, we find that such neural networks actually provide correlated error schemes, and the system performance falls far behind the theoretically predicted performance for the error of statistically independent classifiers. This holds true even if we train networks to behave independently or feed the networks with different (toy) sensors. This demonstrates that independence cannot be taken for granted and it might be even hard to achieve through training methods. We give our conclusions and provide a brief outlook on other approaches that have the potential to resolve the issue of outrageous amounts of labeled data for a direct assurance case in the final Sect. 5.

## 2 How Much Data is Needed for Direct Statistical Evidence of Better-Than-Human Driving?

We focus on the loss of human life as the most important safety requirement. Our frame of reference is set by the traffic in Germany in the year 2019. For this year, the Federal Ministry of Transport and Digital Infrastructure reports 3,046 fatalities which results in 4.0 fatalities per billion kilometers driven on German streets in total and 1.1 fatalities per billion kilometers on motorways; see [Ger20, p. 165] for these and more detailed data.

If we neglect that some accidents do not involve human drivers, that in deadly accidents oftentimes more than one person is killed, and that a large number of those killed did not cause the fatal accident, we obtain a lower bound of at least 250 million kilometers driven per fatality caused by the average human driver. Research on how much this is underestimating the actual distance is recommended but beyond the

scope of this chapter. For an upper bound, we multiply this number by an ad hoc safety factor of 10.

Assuming an average velocity in the range of 50–100 km/h, this amounts to an average time of about 2.5–50 million hours or 285–5,700 years of permanent driving until the occurrence of a single fatal accident. If a camera sensor works at a frame rate of 20–30 fps,  $(1.8\text{--}54) \times 10^{11}$  frames are processed by the AI-system in this time, corresponding to 0.18–5.4 exabyte (1 exabyte =  $1 \times 10^{18}$  bytes) of data, assuming 1 megabyte per high-resolution image.

Several factors can be identified that would leverage or discount the amount of data required for a direct test of better-than-human safety. We do not claim that our selection of factors is exhaustive and new ideas might change the figures in the future. Nevertheless, here we present some factors that certainly are of importance.

First, due to the strong correlation of consecutive frames, the frame rate of 20–30 fps presumably can be reduced for labeled data. Here, we take the approach that correlation probably is high if the automated car has driven less than one meter, but after 10 m driven there is probably not much correlation is left that one could infer the safety of the automated car and its environment from the fact that it was safe 10 m back. At the same time, this approach eliminates the effect of the average traveling speed.

One could argue further that on many frames, no safety-relevant instance of other road users is given and one could potentially avoid labeling such frames. However, from the history of fatal accidents with the involvement of autopilots we learn that such accidents could even be triggered in situations considered to be non-safety-critical from a human perspective; see, e.g., [Nat20]. As direct measurement of safety should not be based on assumptions, unless they can be supported by evidence, we do not suggest introducing a discounting factor as we would have to assume without proof that we could separate hazardous from non-hazardous situations. This of course does not exclude that a refined methodology is developed in the future that is capable to provide this separation and we refer to the extensive literature on corner cases; see, e.g., [BBLFs19, BTLFs20, BTLFs21, HBR+21].

On the other hand, as we will present in Sect. 3, a statistically valid estimate of the frequency of rare events requires a leverage factor of at least 3–10 applied on the average number of frames per incident; see Sect. 3.1 for the details and precise values.

Further, in the presence of a clear trend of the reduction of fatalities in street traffic [Ger20] (potentially, partially due to AI-based assistance systems already), a mere reproduction of the present-day level of safety in driving does not seem to be sufficient. Without deeper ethical or scientific justification, we assume that humans expect at least 10–100 times lower failure rate of robots than they would concede themselves, while at the same time, we recommend further ethical research and political debate on this critical number. Even with a reduction number of 100, the approximately 30 fatalities due to autonomous vehicles would well exceed the risk

of being struck by lightning causing  $\sim 4$  fatalities per year in Germany,<sup>3</sup> which often is considered as a generally acceptable risk.

In addition, several failure modes exist aside from AI-based perception that cause fatalities in transportation. We therefore must not reserve the entire cake of acceptable risk to perception-related errors, only. Instead, here we suggest a fraction of  $\frac{1}{10}$  to  $\frac{1}{2}$  of the entire acceptable risk for perception-related root causes of fatalities. Also at this place, we recommend more serious research, ethical consideration, and public debate.

Drawing all this together, we obtain a total number of frames that ranges from 1.50 trillion frames in the best-case scenario to 23,000 trillion frames, or 1.5–23,000 exabyte (in the year of reference 2019, the entire Internet contained 33,000 exabyte<sup>4</sup> of data). This computation is summarized from Table 1.

Note that replacing fatalities with injuries reduces the amount of data by roughly a factor of one hundred (exact number for 2019 4/509) [Ger20].

Direct measurement of reliability of an AI-perception system requires labeled data, and the time to annotate a single frame by a human ranges from a couple of minutes for bounding box labeling up to 90 min for a fully segmented image [Sch19]. Working with the span of 5–90 min per annotated frame and a span of wages from a minimum wage of 9.19 Euro for Germany in our year of reference 2019 as lower bound to 15 EUR as upper bound, the cost of labeling a single image ranges from 0.775 to 22.5 EUR.

The total cost for labeling of test data to produce direct statistical evidence therefore ranges from 1.16 trillion in the best case to 51,800 trillion Euro in the worst case. This compares to 3.5 trillion Euro of Germany’s gross domestic product in 2019.

We conclude in agreement with [KP16] that direct and assumption-free statistical evidence of safety of the AI-based perception function of an automated vehicle that complies with the safety requirements derived from the ethics committee’s final report is largely infeasible with the present-day technology.

Certainly, this does not say anything about whether an AI-based system for automated driving actually *would be* driving better than human. Many experts, including the authors, believe it *could be*, at least in the long run. But the subjective belief of technology-loving experts—in the absence of direct evidence—is certainly insufficient to give credibility to the *promise* of enhanced safety due to automated driving in the sense of the ethics committee’s introductory statement.

This of course does not exclude that safety arguments which are based on indirect forms of evidence are reasonable and possible, if they are based on assumptions that can be and are carefully checked. In fact, in the following, we discuss one such potential strategy based on redundancy and report some problems and some progress with this approach applied to small, academic examples.

---

<sup>3</sup> <https://www.vde.com/de/blitzschutz/infos/bitzenfaelle-blitzschaeden#statistik>.

<sup>4</sup> Here, for clarity, we use powers of 10, e.g., 1000, instead of powers of 2, e.g., 1024.

**Table 1** Factors and numbers that influence the data requirement for a statistically sound assurance case by direct testing. Safety factors ( $\uparrow$ ) multiply and reduction factors ( $\downarrow$ ) divide the number of frames/cost

Description	$\uparrow\downarrow$	Quantity	Quantity	Unit	Source	Frames	Frames
		Lower Bound	Upper Bound			Lower Bound	Upper Bound
Meters to fatal accident (2019)		$2.50 \times 10^{11}$	$2.50 \times 10^{12}$	m	[Ger20] & ad hoc		
Meters driven per frame	$\downarrow$	1	10	m/f	ad hoc assumption	$2.50 \times 10^{10}$	$2.50 \times 10^{12}$
Factor for stat. evidence	$\uparrow$	2.99 $\alpha = 5\%$	9.21 $\alpha = 0.01\%$	factor	Sect. 3.1	$7.49 \times 10^{10}$	$2.30 \times 10^{13}$
Add. safety by autom. driving	$\uparrow$	10	100	factor	ad hoc assumption	$7.49 \times 10^{11}$	$2.30 \times 10^{15}$
Fraction of perception risk from total risk	$\downarrow$	$\frac{1}{10}$	$\frac{1}{2}$	factor	ad hoc assumption	$1.50 \times 10^{12}$	$2.30 \times 10^{16}$
						Cost (EUR) lower bound	Cost (EUR) upper bound
Labeling time per frame	$\uparrow$	5	90	min	[Sch19] & ad hoc		
Hourly wages	$\uparrow$	9.19	15	EUR/h	minimum wage GER 2019 & ad hoc		
Cost per frame	$\uparrow$	0.775	22.5	EUR/f	2 rows above		
Total cost				EUR	frames $\times$ cost per frame	$1.16 \times 10^{12}$	$5.18 \times 10^{17}$

### 3 Measurement of Failure Probabilities

#### 3.1 Statistical Evidence for Low Failure Probability

In this subsection, we provide the mathematical reasoning for the leverage factor of 2.99–9.21 that accounts for statistical evidence. Let us denote by  $p$  the actual probability of a fatal accident for one single kilometer of automated driving. We are looking for statistical evidence that  $p \leq p_{tol} = f_{tol} \cdot p_{human}$ , where  $f_{tol}$  is a debit factor for

enhanced safety of robots and multiple technical risks. From Table 1, we infer that  $f_{\text{tol}} \in [\frac{1}{1000}, \frac{1}{20}]$  taking into account the fraction of perception risk from total risk and the additional safety due to automated driving, cf. Sect. 2. Here,  $p_{\text{human}} \approx \frac{1}{250,000,000}$  is the (estimated, upper bound) probability of a fatal accident caused by a human driver per driven kilometer. With  $\hat{p} = \frac{N_{\text{obs}}}{N_{\text{test}}}$ , we denote the estimated probability of a fatal accident per kilometer driven for the autonomous vehicle based on the observed number of fatal accidents  $N_{\text{obs}}$  on  $N_{\text{test}}$  kilometers of test driving. We want to test for the alternative hypothesis  $H_1$  that  $p$  is below  $p_{\text{tol}}$  at a level of confidence  $1 - \alpha$  with  $\alpha \in (0, 1)$  a small number, e.g.,  $\alpha = 5\%, 1\%, 0.1\%, 0.01\%$  or even smaller. We thus assume the null hypothesis  $H_0$  that  $p \geq p_{\text{tol}}$ , using that under the null hypothesis  $N_{\text{obs}} \sim B(N_{\text{test}}, p_{\text{tol}})$  is Bernoulli-distributed with probability  $p_{\text{tol}}$  and  $N_{\text{test}}$  repetitions. The exact one-sided Bernoulli test rejects the null hypothesis and accepts  $H_1$  provided that

$$\begin{aligned} 1 - \alpha &\leq P_{N \sim B(N_{\text{test}}, p_{\text{tol}})}(N > N_{\text{obs}}) = 1 - P_{N \sim B(N_{\text{test}}, p_{\text{tol}})}(N \leq N_{\text{obs}}) \\ &= 1 - \sum_{j=0}^{N_{\text{obs}}} \binom{N_{\text{test}}}{j} (p_{\text{tol}})^j (1 - p_{\text{tol}})^{N_{\text{test}}-j}. \end{aligned} \quad (1)$$

The reasoning behind (1) is the following: Assume  $H_0$ , i.e., the true probability of a fatal accident due to the autonomous vehicle would be higher than  $p_{\text{tol}}$ . Then, with a high probability of at least  $1 - \alpha$  we would have seen more fatal accidents than just  $N_{\text{obs}}$ , which we actually observed. This puts us in front of the alternative to either believe that in our test campaign we just observed an extremely rare event of probability  $\alpha$ , or to discard the hypothesis  $H_0$  that the safety requirements are *not* fulfilled.

Let us suppose for the moment that the outcome of the test campaign is ideal, i.e., no fatal accidents are observed at all, i.e.,  $N_{\text{obs}} = 0$ . In this ideal case, (1) is equivalent to

$$\alpha \geq (1 - p_{\text{tol}})^{N_{\text{test}}} \Leftrightarrow -\frac{\ln(\alpha)}{N_{\text{test}}} \leq -\ln(1 - p_{\text{tol}}) \approx p_{\text{tol}}, \quad (2)$$

where we used the first-order Taylor series expansion of the natural logarithm at 1, which is highly precise as  $p_{\text{tol}}$  is small. Thus, even in the ideal case of zero fatalities observed,  $N_{\text{test}} \geq -\frac{\ln(\alpha)}{p_{\text{tol}}}$  is required. For  $\alpha$  ranging from 5% to 0.01%,  $-\ln(\alpha)$  roughly ranges from 3 (numerical value 2.9976) to 10 (numerical value 9.2103). This explains the back of the envelope estimates in Sect. 2.

Note that the approach of [KP16] differs as it is based on a rate estimate for the Poisson distribution. Nevertheless, as binomial and Poisson distribution for low probabilities approximate each other very well, this difference is negligible, as the difference is essentially proportional to the event of two or more fatal incidents in one kilometer driven.

### 3.2 Test Data for Redundant Systems

**Assuming independence of subsystems:** Let  $(\mathbf{x}, y)$  be a pair of random variables, where  $\mathbf{x} \in \mathcal{X}$  represents the input data presented to two neural networks  $h_1$  and  $h_2$ , and  $y \in \mathcal{Y}$  denotes the corresponding ground truth label. We assume that  $(\mathbf{x}, y)$  follows a joint distribution  $P$  possessing a corresponding density  $p$ . For each neural network, the event of failure is described by

$$\mathcal{F}_i := \{h_i(\mathbf{x}) \neq y\}, \quad i = 1, 2,$$

with  $1_{\mathcal{F}_i}$  being their corresponding indicator variables that are equal to one for an event in  $\mathcal{F}_i$  and zero else. If and only if we assume independence of the events  $\mathcal{F}_i$ , we obtain

$$\mathbb{E}[1_{\mathcal{F}_1} \cdot 1_{\mathcal{F}_2}] = P(\mathcal{F}_1 \cap \mathcal{F}_2) = P(\mathcal{F}_1) \cdot P(\mathcal{F}_2) = \mathbb{E}[1_{\mathcal{F}_1}] \cdot \mathbb{E}[1_{\mathcal{F}_2}],$$

which implies that the covariance fulfills

$$\text{COV}(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2}) = \mathbb{E}[1_{\mathcal{F}_1} \cdot 1_{\mathcal{F}_2}] - \mathbb{E}[1_{\mathcal{F}_1}] \cdot \mathbb{E}[1_{\mathcal{F}_2}] = 0.$$

This is easily extended to  $n$  neural networks  $h_i(\mathbf{x})$ ,  $i \in \mathcal{I} = \{1, \dots, n\}$  and their corresponding failure sets  $\mathcal{F}_i$ . Under the hypothesis of independence of the family of events  $\mathcal{F}_i$ , we obtain

$$p_{\text{system}} = \mathbb{E} \left[ \prod_{i \in \mathcal{I}} 1_{\mathcal{F}_i} \right] = \prod_{i \in \mathcal{I}} P(\mathcal{F}_i) = \prod_{i \in \mathcal{I}} p_{\text{sub},i},$$

where  $p_{\text{system}}$  is the probability of failure of a system of  $\#\mathcal{I} = n$  redundant neural networks working in parallel, where failure is defined as all networks being wrong at the same time [ME14] and  $p_{\text{sub},i} = P(\mathcal{F}_i)$  is the probability of failure for the  $i$ th subsystem  $h_i(\mathbf{x})$ .

Let us suppose for convenience that the probability for the subsystems  $h_i(\mathbf{x})$  are all equal,  $p_{\text{sub},i} = p_{\text{sub}}$ . Then  $p_{\text{system}} = p_{\text{sub}}^n$ . In order to give evidence that  $p_{\text{system}} < p_{\text{tol}}$ , it is thus enough to provide evidence for  $p_{\text{sub}} = p_{\text{sub},i} < p_{\text{tol}}^{\frac{1}{n}}$  for  $i \in \mathcal{I}$ . Subsystem testing to a confidence of  $(1 - \alpha)$  on the system level requires a higher confidence at the subsystem level, which, by a simple Bonferroni correction [HS16], can be conservatively estimated as  $(1 - \frac{\alpha}{n})$ . Consequently, by (2) the amount of data for testing the subsystem  $h_i(\mathbf{x})$  is given by

$$-\frac{\ln\left(\frac{\alpha}{n}\right)}{N_{\text{test},i}} > p_{\text{tol}}^{\frac{1}{n}} \quad \Leftrightarrow \quad N_{\text{test},i} > -\frac{\ln\left(\frac{\alpha}{n}\right)}{p_{\text{tol}}^{\frac{1}{n}}}. \quad (3)$$

As  $p_{\text{tol}}^{\frac{1}{n}}$  is much larger than  $p_{\text{tol}}$ , the amount of testing data required is radically reduced, even if one employs  $n$  separate test sets for all  $n$  subsystems. By comparison of (2) and (3), the factor of reduction is roughly

$$\gamma_n = \frac{N_{\text{test}}}{n N_{\text{test},i}} = \frac{1}{n p_{\text{tol}}^{1-\frac{1}{n}} \left(1 - \frac{\ln(n)}{\ln(\alpha)}\right)}.$$

For example, for  $n = 2$  in the best-case scenario,  $\alpha = 5\%$ , and  $f_{\text{tol}} = \frac{1}{20}$  the reduction factor is  $\gamma_2 = 28,712$ , which reduces the corresponding  $1.5 \times 10^{12}$  frames to 52.2 million frames. This already is no longer an absolutely infeasible number. For the worst-case scenario,  $\alpha = 0.01\%$  and  $f_{\text{tol}} = \frac{1}{1000}$ , the reduction factor is  $\gamma_2 = 232,502$ , resulting in 98.9 billion frames, which seems out of reach, but not to the extent of  $2.3 \times 10^{16}$  frames.

Keeping the other values fixed,  $n = 3$  even yields a reduction factor of  $\gamma_3 = 974,672$  in the best-case scenario and  $\gamma_3 = 11,818,614$  in the worst-case scenario, resulting in 1.54 million frames in the best and 1.95 billion frames in the worst-case scenario. These numbers look almost realizable, given the economic interests at stake.

However, the strategy based on redundant subsystems comes with a catch. It is only applicable, *if* the subsystems are independent. But this is an assumption that is not necessarily true. We therefore investigate what happens, if the errors of subsystems are not independent.

**Assuming no independence of subsystems:** In this case, the covariance of the error indicator variables  $1_{\mathcal{F}_i}$  is not equal to zero and can be regarded as a measure of the joint variability for the random variables  $1_{\mathcal{F}_i}$ . The normalized version of the covariance is the Pearson correlation

$$\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2}) = \frac{\text{COV}(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})}{\sigma(1_{\mathcal{F}_1}) \cdot \sigma(1_{\mathcal{F}_2})} \in [-1, 1],$$

where  $\sigma(1_{\mathcal{F}_i}) = \sqrt{p_{\text{sub},i}(1 - p_{\text{sub},i})}$  denotes the standard deviation of  $1_{\mathcal{F}_i}$ , which is supposed to be greater than zero for  $i = 1, 2$ . The correlation measures the linear relationship between the random variables  $1_{\mathcal{F}_i}$  and takes values  $\pm 1$  if the relationship between the random variables is deterministic.

Let us first consider a system with two redundant subsystems in parallel,  $h_i(\mathbf{x})$ ,  $i = 1, 2$ , where we however drop the assumption of independence. Then we obtain

$$\begin{aligned} p_{\text{system}} &= \frac{\mathbb{E}[1_{\mathcal{F}_1} \cdot 1_{\mathcal{F}_2}]}{\text{COV}(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2}) + \mathbb{E}[1_{\mathcal{F}_1}] \cdot \mathbb{E}[1_{\mathcal{F}_2}]} \\ &= \rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2}) \sqrt{p_{\text{sub},1}(1 - p_{\text{sub},1})} \sqrt{p_{\text{sub},2}(1 - p_{\text{sub},2})} + p_{\text{sub},1} p_{\text{sub},2}. \end{aligned} \quad (4)$$

Assuming again equal failure probabilities for the subsystems  $p_{\text{sub}} = p_{\text{sub},1} = p_{\text{sub},2}$  and using  $1 - p_{\text{sub}} \approx 1$  as a good approximation as  $p_{\text{sub}}$  for a safe system is very small, we obtain from (4)

$$p_{\text{system}} \approx \rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})p_{\text{sub}} + p_{\text{sub}}^2, \quad (5)$$

i.e., we can only expect a reduction of the frames needed for testing which is comparable to the case, where statistical independence holds, if  $\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})$  is of the same small order of magnitude as  $p_{\text{sub}}$ . If, e.g., we assume an extremely weak correlation of  $\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2}) = 0.01$ , we can essentially neglect the  $p_{\text{sub}}^2$ -term as  $p_{\text{sub}} \ll 0.01$  and realize that the reduction factor essentially is  $\frac{1}{\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})} = 100$ , only. Thus, even for such a pretty uncorrelated error scheme, the number of frames required for testing would be lower bounded by  $1.5 \times 10^{10}$  to  $2.3 \times 10^{14}$  frames, even neglecting Bonferroni correction and independent test sets which make up a multiplication factor  $B_2 = n(1 - \frac{\log(n)}{\log(\alpha)})$  yielding  $B_2 = 2.46$  and  $B_2 = 2.15$ , respectively. With these effects taken into account, we arrive at 36.9 billion frames in the best-case scenario and  $4.94 \times 10^{14}$  frames in the worst case, where even the lower number of frames seems hardly feasible.

A related computation for  $n = 3$  yields to leading order, using (5) and approximating the complement of small probabilities with one and neglecting terms of order  $p_{\text{sub}}^2$ ,

$$\begin{aligned} p_{\text{system}} &= \mathbb{E}[1_{\mathcal{F}_1} \cdot 1_{\mathcal{F}_2} \cdot 1_{\mathcal{F}_3}] \\ &\approx \rho(1_{\mathcal{F}_1 \cap \mathcal{F}_2}, 1_{\mathcal{F}_3}) \sqrt{\mathbb{E}[1_{\mathcal{F}_1} \cdot 1_{\mathcal{F}_2}] p_{\text{sub}}} + \mathbb{E}[1_{\mathcal{F}_1} \cdot 1_{\mathcal{F}_2}] p_{\text{sub}} \\ &\approx \rho(1_{\mathcal{F}_1 \cap \mathcal{F}_2}, 1_{\mathcal{F}_3}) \sqrt{(\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2}) p_{\text{sub}} + p_{\text{sub}}^2) p_{\text{sub}}} \\ &\quad + (\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2}) p_{\text{sub}} + p_{\text{sub}}^2) p_{\text{sub}} \\ &\approx \rho(1_{\mathcal{F}_1 \cap \mathcal{F}_2}, 1_{\mathcal{F}_3}) \sqrt{\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})} p_{\text{sub}}. \end{aligned} \quad (6)$$

If we thus assume that both correlations  $\rho(1_{\mathcal{F}_1 \cap \mathcal{F}_2}, 1_{\mathcal{F}_3})$  between the failure of subsystem  $h_3(\mathbf{x})$  and the composite redundant subsystem from  $h_3(\mathbf{x})$  and  $h_2(\mathbf{x})$  are both equal to 0.01, we obtain a total reduction factor of roughly  $\frac{1}{\rho(1_{\mathcal{F}_1 \cap \mathcal{F}_2}, 1_{\mathcal{F}_3}) \sqrt{\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})}} = 1,000$ , which still leads to roughly  $1.50 \times 10^9 - 2.30 \times 10^{13}$  frames, even without Bonferroni correction and independent test sets for subsystems. With both taken into account, the amount of data ranges from 5,04 billion frames in the best-case scenario to  $9.43 \times 10^{13}$  frames in the worst case, where only the figure obtained in the best case, based on problematic choices, seems remotely feasible. However, in the presence of domain shifts in time and location, it seems questionable if the road of testing weakly correlated subsystems is viable (supposed they are weakly correlated).

We also note that correlation coefficients as low as  $\rho = 0.01$  are rarely found in nature and in addition, it requires empirical testing to provide evidence for a low correlation. The correlations we measure in Sect. 4 for the case of simple classification

problems miss this low level by at least an order of magnitude, leading to an extra factor of at least 10 in the above considerations.

### 3.3 Data Requirements for Statistical Evidence of Low Correlation

In the preceding section, we have analyzed that low correlation between subsystems efficiently reduces the data required for testing better-than-human safety of autonomous vehicles. However, to achieve this, e.g., in the case of two redundant subsystems, the correlation has to be in the order of magnitude of  $p_{\text{sub}} = p_{\text{tol}}^{\frac{1}{2}}$ . Statistical evidence for such a low level of correlation requires data itself. Let us suppose the ideal situation once more that a correlation coefficient is strictly zero  $\rho = 0$  and we would like to compute the number of pairs of observations of the random variables  $(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})$  that is needed to prove that  $\rho$  is in the order of magnitude  $p_{\text{sub}}$ , as required for a decent downscaling of the number of test data frames. In other words, we have to estimate a number of samples needed to provide statistical evidence at a given significance level  $\alpha$  that  $\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})$  is less than  $p_{\text{sub}} = p_{\text{tol}}^{\frac{1}{2}}$ .

As shown by Raymond Fisher and others, see, e.g., [LL88], the quantity  $\hat{Z} = \frac{1}{2} \log \left( \frac{1+\hat{\rho}}{1-\hat{\rho}} \right)$  is asymptotically normally distributed with expected value  $\frac{1}{2} \log \left( \frac{1+\rho}{1-\rho} \right) = 0$  in our case, where we assumed  $\rho = 0$ . The standard deviation is given by  $\sqrt{\frac{1}{N_{\text{test}} - 3}}$ . Here,  $\hat{\rho}$  stands for the empirical correlation coefficient of the pair of observations [HS16].

A two-sided confidence interval for a given level of confidence  $1 - \alpha$  for the observed value  $\hat{z}$  of  $\hat{Z}$  thus is given by

$$\hat{z}_{\pm} = \hat{z} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{N_{\text{test}} - 3}}, \quad (7)$$

where  $z_{1-\frac{\alpha}{2}}$  is the  $1 - \frac{\alpha}{2}$  quantile of the standard normal distribution. Transforming back (7), we obtain lower and upper bounds

$$\hat{\rho}_{\pm} = \frac{\exp(2\hat{z}_{\pm}) - 1}{\exp(2\hat{z}_{\pm}) + 1}. \quad (8)$$

Under our best-case hypothesis  $\rho = 0$ , the boundaries  $\hat{z}_{\pm}$  of the confidence interval converge to zero; we may apply the  $\delta$ -rule with the derivative  $\frac{d}{dz} \frac{\exp(2z)-1}{\exp(2z)+1} \Big|_{z=0} = \frac{4 \exp(2z)}{(\exp(2z)+1)^2} \Big|_{z=0} = 1$ . Let us consider the width  $W$  of the confidence interval for  $\rho$  for the best possible outcome obtained for  $\hat{z} = 0$ . By (8) and the  $\delta$ -rule, asymptotically for large  $N_{\text{test}}$  it is given by  $W = 2z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{N_{\text{test}} - 3}}$ . Even if this is the case, to infer

that  $|\rho| \leq p_{\text{sub}}$  with confidence  $1 - \alpha$ , one requires, for the case of two subsystems,  $z_{1-\frac{\alpha}{2}} \sqrt{\frac{1}{N_{\text{test}}-3}} \leq p_{\text{sub}} = p_{\text{tol}}^{\frac{1}{2}}$ . If  $N_{\text{test}}$  is large, we can neglect the  $-3$  term and obtain for the best case

$$N_{\text{test}} \approx \frac{z_{1-\frac{\alpha}{2}}^2}{p_{\text{tol}}}.$$
 (9)

Not unexpectedly, this brings back the bad scaling behavior observed in (2) and the related problematic data requirements, which are essentially the same as for the non-redundant, direct approach. The numbers for  $z_{1-\frac{\alpha}{2}}^2$  for  $\alpha = 5\% \dots \alpha = 0.01\%$  range from 2.706 to 13.831 which essentially confirms the range of roughly 3 ... 10 for the statistical safety factor obtained from (1) and (2).

## 4 Correlation Between Errors of Neural Networks in Computer Vision

As of now, deep neural networks (DNNs) for perception tasks are far away from being perfect. Motivated by common practices in reliability engineering, redundancy, i.e., the deployment of multiple system components pursuing the same task in parallel, might be one possible approach toward improving the reliability of a perception system.

Redundancy can enter into perception systems in many ways. Assume a system setup with multiple sensors, e.g., camera, LiDAR, and RaDAR. There are multiple options to design a deep-learning-driven perception system processing the different sensors' data. A non-exhaustive list of designs may look as follows:

1. Only a single sensor is processed; this is done by a single DNN;
2. Only a single sensor is processed; this is done by a committee of DNNs;
3. All sensors are processed by a single-sensor-fusing DNN;
4. All sensors are processed by a committee of sensor-fusing DNNs;
5. Each sensor is processed by a separate DNN; the results are fused afterwards;
6. Each sensor is processed by a committee of DNNs; the results are fused afterwards.

Except for the first design, all other designs incorporate redundancy. Herein, there are two types of redundancy, redundancy via multiple sensors (all pursuing the same task of perceiving the environment) and redundancy via multiple DNNs.

Certainly, approach one is only eligible for direct testing, see Sect. 3.1, and the same is true for the “early fusion” approach 3. All the other approaches could potentially benefit from redundancy, if independence or low correlation of the errors can be assumed. Therefore, the degree of independence, which can be understood and quantified as the degree of uncorrelatedness, is a quantity of interest for safety and also for testing; see Sect. 3.2. However, as explained in Sect. 3.2, in order to use redundancy as a part of the solution to the testing problem outlined in Sect. 2, correlation has to be extremely low.

For the case of simple DNNs processing the same sensor’s data, we give evidence that such low correlation in general does not hold. The evidence we find rather points in the opposite direction that it is hard to obtain a correlation that is below 0.5, even if the training datasets and network architecture are kept well separated. On the other hand, one could try to train DNNs such that their failure events are uncorrelated.

In this section, we show preliminary numerical results on MNIST (handwritten digits), EMNIST (also containing handwritten letters), and 3D-MNIST for

- training DNNs for independence/less correlated failure events;
- the role of different sensors (by viewing 3D-MNIST examples from different directions).

Although our findings do not directly apply to physically more diverse sensors like Camera, LiDAR, and RaDAR, these preliminary results indicate that the independence of DNN-based perception systems cannot be taken simply for granted, even if different sources of data are employed.

## 4.1 Estimation of Reliability for Dependent Subsystems

Most commonly, the so-called active parallel connection of  $n$  subsystems is used, wherein the entire system (meaning the ensemble of DNNs) is assumed to be functional iff at least one subsystem (which corresponds to one committee member  $h_i$ ) is functional. However, the active parallel connection is not the only decision rule of interest which can be applied to the committee  $h_i$ ,  $i = 1, \dots, n$ . For instance, considering a pedestrian detection performed by a committee  $h_i$  that detects a pedestrian if at least one of the DNNs does so. For increasing  $n$ , we would expect an increase in false positive detections, therefore facing the typical trade-off of false positives and false negatives. In order to steer this trade-off, we use  $k$ -out-of- $n$  systems that are functional iff at least  $k$  out of  $n$  components  $h_i$  are functional, i.e., at most  $n - k$  components fail. Hence, we are interested in the event

$$\left\{ \sum_{i=1}^n 1_{\mathcal{F}_i} < n - k \right\}$$

and its probability which is the probability of the  $k$ -out-of- $n$  system being functional. The reliability of  $k$ -out-of- $n$  system can be expressed analytically in terms of the reliability of its components; see also [Zac92]. For  $k = 1$ , this boils down to the active parallel connection.

If we assume independence of the failure events  $\mathcal{F}_i$  and that all networks fail with equal probability  $P(\mathcal{F}_i) = p_{\text{sub},i} = p_{\text{sub}}$ , then the probability that at least  $k$ -out-of- $n$  networks are functional can be calculated via

$$P\left(\sum_{i=1}^n \mathbb{1}_{\mathcal{F}_i} < n - k\right) = \sum_{j=k}^n \binom{n}{j} (1 - p_{\text{sub}})^j \cdot p_{\text{sub}}^{n-j} = 1 - F_B(k-1; n, 1 - p_{\text{sub}}), \quad (10)$$

where  $F_B$  denotes the distribution function of the binomial distribution. This quantity serves as a reference in our experiments.

## 4.2 Numerical Experiments

In this section, we conduct the first experiments using the datasets EMNIST, CIFAR10, and 3D-MNIST. The original dataset MNIST [LCB98] contains 60,000 grayscale images of size  $28 \times 28$  displaying handwritten digits (10 classes). EMNIST is an extension of MNIST that contains handwritten digits and characters of the same  $28 \times 28$  resolution. Of that dataset, we only considered the characters (26 classes) of which there are 145,600 available. We used 60,000 images for training, 20,000 for validation, and the rest for testing. CIFAR10 contains 60,000 RGB images of size  $32 \times 32$  categorized into 10 classes (from the categories animals and machines). We used the default split of 50,000 training and 10,000 test examples. We reserved a quarter of the training set for validation. 3D-MNIST contains point clouds living on a  $16^3$ -lattice. The dataset contains 10,000 training and 2,000 test examples.

We used convolutional DNNs implemented in Keras [C+15] with simple architectures; if not stated otherwise they contain 2 convolutional layers with 32 and 64  $3 \times 3$ -filters, respectively, each of them followed by a leakyReLU activation and  $2 \times 2$  max pooling, and finally a single dense layer. For training, we used a batch size of 256, weight decay of  $10^{-4}$ , and the Adam optimizer [KB15] with default parameters, except for the learning rate. We started with a learning rate of  $10^{-2}$ , trained until stagnation, and repeated that with a learning rate of  $10^{-3}$ .

**Reducing correlations with varying training data, architecture, and weight initializers:** First, we study to what extent independence can be promoted by varying the training data, architecture, and weight initializers in an active parallel system with  $n = 2$  DNNs. To this end, we split the training data and validation data into two disjoint chunks; consider another network, where we add a third convolutional layer with 128 filters, again followed by leakyReLU and max pooling, and use the Glorot uniform and Glorot normal initializers with default parameters. For the sake of brevity, we introduce a short three-bit notation indicating the boolean truth values corresponding to the questions

$$(\text{same data? same architecture? same initializer?}). \quad (11)$$

For instance, 101 stands for two DNNs being trained with the same data, having different architectures, and using the same initializer. We report results in terms of *average accuracy* estimating  $\frac{1}{n} \sum_{i=1}^n (1 - P(\mathcal{F}_i))$  and *joint accuracy* estimating  $1 - P(\bigcap_{i=1}^n \mathcal{F}_i)$ .

**Table 2** Correlation coefficients and average accuracies for EMNIST and CIFAR10. The configurations read as defined in (11). All runs have been performed 10 times, all numbers are averaged over these 10 runs, and the standard deviation over these 10 runs are given

Configurations	EMNIST		CIFAR10	
	$\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})$	avg. acc. (%)	$\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})$	avg. acc (%)
111	$0.71 \pm 0.01$	$91.17 \pm 0.05$	$0.73 \pm 0.01$	$72.27 \pm 0.45$
110	$0.71 \pm 0.00$	$91.17 \pm 0.06$	$0.74 \pm 0.01$	$72.19 \pm 0.22$
101	$0.71 \pm 0.01$	$91.13 \pm 0.04$	$0.74 \pm 0.01$	$72.08 \pm 0.43$
100	$0.71 \pm 0.01$	$91.13 \pm 0.05$	$0.74 \pm 0.01$	$72.14 \pm 0.36$
011	$0.58 \pm 0.01$	$89.65 \pm 0.14$	$0.66 \pm 0.02$	$66.10 \pm 0.74$
010	$0.58 \pm 0.01$	$89.76 \pm 0.08$	$0.65 \pm 0.01$	$66.08 \pm 0.35$
001	$0.57 \pm 0.01$	$89.74 \pm 0.07$	$0.66 \pm 0.01$	$66.29 \pm 0.65$
000	$0.58 \pm 0.01$	$89.62 \pm 0.13$	$0.65 \pm 0.01$	$66.30 \pm 0.38$

Table 2 shows in all cases correlation coefficients much greater than zero. Corresponding  $\chi^2$  tests with significance level  $\alpha = 0.05$  in all cases rejected the hypothesis that the DNNs are independent. Noteworthily, varying the initializer or the network architecture barely changes the results while changing the training data seems to have the biggest impact, clearly reducing the correlation coefficient. A reduction in correlation also reduces the average performance of the networks. For the sake of comparability, all networks were only trained with half of the training data, since otherwise the configurations with equal data would have the advantage of working with twice the amount of training data compared to the configuration with different training data.

Next, we study in this setting whether we can achieve at least conditional independence. To this end, we aim at conditioning the difficulty of the task by conditioning to softmax entropy quantiles. More precisely, we compute the entropy of the softmax distribution of all data points of both networks. We then sum the entropy values for each data point over the two networks and group all examples into 8 equally sized bins according to ascending summed entropy.

Table 3 shows that the higher the softmax entropy gets, the less the DNNs failures are correlated. This goes down to correlation coefficients of 0.25 for EMNIST and 0.23 for CIFAR10, when considering the softmax entropy bin no. 8 with the highest entropy values. Still,  $\chi^2$  tests reveal that the correlations are too strong to assume independent failures.

**Training two networks for enhanced independence:** Since the measures considered so far do not lead to success, we explicitly try to decorrelate the failures of the DNNs. To this end, we incorporate an additional loss function into training, added to the typically used empirical cross-entropy loss. Let  $p(y|\mathbf{x}, h_i)$  denote the probability estimated by the DNN  $h_i$  that the class  $y$  is the correct one given the input  $\mathbf{x}$ . One possible approach is to explicitly enforce a prediction of  $h_1$  different from that of  $h_2$

**Table 3** Correlation coefficients  $\rho(1_{\mathcal{F}_1}, 1_{\mathcal{F}_2})$  for different quantiles of softmax entropy computed on EMNIST and CIFAR10. The configurations read as defined in (11). The experiments have been repeated 10 times; the corresponding standard errors are of the order of 0.01

	EMNIST								CIFAR10							
Entropy bin	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Config.	Correlation coefficients															
111	1.0	1.0	1.0	1.0	0.99	0.94	0.71	0.45	1.0	0.99	0.94	0.79	0.64	0.57	0.45	0.26
110	1.0	1.0	1.0	1.0	0.99	0.95	0.71	0.45	1.0	1.0	0.96	0.79	0.67	0.55	0.44	0.31
101	1.0	1.0	1.0	1.0	0.99	0.94	0.71	0.43	1.0	0.99	0.94	0.78	0.67	0.55	0.43	0.36
100	1.0	1.0	1.0	0.99	1.0	.095	0.71	0.45	1.0	0.99	0.95	0.79	0.64	0.51	0.46	0.36
011	1.0	1.0	0.99	0.98	0.90	0.71	0.40	0.28	0.99	0.94	0.81	0.65	0.6	0.44	0.37	0.27
010	1.0	1.0	1.0	0.98	0.89	0.73	0.43	0.26	1.0	0.95	0.82	0.64	0.5	0.44	0.40	0.25
001	1.0	1.0	0.98	0.96	0.90	0.73	0.40	0.25	0.99	0.94	0.83	0.63	0.54	0.43	0.37	0.27
000	1.0	1.0	0.99	0.98	0.90	0.72	0.42	0.27	0.99	0.95	0.80	0.65	0.53	0.45	0.35	0.23

if the latter fails and vice versa. This can be achieved by minimizing the following quantity:

$$-\mathbb{E}_{(\mathbf{x}, y) \sim P}[1_{h_i(\mathbf{x}) \neq y} \log(1 - p(h_i(\mathbf{x})|\mathbf{x}, h_j)) + 1_{h_j(\mathbf{x}) \neq y} \log(1 - p(h_j(\mathbf{x})|\mathbf{x}, h_i))]$$

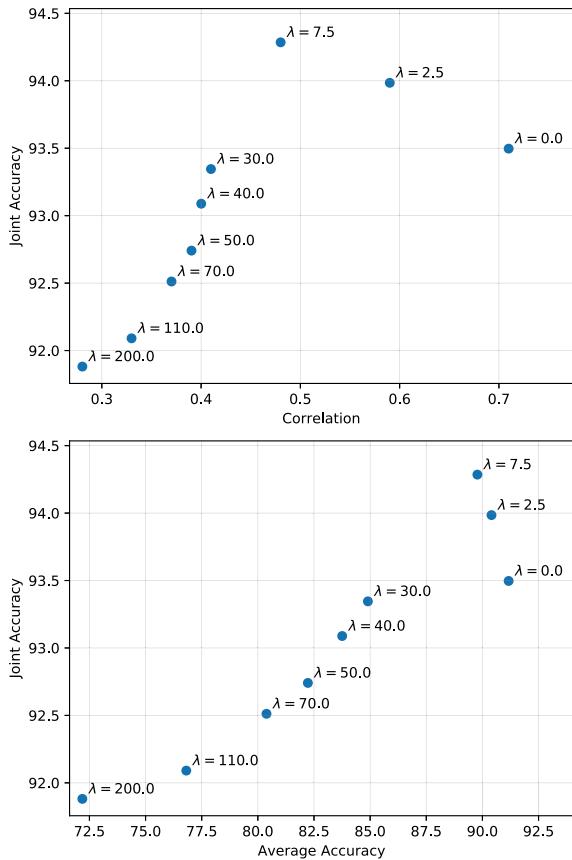
with its empirical counterpart

$$\begin{aligned} J_{i,j}(\{(\mathbf{x}_m, y_m)\}_{m=1,\dots,M}) = & -\frac{1}{M} \sum_{m=1}^M 1_{h_i(\mathbf{x}_m) \neq y_m} \log(1 - p(h_i(\mathbf{x}_m)|\mathbf{x}_m, h_j)) \quad (12) \\ & + 1_{h_j(\mathbf{x}_m) \neq y_m} \log(1 - p(h_j(\mathbf{x}_m)|\mathbf{x}_m, h_i)), \end{aligned}$$

where  $\{(\mathbf{x}_m, y_m)\}_{m=1,\dots,M}$  denotes a sample of  $M$  data points  $(\mathbf{x}_m, y_m)$ . In our experiments, we use a penalization coefficient/loss weight  $\lambda$  and then we add  $\lambda \cdot J_{1,2}(\{(\mathbf{x}_m, y_m)\}_{m=1,\dots,M})$  to the empirical cross-entropy loss. In further experiments not presented here, we also used other loss functions that explicitly enforce anti-correlated outputs or even independence of the softmax distributions. However, these loss functions led to uncontrollable behavior of the training pipeline, in particular when trying to tune the loss weight  $\lambda$ . Therefore, we present results for the loss function in (12).

Figure 1 (top) depicts the correlation as well as the average accuracy for different values of  $\lambda$ , ranging from 0 to extreme values of 200. For increasing values of  $\lambda$ , we observe a clear drop in performance from an initial average accuracy of more than 91% for  $\lambda = 0$  down to roughly 72% for  $\lambda = 200$ . At the same time, the correlation decreased to values below 0.3 which, however, is still not enough to assume independence as being confirmed by our  $\chi^2$  tests. While the average accuracy monotonously decreases, it can be observed that the joint accuracy peaks around  $\lambda = 7.5$ ; see Fig. 1 (bottom). The prediction of the DNN committee is pooled by summing up softmax

**Fig. 1** Study of the influence of the loss weight  $\lambda$  on averaged accuracy and joint accuracy on the EMNIST dataset



probabilities over both DNN class-wise and then selecting the class with maximal sum. The joint accuracy is given by the accuracy of the committee with that decision rule. While the joint accuracy for an ordinarily trained committee with  $\lambda = 0$  is about 93.5%, this can be improved by tenderly decorrelating the DNNs to a correlation coefficient around 0.5 which yields an increase of almost 1% point. At the same time, there is a mild decrease in average accuracy.

Concluding this section, we again study conditional independence for 8 softmax entropy bins (chosen according to equally distributed softmax entropy quantiles), this time comparing independence training  $\lambda = 7.5$  with ordinary training  $\lambda = 0$ ; see Table 4. We observe a considerable decrease in correlation in bin no. 8 (representing the highest softmax entropy) for the EMNIST data down to 0.07. In comparison, the decrease for CIFAR10 is rather mild. However, also this small correlation coefficient of 0.07 is not sufficient for assuming conditional independence.

**Table 4** Correlation coefficients under independence training for different quantiles of softmax entropy computed on EMNIST and CIFAR10

Entropy bin	EMNIST								CIFAR10							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
$\lambda$	Correlation coefficients															
7.5	1.0	1.0	0.99	0.96	0.90	0.59	0.35	0.07	0.89	0.76	0.63	0.53	0.47	0.39	0.32	0.21
0.0	1.0	1.0	1.0	1.0	0.99	0.95	0.72	0.44	1.0	0.99	0.94	0.80	0.64	0.57	0.42	0.33

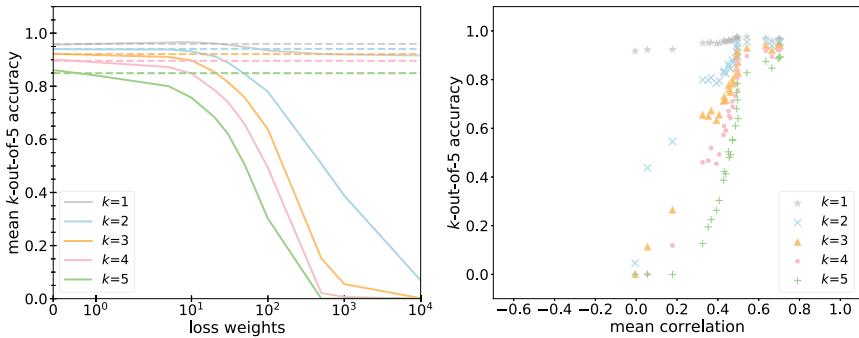
**Training several networks for independence:** We now present results when training an ensemble of  $n = 5$  networks. To this end, we sum up the cross-entropy losses of the  $n = 5$  models and add the penalization term

$$\lambda \cdot \frac{2}{n-1} \sum_{i=2}^n \sum_{j=1}^{i-1} J_{i,j}(\{(\mathbf{x}_m, y_m)\}_{m=1,\dots,M}) \quad (13)$$

which is a straightforward combinatorial generalization of the previously used penalty term. Therein,  $\lambda$  again denotes the loss weight which varies during our experiments.

Besides  $k$ -out-of-5 accuracies, we consider also accuracies from single networks and ensemble accuracies. The corresponding ensemble prediction is obtained by summing up the softmax probabilities (via the class-wise sum over the first  $k$  ensemble members) and then taking the argmax.

Figure 2 depicts the results of our training for independence with 5 models in terms of  $k$ -out-of-5 accuracy. When stating mean accuracy, this refers to the average

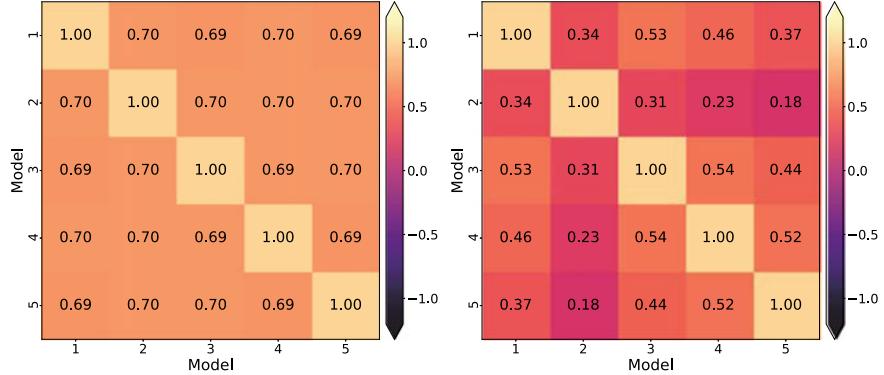


**Fig. 2** Experiments with the EMNIST dataset. Left: mean  $k$ -out-of-5 accuracy (averaged over 30 repetitions) for an ensemble of five networks as a function of the loss weight  $\lambda$ . The solid lines depict the accuracies of the ensembles trained for independence with loss weight  $\lambda$ , the dashed lines depict baseline ensemble accuracies trained without incorporating the loss from (13). Right:  $k$ -out-of-5 accuracy for a single run as a function of the mean correlation (of each network with each other network)

**Table 5** Correlation coefficients for an ensemble of 5 networks trained on EMNIST for independence and a baseline ensemble where each network was trained individually. All results are averaged over 30 runs

Loss weight $\lambda$		0	$10^{-1}$	$10^0$	$10^1$	$10^2$	Baseline model	Theoretical
Mean correlation		0.70	0.69	0.65	0.53	0.43	0.67	0.00
Single network accuracy	$k = 1$	0.92	0.92	0.91	0.88	0.73	0.91	
	$k = 2$	0.92	0.92	0.91	0.91	0.91	0.91	
	$k = 3$	0.92	0.92	0.91	0.88	0.60	0.91	
	$k = 4$	0.92	0.92	0.91	0.87	0.51	0.91	
	$k = 5$	0.92	0.92	0.91	0.85	0.39	0.91	
Ensemble accuracy/Mean $k$ -out-of-5 accuracy	$k = 1$	0.92/0.96	0.92/0.96	0.91/0.96	0.88/0.97	0.73/0.93	0.91/0.96	1.00
	$k = 2$	0.92/0.94	0.92/0.94	0.92/0.94	0.92/0.93	0.91/0.78	0.92/0.94	0.99
	$k = 3$	0.92/0.92	0.92/0.92	0.92/0.92	0.92/0.90	0.90/0.64	0.92/0.92	0.96
	$k = 4$	0.93/0.90	0.93/0.90	0.92/0.89	0.92/0.85	0.90/0.49	0.92/0.90	0.72
	$k = 5$	0.93/0.86	0.93/0.86	0.92/0.84	0.92/0.76	0.90/0.30	0.92/0.85	0.32

of over 30 runs. For a loss weight of  $\lambda = 10^1$ , we observe in the left panel that the 1-out-of-5 accuracy of our independence training is slightly above the accuracy of the baseline ensemble which was trained regularly, i.e., each network was trained independently. Note that this is different from  $\lambda = 0$ , where all networks are still trained jointly with a common loss function which is the sum of the cross-entropy losses. The right-hand panel shows that the 1-out-of-5 accuracy peaks at a mean correlation (the average correlation over the errors of all networks  $i < j$ ) of 0.4. Decorrelating the networks' errors further toward zero-mean correlation is possible, however, the 1-out-of-5 accuracy decreases. The  $k$ -out-of-5 accuracy for  $k > 1$  suffers even more from decorrelating the networks' errors. Note that, in practice, 1-out-of-5, e.g., for a pedestrian detection, might additionally suffer from overproduction of false positives and could be impractical. That hypothesis is indeed supported by Table 5, which shows that for larger loss weights  $\lambda \geq 1$  the individual network accuracies become heterogeneous. In particular, network 5 suffers from extremely low accuracy at  $\lambda = 10^2$ , which is, however, still far away from zero-mean correlation. For the sake of completeness, we give two examples of correlations between the individual models since we only reported mean correlations so far; see Fig. 3. Comparing the right-hand panel with Table 5, we see that the well-performing network no. 2 exhibits comparatively small correlation coefficients with the other networks' errors. Surprisingly, the worse performing models' errors show higher correlation coefficients which reveals that in that case they often err jointly on the same input



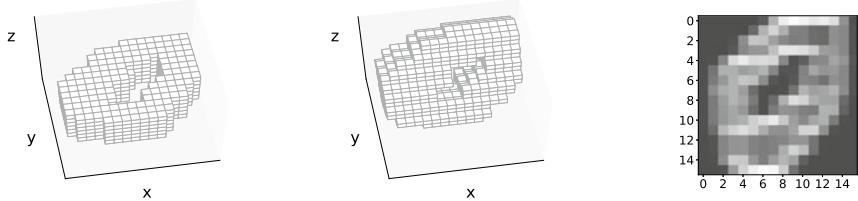
**Fig. 3** Correlation coefficients of the networks’ errors for all combinations of 2 out of 5 models. Left: loss weight  $\lambda = 10^{-1}$ . Right: loss weight  $\lambda = 10^2$

examples. Additionally, we provide theoretical  $k$ -out-of-5 accuracies according to (10) where we choose  $p_{\text{sub}}$  equal to 1 minus the average ensemble accuracy (which is 92.45%). In particular, the  $k$ -out-of-5 accuracies of the ensemble for  $k = 1$  and 2 are clearly below the theoretical ones.

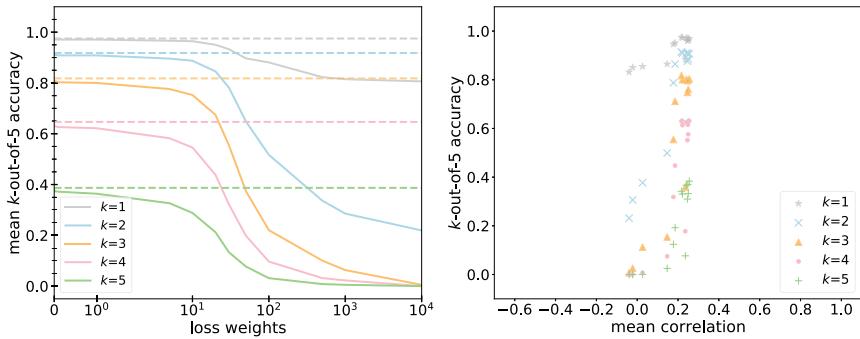
We conclude that independence training may help slightly to improve the performance, however, the benefit seems to be limited when all networks are trained with the same input. Besides these tests, we considered an additional loss function that explicitly penalizes the correlation of the networks’ errors. That approach, being actually more directed toward our goal, even achieved negative correlation coefficients of the networks’ errors. It was also able to slightly improve the ensembles’ performance in terms of  $k$ -out-of- $n$  accuracy over the baseline, however, this improvement was even less pronounced than that one reported in this section. Thus, we do not report those results here.

**Training of independence for different input sensors:** In order to conduct further experiments on the scale of the EMNIST and CIFAR10 datasets, we consider the 3D-MNIST dataset that provides synthesized 3D representations of handwritten digits. We apply rotations around the  $x$ -axis with chosen angles. To this end, we create 5 rotated variants of the dataset with a randomly chosen but fixed angles  $\theta = a\pi/9$ ,  $a = 1, \dots, 9$ ; see Fig. 4 for an illustrative example. Each of our  $k = 1, \dots, 5$  networks obtains one of the 5 rotated variants of the data with a given angle  $\theta_k$ , and is then trained. For the baseline, all networks are again trained independently. For independence training, all networks are trained with common loss functions, as described previously, and the same handwritten digit, however, viewed from a different angle  $\theta_k$ , is presented to the network  $k$ ,  $k = 1, \dots, 5$ .

Figure 5 shows results of numerical experiments conducted analogously to those presented in Fig. 2. Comparing both figures with each other, we observe that the baseline  $k$ -out-of-5 accuracies are much lower than when presenting identical images to the networks. Indeed, although MNIST classification constitutes a much simpler



**Fig. 4** A 3D-MNIST example data point. Left: original input data. Center: input data rotated by  $\pi/3$  along the  $x$ -axis. Right: 2D projection of the rotated data, which is obtained by summation and normalization along the  $y$ -axis



**Fig. 5** Experiments with the 3D-MNIST dataset. Left: mean  $k$ -out-of-5 accuracy (averaged over 30 repetitions) for an ensemble of five networks as a function of the loss weight  $\lambda$ . The solid lines depict the accuracies of the ensembles trained for independence with loss weight  $\lambda$ , the dashed lines depict baseline ensemble accuracies trained without incorporating the loss from (13). Right:  $k$ -out-of-5 accuracy for a single run as a function of the mean correlation (of each network with each other network)

task than classifying the letters from EMNIST; this 3D variant shows much lower individual network performances, not only indicated by the left-hand panel of Fig. 5, but also by Table 6. On the other hand, the highest mean correlation (obtained for the loss weight  $\lambda = 0$ ) for 3D-MNIST depicted by the right-hand panel of Fig. 5 is below 0.3 and therefore much lower than the one depicted by Fig. 2.

Neglecting the reduced performance on 3D-MNIST, these results show that an ensemble, wherein each network obtains a different input, can be clearly improved by increasing the number of ensemble members. The networks only exhibit small correlations among each other. However, it seems that independence training cannot contribute to the performance of the ensemble anymore in the given setup. It remains open, whether independence training may help in a realistic setting with different sensors for perception in automated driving and networks conducting way more difficult tasks such as object detection, instance segmentation, semantic segmentation, or panoptic segmentation. Note that the correlation strengths reported in our experiments in accordance with Sect. 3 do not suffice to substantially reduce the data requirements into a feasible regime. Recalling the discussion in Sect. 3.2, even

**Table 6** Results for an ensemble of 5 networks trained on 3D-MNIST for independence and a baseline ensemble where each network was trained individually. All results are averaged over 30 runs

Loss weight $\lambda$		0	$10^{-1}$	$10^0$	$10^1$	$10^2$	Baseline model	Theoretical
Mean correlation		0.25	0.25	0.24	0.22	0.15	0.24	0.00
Single network accuracy	$k = 1$	0.70	0.70	0.70	0.66	0.38	0.70	
	$k = 2$	0.74	0.73	0.74	0.73	0.75	0.75	
	$k = 3$	0.73	0.72	0.72	0.67	0.25	0.74	
	$k = 4$	0.81	0.80	0.79	0.73	0.17	0.82	
	$k = 5$	0.72	0.73	0.71	0.64	0.20	0.74	
Ensemble accuracy/mean $k$ -out-of-5 accuracy	$k = 1$	0.70/0.97	0.70/0.97	0.70/0.97	0.66/0.96	0.38/0.88	0.70/0.97	1.00
	$k = 2$	0.82/0.91	0.82/0.91	0.82/0.91	0.81/0.89	0.77/0.52	0.82/0.92	0.98
	$k = 3$	0.84/0.81	0.84/0.80	0.84/0.80	0.83/0.75	0.78/0.22	0.85/0.82	0.90
	$k = 4$	0.87/0.63	0.86/0.63	0.86/0.62	0.85/0.55	0.79/0.10	0.87/0.65	0.63
	$k = 5$	0.88/0.38	0.88/0.37	0.88/0.37	0.86/0.29	0.79/0.03	0.89/0.39	0.24

if the subsystem performance remained unaffected by independence training, based on the lowest correlation observed in our experiments, we could at most expect a reduction of the required data amount by one order of magnitude.

## 5 Conclusion and Outlook

**Summary and main take-away messages:** In this chapter, we argued that obtaining statistical evidence from brute-force testing leads to the requirement of infeasible amounts of data. In Sects. 2 and 3, we estimated upper and lower bounds on the amount of data required to test a given perception system with statistical validity for being significantly safer than a human driver. We restricted our considerations to fatalities and arrived at data amounts that are infeasible from both storage and labeling cost perspectives, as already found in [KP16]. In Sect. 3.1, we showed that perfectly uncorrelated redundant AI perception systems could be used to resolve the problem. However, as we have seen in Sect. 3.2, redundant subsystems that are not perfectly uncorrelated require extremely low correlation coefficients between error events produced by neural networks to yield substantial reductions in data requirement. In Sect. 3.3, we furthermore found the amount of data needed to prove

a sufficiently low correlation as large as the amount of data needed for the original test.

In Sect. 4, we present numerical results on the correlation of redundant subsystems. We studied correlation coefficients between error events of redundant neural networks dependent on network architecture, training data, and weight initializers. Furthermore, we trained ensembles of neural networks to become decorrelated. Besides studying correlation coefficients, we considered the system’s performance in terms of 1-out-of- $n$  as well as  $k$ -out-of- $n$  performance. In the numerical experiments, we obtained correlation coefficients that would allow for a reduction in the amount of data required by at most one order of magnitude. For the testing problem, this would still represent an infeasible data requirement. However, redundancy could contribute to a moderate reduction of the amount of data required for testing and potentially be combined with other approaches.

There are alternative approaches to test the safety of perception systems other than brute-force testing. Here, we give a short outlook on two of them that are very actively developed in the research community.

**Outlook on testing with synthetic data:** One possibility to obtain vast amounts of data for testing is to consider synthetic sources of data. The question of whether synthetic data can be used for testing has already been addressed, e.g., in [RBK+21]. In principle, arbitrary amounts of test data can be created from a driving simulation such as CARLA [DRC+17]. The domain shift between synthetic and real data can be bridged with generative adversarial networks (GANs). The latter have been proven to be learnable in the large sample limit [BCST20, AGLR21], meaning that for increasing amounts of data and capacity of the generator and discriminator, the learned map from synthetic to real data is supposed to converge to the true one. Combining synthetic data and adversarial learning is therefore a promising candidate for testing DNNs. However, in this setup there remain other gaps to be bridged (number of assets, environmental variability, and infeasibility of the empirically risk-minimizing generator).

**Outlook on testing with real data:** There exists a number of helpful approaches to estimating the performance of unlabeled data. A DNN of strong performance (or ensembles of those) can be utilized to compute pseudo ground truth. The model to be equipped with an AI perception system can be learned in a teacher-student fashion [AHKS19, BHSF19], and the discrepancies between the student model and the teachers can be compared with errors on a moderate subset with ground truth, for instance, in terms of correlations of errors and discrepancies. Furthermore, in order to process the vast amounts of recorded data and perform testing more efficiently, well-performing uncertainty quantification methods [KG17, KRPM+18, RCH+20] in combination with corner case detection methods [BBLFs19, HBR+21] can help to pre-select data for testing. Besides that, many additional approaches toward improving the reliability of DNN exist. However, while a big number of tools already exist, their proper application to DNN testing and inference of statistically relevant statements on the system’s safety still require thorough research.

These approaches and other upcoming research might be part of the solution to the testing problem in future.

**Concluding remark:** As a concluding remark, this chapter does not intend to discourage safety arguments, as also conceptualized in this volume. We do not deny the value of empirical evidence in order to, e.g., prove the relative superiority of one AI system over another with regards to safety. The inherent difficulty to provide direct evidence for the better-than-human safety of automated driving as required by the ethics committee of the German Ministry of Transportation and Digital Infrastructure [FBB+17] should not be mistaken as an excuse for a purely experimental approach. Bringing automated vehicles to the street without prior risk assessment implies that risks would be judged a posteriori based on the experience with a large fleet of automated vehicles.

Such matters attain urgency in the light of recent German legislation on the experimental usage of automated driving under human supervision<sup>5</sup> which only refers to the technical equipment for the sensing of automated vehicles, but does not specify the minimal performance for the AI-based perception based on the sensor information. Related regulations in other countries face similar problems.<sup>6</sup>

The debate on how to ensure a safe transition to automated driving that complies with high ethical standards, therefore, remains of imminent scientific and public interest.

**Acknowledgements** The authors thank Lina Haidar for support and numerical results in Section 4.2. Financial support by the German Federal Ministry of Economic Affairs and Energy (BMWi) via Grant No. 19A19005R as a part of the Safe AI for Automated Driving consortium is gratefully acknowledged.

## References

- [AGLR21] H. Asatryan, H. Gottschalk, M. Lippert, M. Rottmann, A convenient infinite dimensional framework for generative adversarial learning, pp. 1–29 (2021). [arXiv:2011.12087](https://arxiv.org/abs/2011.12087)
- [AHKS19] S. Abbasi, M. Hajabdollahi, N. Karimi, S. Samavi, Modeling teacher-student techniques in deep neural networks for knowledge distillation, pp. 1–6 (2019). [arXiv:1912.13179](https://arxiv.org/abs/1912.13179)
- [BBLFs19] J.-A. Bolte, A. Bär, D. Lipinski, T. Fingscheidt, Towards corner case detection for autonomous driving, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pp. 438–445, Paris, France (2019)
- [BCST20] G. Biau, B. Cadre, M. Sangnier, U. Tanielian, Some theoretical properties of GANs. *Ann. Stat.* **48**(3), 1539–1566 (2020)
- [BHSFs19] A. Bär, F. Hüger, P. Schlicht, T. Fingscheidt, On the robustness of redundant teacher-student frameworks for semantic segmentation, in *Proceedings of the IEEE/CVF*

---

<sup>5</sup> <https://dserver.bundestag.de/btd/19/274/1927439.pdf>.

<sup>6</sup> Framework for Automated Driving System Safety, No. NHTSA-2020-0106, 49 CFR Part 571 (Nov. 19, 2020).

- Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1380–1388, Long Beach, CA, USA (2019)
- [BTLFs20] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, T. Fingscheidt, Systematization of corner cases for visual perception in automated driving, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pp. 1257–1264, Virtual Conference (2020)
- [BTLFs21] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, T. Fingscheidt, Corner cases for visual perception in automated driving: some guidance on detection approaches, pp. 1–8 (2021). [arXiv:2102.05897](https://arxiv.org/abs/2102.05897)
- [C+15] F. Chollet, et al. Keras (2015). Accessed 18 Nov 2021
- [CATvS17] G. Cohen, S. Afshar, J. Tapson, A. van Schaik, EMNIST: an extension of MNIST to handwritten letters, pp. 1–10 (2017). [arXiv:1702.05373](https://arxiv.org/abs/1702.05373)
- [DRC+17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: an open urban driving simulator, in *Proceedings of the Conference on Robot Learning CORL*, pp. 1–16, Mountain View, CA, USA (2017)
- [FBB+17] U. di Fabio, M. Broy, R.J. Brügger, U. Eichhorn, A. Grunwald, D. Heckmann, E. Hilgendorf, H. Kagermann, A. Losinger, M. Lutz-Bachmann, A. Markl, K. Müller, K. Nehm, Ethik-Kommission - Automatisiertes und vernetztes Fahren. Technical report, Federal Ministry of Transport and Digital Infrastructure (2017)
- [Ger20] German Federal Ministry for Transportation & Infrastructure. Verkehr in Zahlen 2020/2021. Deutsches Zentrum für Luft- und Raumfahrt (2020)
- [HBR+21] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt, B. Sick, An application-driven conceptualization of corner cases for perception in highly automated driving, pp. 1–8 (2021). [arXiv:2103.03678](https://arxiv.org/abs/2103.03678)
- [HS16] J. Hedderich, L. Sachs, *Angewandte Statistik (in German)* (Springer, Berlin, 2016)
- [KB15] D.P. Kingma, J. Ba, ADAM: a method for stochastic optimization, in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–15, San Diego, CA, USA (2015)
- [KG17] A. Kendall, Y. Gal, What uncertainties do we need in Bayesian deep learning for computer vision? in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, pp. 5574–5584, Long Beach, CA, USA (2017)
- [KP16] N. Kalra, S.M. Paddock, Driving to safety: how many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A: Policy Pract.* **94**, 182–193 (2016)
- [Kri09] A. Krizhevsky, Object classification experiments. Technical Report, Canadian Institute for Advanced Research (2009)
- [KRPM+18] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J.R. Ledsam, K. Maier-Hein, S.M. Ali Eslami, D. Jimenez Rezende, O. Ronneberger, A probabilistic U-Net for segmentation of ambiguous images, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, pp. 6965–6975, Montréal, QC, Canada (2018)
- [LCB98] Y. LeCun, C. Cortes, C.J.C. Burges, The MNIST database of handwritten digits (1998). Accessed 18 Nov 2021
- [LL88] G.R. Loftus, E.F. Loftus, *Essence of Statistics* (Brooks/Cole Publishing Co, Salt Lake City, 1988)
- [LWC+19] L. Liu, W. Wei, K.-H. Chow, M. Loper, E. Gursoy, S. Truex, Y. Wu, Deep neural network ensembles against deception: ensemble diversity, accuracy and robustness, in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 274–282, Monterey, CA, USA (2019)
- [ME14] W.Q. Meeker, L.A. Escobar, *Statistical Methods for Reliability Data* (Wiley, New York, 2014)
- [Nat20] National Transportation Safety Board NSTB. Collision Between a Sport Utility Vehicle Operating With Partial Driving Automation and a Crash Attenuator (2020). Accessed 18 Nov 2021

- [RBK+21] J. Rosenzweig, E. Brito, H.-U. Kobialka, M. Akila, N.M. Schmidt, P. Schlicht, J.D. Schneider, F. Hüger, M. Rottmann, S. Houben, T. Wirtz, Validation of simulation-based testing: bypassing domain shift with label-to-image synthesis, pp. 1–8 (2021). [arXiv:2106.05549](https://arxiv.org/abs/2106.05549)
- [RCH+20] M. Rottmann, P. Colling, T.-P. Hack, R. Chan, F. Hüger, P. Schlicht, H. Gottschalk, Prediction error meta classification in semantic segmentation: detection via aggregated dispersion measures of softmax probabilities, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, Virtual Conference (2020)
- [Sch19] F.A. Schmidt, *Crowdproduktion von Trainingsdaten: Zur Rolle von Online-Arbeit beim Trainieren autonomer Fahrzeuge*, vol. 417 of *Study (in German)* (Hans-Böckler-Stiftung, 2019)
- [WLX+20] Y. Wu, L. Liu, Z. Xie, J. Bae, K.-H. Chow, W. Wei, Promoting high diversity ensemble learning with ensemblebench, in *Proceedings of the IEEE International Conference on Cognitive Machine Intelligence (CogMI)*, pp. 208–217, Atlanta, GA, USA (2020)
- [Zac92] S. Zacks, *Introduction to Reliability Analysis: Probability Models and Statistical Methods* (Springer, Berlin, 1992)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



# Analysis and Comparison of Datasets by Leveraging Data Distributions in Latent Spaces



Hanno Stage, Lennart Ries, Jacob Langner, Stefan Otten, and Eric Sax

**Abstract** Automated driving is widely seen as one of the areas, where key innovations are driven by the application of deep learning. The development of safe and robust deep neural network (DNN) functions requires new validation methods. A core insufficiency of DNNs is the lack of generalization for out-of-distribution datasets. One path to overcome this insufficiency is through the analysis and comparison of the domains of training and test datasets. This is important because otherwise, deep learning cannot advance automated driving. Variational autoencoders (VAEs) are able to extract meaningful encodings from datasets in their latent space. This chapter examines various methods based on these encodings and presents a broad evaluation on different automotive datasets and potential domain shifts, such as weather changes or new locations. The used methods are based on the distance to the nearest neighbors between datasets and leverage several network architectures and metrics. Several experiments with different domain shifts on different datasets are conducted and compared with a reconstruction-based method. The results show that the presented methods can be a promising alternative to the reconstruction error for detecting automotive-relevant domain shifts between different datasets. It is also shown that VAE loss variants that focus on a disentangled latent space can improve the stability of the domain shift detection quality. Best results were achieved with nearest neighbor methods using VAE and JointVAE, a VAE variant with a discrete and a continuous

---

H. Stage (✉) · L. Ries · J. Langner · S. Otten · E. Sax  
FZI Research Center for Information Technology, Haid-und-Neu-Str. 10-14,  
76131 Karlsruhe, Germany  
e-mail: [stage@fzi.de](mailto:stage@fzi.de)

L. Ries  
e-mail: [ries@fzi.de](mailto:ries@fzi.de)

J. Langner  
e-mail: [langner@fzi.de](mailto:langner@fzi.de)

S. Otten  
e-mail: [otten@fzi.de](mailto:otten@fzi.de)

E. Sax  
e-mail: [sax@fzi.de](mailto:sax@fzi.de)

latent space, in combination with a metric based on the well-known z-score, and with the NVAE, a VAE variant with optimizations regarding reconstruction quality, in combination with the deterministic reconstruction error.

## 1 Introduction

Artificial intelligence (AI) and deep learning are key enablers in automated driving [RMM21]. One important task in the development of automated cars is the perception of the vehicle environment. While several sensors can be used to solve this task, the most promising approaches in industry and research rely at least partly on camera-based perception [RNFP19]. In the perception sub-tasks of object detection and semantic segmentation, deep learning-based approaches already outperform classical methods [GLGL18, ZXW19].

However, those deep learning-based approaches suffer from several insufficiencies. Following Sämann et al., one central insufficiency is the ability to generalize from given training data [SSH20]. If the underlying distribution of the input data differs from the distribution of the training dataset, the prediction accuracy may decrease.

In the context of automated driving, a decrease in prediction accuracy can have dramatic consequences, e.g., fatal accidents due to traffic participants which were not detected, e.g., Uber accident [SSWS19]. One possibility to decrease the probability of these consequences is the optimization of used datasets. A high coverage over all possible situations in the training data is crucial to improve the quality of trained neural networks. However, since the input spaces (e.g., image data) are very high-dimensional and complex, the rule-based assessment of coverage is not feasible [BLO+17].

Hence, in this work, the capabilities of variational autoencoders (VAEs) of extracting meaningful distributions from datasets are used to compare different datasets in an unsupervised fashion. Since the VAE works directly with raw data, no assumptions over relevant dataset dimensions for the comparison are necessary. Additionally, no already trained perception function is necessary. The underlying data distributions found by the VAE are analyzed to find similarities and novelties, using the latent space of the VAE. This makes it possible to estimate whether a particular dataset is in-domain or out-of-domain of a second dataset, thus allowing the detection of *domain shifts* between datasets.

During early function development, the presented methods can be used to extract datasets with manageable sizes, still maintaining high coverages over relevant information in the complete data pool. These smaller datasets lead to faster training and validation and allow faster iterations. Moreover, the method can detect an unknown domain shift between the training, validation, and test datasets, which could lead to wrong conclusions about the performance of the function. In later development stages, the dataset comparison enables a prediction, if an already trained function

can be safely used in a different environment. Additionally, for domain shift detections with very high confidences, an online parallel execution of a perception neural network is thinkable, acting as a warning system for possibly unknown input data.

A domain shift can be caused by a huge variety of factors, some of them potentially unknown to the developer. This makes the application of rule-based approaches impossible. So, the idea of the presented dataset comparison method is to learn a multivariate distribution for all relevant aspects of one dataset and find useful metrics to compare these distributions. The used metrics in this investigation are related to known novelty detection techniques for VAEs.

The main contribution of this chapter is the application of various novelty detection methods and various VAE architectures in several experiments, each testing a different kind of domain shift. The applied novelty detection methods comprise five different nearest neighbor approaches applied to the latent spaces. As a benchmark, the reconstruction error is also computed [VGM+20]. The applicability of these methods is investigated on three datasets, each providing one or more inherent domain shifts (e.g., weather, location, or daytime). The presented broad evaluation of different approaches can give insights into the practical application of such approaches in the development of safe perception functions for automated driving.

## 2 Related Works

In this section, we provide an overview of related topics. There are various ways to useVAEs for anomaly detection in datasets [AC15, SWXS18, VGM+20]. An et al. compute the reconstruction probability of a given test sample by inspecting several samples drawn from the latent distribution of test samples [AC15]. The underlying assumption is that anomalous data samples have a higher variance and will therefore show a lower reconstruction quality. Vasilev et al. propose several methods and metrics to compute novelty scores using VAEs [VGM+20]. They describe 19 different methods for novelty calculation, including latent space-based and reconstruction-based approaches. Their evaluation on a magnetic resonance imaging dataset and on MNIST showed that a variety of metrics outperforms classical approaches such as nearest neighbor distance or principal component analysis (PCA) in the feature space.

In the domain of skin disease detection, VAE-based novelty detection methods were applied by Lu et al. [LX18]. They decompose the VAE loss into the reconstruction term and the KL term and use these as a novelty metric. Their results show better performance of the reconstruction-based approaches.

In the automotive domain, the issue of designing meaningful test datasets is addressed on various levels. Bach et al. address the selection of well-adjusted datasets for testing with a visual method for time series in recorded real-world data [BLO+17]. Langner et al. enhanced this approach with a method for dataset design, using the novelty detection capabilities of classical autoencoders [LBR+18]. A related approach

uses methods from natural language processing (NLP) to describe sequential contexts in automotive datasets, allowing similarity analysis and novelty detection [RSBS20].

These approaches can only indirectly find domain shifts for perception functions, since they are not designed to work in the feature space itself. For the detection of domain shifts directly in raw data, Löhdefink et al. propose the analysis of the reconstruction error of an autoencoder to predict the quality of a semantic segmentation network [LFK+20]. Sundar et al. address the issue of the diversity of automotive data by training various  $\beta$ -VAEs using different labels such as weather or daytime [SRR+20].

There is a lot of research in relation to application domains, proposed metrics and network architectures in the area of dataset analysis with VAEs. However, most of the related approaches restrict their analysis on datasets with only a few degrees of freedom (e.g., [VGM+20] and [LX18]) or test their method on only one type of domain shift (e.g., different locations). In this work, a broad spectrum of approaches (different network architectures, different metrics) were evaluated on three datasets including five different domain shifts.

### 3 Building Blocks of Our Approach

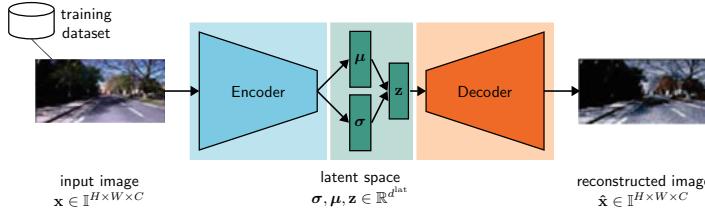
In this work, four different VAE architectures and five different metrics for anomaly detection are examined on their applicability to compare automotive datasets. Four methods use a nearest neighbor approach in the latent space for detection of domain shifts.

In this section, the VAEDNN is introduced. Afterwards, the used novelty detection methods are presented, and the section concludes with the explanation for dataset comparison and domain shift detection.

#### 3.1 Variational Autoencoder DNN

The difference between the original autoencoder and the VAE is that the VAE uses a stochastic model. The VAE consists of a stochastic encoder, a latent space, and a stochastic decoder. The general structure of a VAE can be seen in Fig. 1. Simplified, one could say that the VAE transfers the input data into a low-dimensional space to then reconstruct the input data from this space. The underlying stochastic model is now explained.

The probabilistic generative model of the VAE  $p_\theta(\mathbf{x}|\mathbf{z})$  is a deep neural network (DNN) with network parameters  $\theta$  mapping  $\mathbf{z}$  into  $\mathbf{x}$  and is named decoder [VGM+20, Dup18]. This decoder model learns a joint distribution of the data  $\mathbf{x}$  and a continuous latent variable  $\mathbf{z} \in \mathbb{R}^{d_{\text{lat}}}$  [CHK21]. In other words, the decoder describes the distribution of the decoded variable  $\hat{\mathbf{x}}$  given the encoded one  $\mathbf{z}$ . The inference model of the



**Fig. 1** Basic variational autoencoder structure, consisting of an encoder to transform data into the latent space and a decoder to reconstruct images from sampled values  $\mathbf{z}$ . The encoder receives an image  $\mathbf{x}$  as input, where  $H$  is the height of the image,  $W$  is the width of the image, and  $C$  is the number of color channels

VAE is trained to approximate the posterior distribution  $q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$  with parameters  $\phi$  mapping  $\mathbf{x}$  into  $\mathbf{z}$  and is named encoder [VGM+20]. The encoder describes the distribution of the encoded variable given the decoded one.

The probabilistic generated model and the inference model then give rise to the first loss term, since the aim for the input data of the encoder is to be reconstructed by the decoder. The reconstruction error loss is defined as follows:  $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]$  [Dup18]. In order to improve the disentanglement properties of  $q_\phi(\mathbf{z}|\mathbf{x})$ , the VAE has a loss optimization that tries to fit  $q_\phi(\mathbf{z}|\mathbf{x})$  to  $p(\mathbf{z})$  [HMP+17]. This loss optimization can be both controlling the capacity of the latent information bottleneck and embodying statistical independence. This is achieved by defining  $p(\mathbf{z})$  as an isotropic unit Gaussian distribution  $p(\mathbf{z}) = \mathcal{N}(0, I)$  [HMP+17]. In order to train the disentanglement in the bottleneck, the VAE uses the Kullback–Leibler (KL) divergence of the approximate from the true posterior  $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$ . Before we define the loss term, we introduce the variable  $\beta$ , which serves as a weighting factor. This allows us to define the loss function for the VAE and  $\beta$ -VAE as follows:

$$J(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})). \quad (1)$$

The first term of the loss function is the reconstruction error, which is responsible for the VAE encoding the informative latent variable  $\mathbf{z}$  and allowing the input data  $\mathbf{x}$  to be reconstructed [CHK21]. The second term regulates the posterior distribution by adjusting the distribution of the encoded latent variables  $q_\phi(\mathbf{z}|\mathbf{x})$  to the prior  $p(\mathbf{z})$  [CHK21]. If  $\beta > 1$  is selected, the loss function of the  $\beta$ -VAE as shown in (1) is used, and if  $\beta = 1$  is selected, (1) breaks down to the standard VAE loss [HMP+17]. The variable  $\beta$  scales the regularization term for the posterior distribution, with  $\beta > 1$  enforcing the  $\beta$ -VAE to encode more disentangled latent variables by matching the encoded latent variables with the prior by higher pressure on  $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$  [HMP+17, CHK21].

The VAE and  $\beta$ -VAE only use continuous latent variables to model the latent variable  $\mathbf{z}$  [CHK21]. The next VAE architecture that we use for our experiments additionally uses discrete variables to disentangle the generative factors of the observed data [CHK21]. The JointVAE uses the discrete variable  $\mathbf{c}$  to decode the generative

factors of the observed data [Dup18, CHK21]. The loss function of the JointVAE is an extension of the loss function of the  $\beta$ -VAE and results in (2).

$$\begin{aligned} J(\theta, \phi) = & \mathbb{E}_{q_\phi(z, c|x)} [\log p_\theta(x|z, c)] - \gamma |D_{KL}(q_\phi(z|x)||p(z)) - C_z| \\ & - \gamma |D_{KL}(q_\phi(c|x)||p(c)) - C_c|. \end{aligned} \quad (2)$$

The term  $D_{KL}(q_\phi(c|x)||p(c))$  is the Kullback–Leibler divergence for the discrete latent variables and  $D_{KL}(q_\phi(z|x)||p(z))$  is the Kullback–Leibler divergence for the continuous latent variables [Dup18]. The variables  $\gamma$ ,  $C_z$ , and  $C_c$  are hyperparameters [Dup18]. The hyperparameters  $C_z$  and  $C_c$  are gradually increased during training and both hyperparameters control the amount of information the model can encode [Dup18]. The hyperparameter  $\gamma$  forces the KL divergence to match the capacity  $C_z$  and  $C_c$ . As with the standard VAE,  $q_\phi(z, c|x)$  is the inference model which maps  $x$  into  $z$  and  $c$ .

In contrast to  $\beta$ -VAE and JointVAE which are focused on optimized latent space distributions, the goal of the Nouveau VAE (NVAE) is to reconstruct the best possible images [VK21]. Compared to the other used VAEs, the NVAE is a hierarchical VAE. This means that the NVAE has a hierarchical encoder and decoder structure and multiple latent spaces on different stages. Another special property is that this VAE has a bidirectional encoder [VK21]. This means that the information does not only pass from one encoder to the other but also the information from a deeper encoder passed back to the higher encoder. A focus during the development of the NVAE was to design expressive neural networks for VAEs [VK21].

In this chapter, the VAE [KW14],  $\beta$ -VAE [HMP+17], JointVAE [Dup18], and Nouveau VAE (NVAE) [VK21] are considered for the five experiments.

### 3.2 Methods for Novelty Detection

Nearest neighbor approaches for novelty detection have the assumption that the distributions of new and unknown data are different from the distributions of normal data. So, the latent space distance between a test sample distribution  $p_t$  and the closest normal sample can be used as a novelty score. Since latent spaces consist of high-dimensional distributions, the selection of a feasible metric to compute meaningful distances is non-trivial. According to Sect. 3.1, the used normal distributions  $p_n$  are defined by their parameters  $\mu_n$  and  $\sigma_n$ . As the JointVAE has different latent spaces, for the calculation of the nearest neighbor only the continuous latent space was used.

In this chapter, five metrics in the latent space and one metric in the pixel space are evaluated. The five metrics in the latent space can be further divided into two groups. Group one includes the metrics that only use the means ( $\mu$ ) of the distribution, and metric group two uses both means ( $\mu$ ) and variance ( $\sigma$ ) for the calculation.

First, the metrics from group one are presented that only use means for the calculation. The cosine distance is a widely used metric to determine the similarity of

vectors [NB10, Ye11]. Since it is designed for vectors, it can not directly be applied to distributions. Therefore, only the means  $\mu$  of the normal distributions are used for calculating the cosine distance. The cosine distance can be computed by

$$m_{\cos}(p_n, p_t) = \frac{\mu_n \mu_t^T}{\|\mu_n\| \|\mu_t\|}, \quad (3)$$

where  $\mu_n = (\mu_n(\delta))$  and  $\mu_t = (\mu_t(\delta))$  describe the  $d$ -dimensional mean vectors of two distributions  $p_n$  and  $p_t$ , related to a normal and a test sample.

The Euclidean distance is the distance that can be measured with a ruler between two points and is defined as

$$m_{\text{euc}}(p_n, p_t) = \sqrt{\sum_{\delta=1}^d (\mu_n(\delta) - \mu_t(\delta))^2}. \quad (4)$$

Here and in the following definitions,  $d$  is the dimensionality of the latent space. The Manhattan distance is inspired by the quadratic layout of Manhattan, New York City, and measures the distance between two vectors as the sum of their components' absolute differences [BR14].

$$m_{\text{man}}(p_n, p_t) = \sum_{\delta=1}^d |\mu_n(\delta) - \mu_t(\delta)|. \quad (5)$$

The former presented metrics only used the means  $\mu$ . Next, two metrics are introduced that use both means  $\mu$  and standard deviations  $\sigma$  for the calculation of the nearest neighbor. The next metric was defined by the authors of this chapter, using the well-known z-score. From the test samples' distribution, a vectorial value  $\mathbf{z}_t = (z_t(\delta))$  is sampled. Then the distance is computed via

$$m_{\text{zsc}}(p_n, p_t) = \sum_{\delta=1}^d \left\| \frac{z_t(\delta) - \mu_n(\delta)}{\sigma_n(\delta)} \right\|^2, \quad (6)$$

where the difference between  $\mathbf{z}_t$  and the normal sample mean  $\mu_n$  being element-wise divided by its standard deviation  $\sigma_n$ . Since we are only interested in the absolute distance of the distributions, we compute the squared L2 norm. The Bhattacharyya distance

$$m_{\text{bha}}(p_n, p_t) = \sum_{\delta=1}^d \frac{1}{4} \cdot \log \left( \frac{1}{4} \left( \frac{\sigma_n^2(\delta)}{\sigma_t^2(\delta)} + \frac{\sigma_t^2(\delta)}{\sigma_n^2(\delta)} + 2 \right) \right) + \frac{1}{4} \left( \frac{(\mu_n(\delta) - \mu_t(\delta))^2}{\sigma_n^2(\delta) + \sigma_t^2(\delta)} \right), \quad (7)$$

as proposed by Vasilev et al. [VGM+20], is also tested. The Bhattacharyya distance measures the similarity of two distributions [CA79].

To consider also a metric which does not use the latent space, the reconstruction error in the pixel space is considered. Since the VAE is trained only with normal data, the reconstruction quality should decrease, when new data samples are processed. Specifically, we chose the deterministic reconstruction error

$$m_{\text{dre}}(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|^2, \quad (8)$$

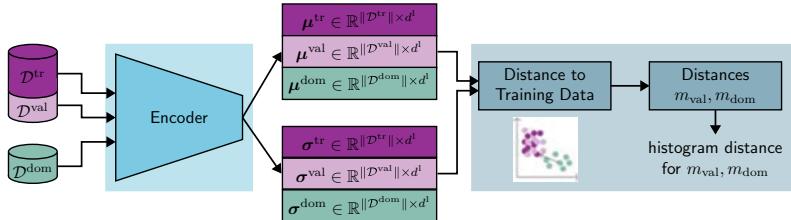
as proposed by Vasilev et al. [VGM+20], since it yielded the best results in their experiments.

### 3.3 Dataset Comparison Approach

In this section, we explain how the methods described in Sect. 3.2 can be leveraged to compare the whole datasets by explaining the data pipeline, also visible in Fig. 2. As starting points of the pipeline, a training dataset  $\mathcal{D}^{\text{tr}}$ , a validation dataset  $\mathcal{D}^{\text{val}}$ , which is in the distribution of the training dataset, and a test dataset  $\mathcal{D}^{\text{dom}}$  are needed.  $\mathcal{D}^{\text{dom}}$  is the dataset, for which the method tests, whether the dataset is in-domain or out-of-domain of the respective training dataset. The presented metrics should therefore be able to separate  $\mathcal{D}^{\text{val}}$  and  $\mathcal{D}^{\text{dom}}$ . Dataset  $\mathcal{D}^{\text{val}}$  most probably will have single images with higher novelty scores and  $\mathcal{D}^{\text{dom}}$  can have already known images. Therefore, the metrics most probably cannot perfectly separate  $\mathcal{D}^{\text{val}}$  and  $\mathcal{D}^{\text{dom}}$ . This is solved by an analysis of the novelty metric results on a dataset level.

The first step of the pipeline is the training of a VAE with the  $\mathcal{D}^{\text{tr}}$  dataset. For the novelty detection, the decoder can be discarded after training. As shown in Fig. 2, the trained encoder can now be used to generate the latent space representations of  $\mathcal{D}^{\text{tr}}$ ,  $\mathcal{D}^{\text{val}}$ , and  $\mathcal{D}^{\text{dom}}$ , spanned by the vectors  $\mu^{\text{tr}}, \sigma^{\text{tr}}, \mu^{\text{val}}, \sigma^{\text{val}}, \mu^{\text{dom}},$  and  $\sigma^{\text{dom}}$ .

Now, the novelty scores provided in Sect. 3.2 can be applied to every latent space representation of all data samples in  $\mathcal{D}^{\text{val}}$  and  $\mathcal{D}^{\text{dom}}$ . Hence, for the decision, whether  $\mathcal{D}^{\text{dom}}$  has a large gap to  $\mathcal{D}^{\text{val}}$ , the novelty scores have to be com-



**Fig. 2** Approach for dataset comparison based on latent space-based novelty scores. After training, the encoder is used to transform images into their latent space. In this space, different metrics can be applied to calculate nearest neighbor distances between different datasets. Domain shifts can then be detected by analyzing the histograms deviations

pared. In case of the nearest neighbor approaches, the resulting novelty scores  $m_{\text{val}} = \{(val_1, w_{\text{val}_1}), \dots, (val_n, w_{\text{val}_n})\}$  and  $m_{\text{dom}} = \{(dom_1, w_{\text{dom}_1}), \dots, (dom_n, w_{\text{dom}_n})\}$  are represented as a histogram where  $n \in \mathbb{N}$  is the number of buckets,  $\text{val}_a$  and  $\text{dom}_b$  are bucket representatives with  $0 < a, b \leq n$ , with  $a, b \in \mathbb{N}$ , and  $w_{\text{val}_a}, w_{\text{dom}_b}$  being the weights of the bucket [RTG00]. To compare the histograms we use the earth mover's distance (EMD) [RTG00]. The earth mover's distance can calculate the difference between the histogram  $m_{\text{val}}$  and  $m_{\text{dom}}$  by calculating the earth amount  $f_{ab}$  which is to be moved from bucket  $\text{val}_a$  to  $\text{dom}_b$  [RTG00]. To calculate the distance, the EMD also needs the ground distance matrix  $\mathbf{D} = (d_{ab})$ , where  $d_{ab}$  is the ground distance between bucket  $\text{val}_a$  and  $\text{dom}_b$  [RTG00]. With these terms, the EMD can be defined as follows:

$$EMD(m_{\text{val}}, m_{\text{dom}}) = \frac{\sum_{a=1}^n \sum_{b=1}^n d_{ab} f_{ab}}{\sum_{a=1}^n \sum_{b=1}^n f_{ab}}. \quad (9)$$

Here,  $\sum_{a=1}^n \sum_{b=1}^n d_{ab} f_{ab}$  is the flow between  $m_{\text{val}}$  and  $m_{\text{dom}}$ , that minimizes the overall cost with the following constraints [RTG00]:

$$f_{ab} \geq 0 \quad 1 \leq a \leq n, 1 \leq b \leq n, \quad (10)$$

$$\sum_{b=1}^n f_{ab} \leq w_{\text{val}_a} \quad 1 \leq a \leq n, \quad (11)$$

$$\sum_{a=1}^n f_{ab} \leq w_{\text{dom}_b} \quad 1 \leq b \leq n, \quad (12)$$

$$\sum_{a=1}^n \sum_{b=1}^n f_{ab} = \min \left( \sum_{a=1}^n w_{\text{val}_a}, \sum_{b=1}^n w_{\text{dom}_b} \right). \quad (13)$$

The used histograms are normalized on the x-axis to a range of values between 0 and 1. Additionally, the y-axis is also normalized, such that the histogram bars sum to 1. With this normalization, comparability between different experiments is achieved, since the results are independent of absolute metric values and dataset sizes.

For the reconstruction error, the histogram approach is comparable with the distances to the nearest neighbors. The histograms then are defined over the deterministic reconstruction error instead of the latent space distance.

## 4 Experimental Setup

In this section, first the selection of the datasets is explained. Afterwards, the experiments are explained and the hyperparameter values are given. The chapter is concluded with the training parameters for the experiments.

## 4.1 Datasets

For the evaluation of our method, we selected different datasets which all have different focuses. The first dataset is the Oxford RobotCar Dataset [MPLN17]. In this dataset, the same route was driven multiple times over the time period of 1 year, except for a few small deviations. The special feature of this dataset is that different weather conditions and seasons were recorded on the same route. This allows various domain shift analyses, such as changing weather conditions, e.g., sunny weather vs. snow. In the training, validation, snow test, rain test, and night test data splits there are 9130, 3873, 4094, 9757, and 9608 images, respectively. For all experiments, the test split is created to test whether the data is in-domain or out-of-domain.

As a second dataset, the Audi autonomous driving dataset (A2D2) [GKM+20] was used. This dataset is relatively new and was acquired in different German cities. For our evaluation, we compared the images from the city of Gaimersheim with the images from the city of Munich. The assumption with this dataset is that it shows relatively little variation because it was recorded in two German cities in the state of Bavaria. Thus, it should result in a very low dataset distance. The small difference is expected because Munich is larger than Gaimersheim, and therefore it is a comparison between an urban and a suburban city. The dataset is divided into training dataset of 12550 images, validation dataset of 3138 images, and test dataset of 5490 images.

As a third dataset, the nuScenes dataset [CBL+20] was selected. This dataset was selected because it was recorded in Boston and Singapore. The interesting point in the comparison is that there is right-hand traffic in Boston and left-hand traffic in Singapore. A number of 30013 images are used for training, 40152 images for validation, and 20512 images for test.

All datasets consist of a number of unconnected video sequences. To avoid similar images in different datasets splits, the split between  $\mathcal{D}^{\text{tr}}$ ,  $\mathcal{D}^{\text{val}}$ , and  $\mathcal{D}^{\text{dom}}$  was done sequence-wise. This ensures, that every scene only occurs in exactly one dataset. Only for the A2D2 dataset, the split between  $\mathcal{D}^{\text{tr}}$ ,  $\mathcal{D}^{\text{val}}$  was inside the Gaimersheim sequence, resulting in a few very similar images just before and after the split. However, this should not have a big effect, since the similar images take only a very small percentage of the whole dataset.

## 4.2 Experiments

For the evaluation of our method, we selected three different datasets, which all have a different focus. With these datasets, five different domain shifts were generated. These domain shifts can be seen in Table 1.

The five different domain shifts were investigated with four different VAE variants. Each variant was tested with 6 different metrics, leading to 120 experiments in total.

Experiments 1–3 were performed with the Oxford dataset. For training, images were taken on sunny days and as validation data, images were selected that were

**Table 1** All our datasets and experiments

Experiment	Dataset $\mathcal{D}^{\text{tr}}$	Dataset $\mathcal{D}^{\text{val}}$	Dataset $\mathcal{D}^{\text{dom}}$
Experiment 1	Oxford	Oxford alternate route	Snow
Experiment 2	Oxford	Oxford alternate route	Rain
Experiment 3	Oxford	Oxford alternate route	Night
Experiment 4	Gaimersheim	Gaimersheim (holdout)	Munich
Experiment 5	Boston	Boston (holdout)	Singapore

taken on an alternative route under the same weather conditions. This results in the following experiments for the Oxford dataset. Experiment 1 tests if the latent space can be used to distinguish sunny days from snowy days. The second experiment tests how rain can be distinguished from sun, and the third experiment investigates how sunny days differ from night scenes (Fig. 3).

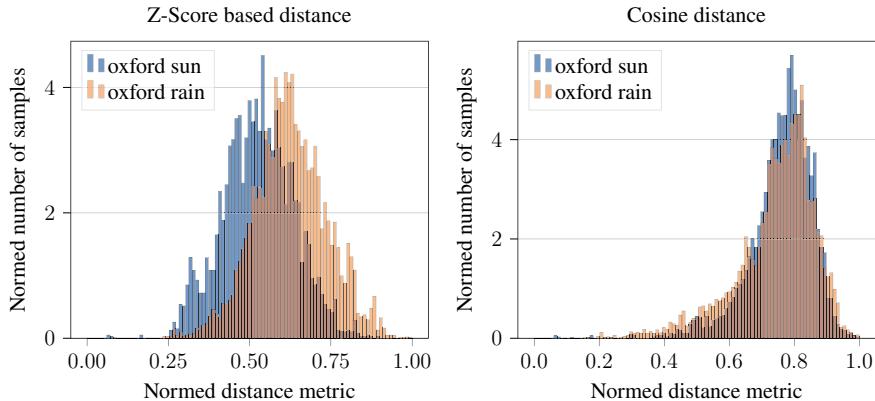
Experiment 4 uses the A2D2 dataset. This experiment only has a very small domain shift, since we compare data recorded in the sub-urban area of the small German city Gaimersheim compared to the urban area of Munich. Experiment 5 uses the nuScenes dataset. This experiment used images from Boston as training data and, as in Experiment 4, used a holdout of the data as validation data. Images from Singapore were used as a test set. Since in Singapore also night scenes were recorded, it has been ensured that no night scenes are in the test dataset.

To allow comparison of the experiments and metrics, baseline experiments were conducted in addition to the domain shift experiments. This is necessary to find out which value the metric has for in-domain data. For this purpose, the validation dataset  $\mathcal{D}^{\text{val}}$  was divided into  $\mathcal{D}^{\text{val}_0}$  and  $\mathcal{D}^{\text{val}_1}$ . This makes it possible to define a baseline for each experiment by setting  $\mathcal{D}^{\text{val}} = \mathcal{D}^{\text{val}_0}$  and  $\mathcal{D}^{\text{dom}} = \mathcal{D}^{\text{val}_1}$ .

### 4.3 VAE Hyperparameters

To ensure comparability between the VAE,  $\beta$ -VAE, and JointVAE, we used the same layer architecture and hyperparameters. The goal of this chapter is to be able to give an initial indication of whether a domain shift can be detected with simply chosen hyperparameters. It is to be expected, that with optimization of hyperparameters on multiple datasets, better results can be achieved.

To evaluate the performance of the presented methods, we have used the standard VAE,  $\beta$ -VAE, JointVAE, and the NVAE. The architecture was the same for the VAE,  $\beta$ -VAE, and the JointVAE. We use five hidden layers with dimensions 32, 64, 128, 256, and 512. After the hidden layers, we add three residual blocks. The residual blocks follow a latent space with 100 neurons. Both  $\beta$  and  $\gamma$  were set to 10.5 and the input size of the images was specified to a format of  $512 \times 256$ . For the NVAE, we adopted the architecture from the paper [VK21].



**Fig. 3** These histograms show the results for Experiment 2 and the JointVAE with the z-score and cosine distance. Training was done with the sunny Oxford images (oxford sun), and it was tested if rainy images (oxford rain) are out-of-domain. You can clearly see that the histogram on the left shows a difference between rain and sun, and the histograms on the right are rather on top of each other and poorly separated

#### 4.4 Training

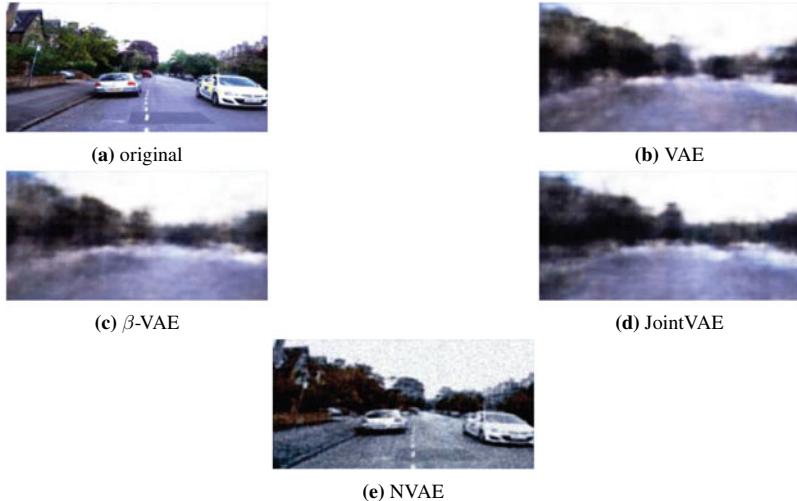
The VAE models used for the domain shift analysis were trained and implemented using Pytorch [PGC+17]. All VAE architectures were trained for 50 epochs with a batch size of 128 images. In addition, a learning rate of 0.0001 in combination with the Adam optimizer [KB15] was used. Figure 4 shows the original image and the reconstruction of the different VAEs. The selected image is from the validation dataset.

### 5 Experimental Results and Discussion

In this section, the results are presented. Section 5.1 first explains the results with the nearest neighbor method and after that in Sect. 5.2 the results will be discussed.

#### 5.1 Results

To make the results of the different architectures and the different metrics comparable, we calculated the ratio of the in-domain data with the out-of-domain data for each experiment and architecture. This is done by



**Fig. 4** Figure **a** shows an image from the Oxford validation dataset  $\mathcal{D}^{\text{val}}$ . Figures **b–e** show the reconstruction of the VAE architectures used in this chapter. It is clear that the NVAE has the best reconstruction quality. As expected, the reconstruction of the VAE (**b**),  $\beta$ -VAE (**c**), and JointVAE (TheVAEmodelsu) is worse compared to the NVAE. However, there are differences in reconstruction quality of (**b**)–(**d**), e.g., in the reconstruction of the JointVAE, compared to the VAE and  $\beta$ -VAE, you can imagine the car on the right edge. However, the fact that the reconstruction is worse is not a problem, because the focus of our method lies on the latent space

$$Q_{EMD} = \frac{EMD(\mathcal{D}^{\text{val}}, \mathcal{D}^{\text{dom}})}{EMD(\mathcal{D}^{\text{val}_0}, \mathcal{D}^{\text{val}_1})} \quad (14)$$

and leads to  $Q_{EMD} > 1$  for a detected domain shift. The intuition behind this is that the earth mover's distance (meaning the distribution of latent space distances) between two in-domain datasets has to be very low, whereas the latent space distances to out-of-domain data should be higher. The results for the Oxford data can be seen in Tables 2, 3, and 4. It can be seen that the NVAE achieves the best results in all experiments. The reconstruction error works with the NVAE in all three experiments, even if the z-score and the NVAE work slightly better in the rain images.

The values in the A2D2 experiment are lower compared to the Oxford values. This was expected, as the domain shift is significantly smaller compared to the Oxford domain shifts. Different architectures and metrics are close to each other in this experiment. The VAE with the z-score has the best result.

In the nuScenes dataset, the ratio is higher than in the A2D2 experiment. For this experiment, different combinations are again nearly the same, but the VAE with the Manhattan distance is the best.

An important aspect of the experiment results is the stability of results, meaning a combination of architecture and metric should be able to correctly detect every kind of domain shift. For most combinations, this is not fulfilled, as can be seen

**Table 2** The result of  $Q_{EMD}$  (14) from Oxford alternate route and  $\mathcal{D}^{\text{dom}}$  from Oxford snow

Architecture	Bhattacharyya distance	Cosine distance	Euclidean distance	Manhattan distance	Z-score	Reconstruction error
VAE	1.18	1.49	0.52	0.47	1.01	1.32
$\beta$ -VAE	7.29	4.12	1.03	0.85	1.95	1.10
JointVAE	3.77	0.50	0.37	0.45	3.00	1.25
NVAE	1.57	1.29	1.23	1.33	0.80	8.92

**Table 3** The result of  $Q_{EMD}$  (14) from Oxford alternate route and  $\mathcal{D}^{\text{dom}}$  from Oxford rain

Architecture	Bhattacharyya distance	Cosine distance	Euclidean distance	Manhattan distance	Z-score	Reconstruction error
VAE	1.45	0.77	1.00	1.06	1.01	1.22
$\beta$ -VAE	1.88	1.24	0.76	0.89	1.72	1.14
JointVAE	1.27	0.68	0.97	1.10	2.52	1.25
NVAE	2.14	0.86	6.08	3.80	5.40	5.00

**Table 4** The result of  $Q_{EMD}$  (14) from Oxford alternate route and  $\mathcal{D}^{\text{dom}}$  from Oxford night

Architecture	Bhattacharyya distance	Cosine distance	Euclidean distance	Manhattan distance	Z-score	Reconstruction error
VAE	1.60	1.86	1.55	1.68	1.45	5.88
$\beta$ -VAE	1.00	2.06	0.69	0.81	1.76	6.74
JointVAE	2.27	1.91	1.54	1.58	2.72	5.83
NVAE	0.00	0.43	0.85	0.60	0.60	10.42

**Table 5** The result of  $Q_{EMD}$  (14) from Gaimersheim (holdout) and  $\mathcal{D}^{\text{dom}}$  from Munich

Architecture	Bhattacharyya distance	Cosine distance	Euclidean distance	Manhattan distance	Z-score	Reconstruction error
VAE	0.61	0.41	0.72	0.80	1.76	0.65
$\beta$ -VAE	0.54	0.28	0.40	0.42	0.68	0.85
JointVAE	0.48	0.32	0.55	0.74	1.31	0.61
NVAE	1.29	1.70	1.31	1.69	1.25	1.57

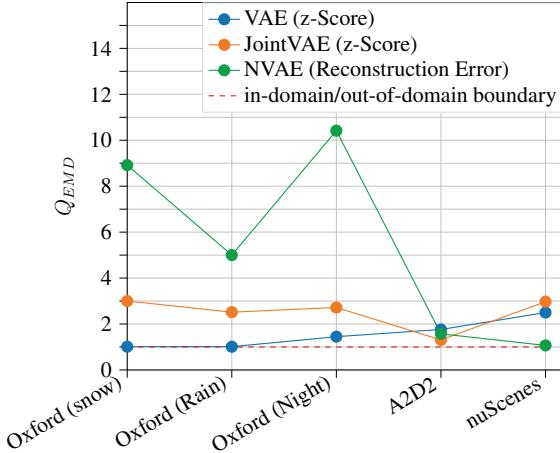
in Tables 2, 3, 4, 5, and 6. One example of this is the VAE with the Bhattacharyya distance: A combination with values over one in four of the five experiments is given (Tables 2, 3, 4, 6), while failing to detect a domain shift for the A2D2 experiment in Table 5.

**Table 6** The result of  $Q_{EMD}$  (14) from Boston (holdout) and  $\mathcal{D}^{\text{dom}}$  from Singapore

Architecture	Bhattacharyya distance	Cosine distance	Euclidean distance	Manhattan distance	Z-score	Reconstruction error
VAE	3.16	2.49	4.39	4.78	2.50	1.86
$\beta$ -VAE	1.34	2.42	4.38	4.03	1.87	1.81
JointVAE	1.05	2.19	4.69	4.52	2.97	1.79
NVAE	0.76	2.69	1.43	4.43	1.07	1.32

## 5.2 Discussion

Since this chapter aims to provide insights into the applicability of latent space methods for dataset comparison, some central findings are presented here. One goal was to use various experiments to find a combination that finds all domain shifts, if possible, irrespective of the extent of the domain shift. To achieve the goal, different latent space methods were used, as there are critics who say that a nearest neighbor search does not work in high-dimensional space [AHK01]. In the work of Goos et al. [AHK01], it was said that the Euclidean distance does not work in high-dimensional spaces as well, and that the Manhattan distance achieves better results. For the NVAE, the latent space dimension of 4096 is significantly higher than for the other VAE architectures with 100 dimensions. The NVAE has this high number of latent space dimensions because we used the parameters from the NVAE paper [VK21]. The parameters were adopted to achieve the best possible reconstruction without hyperparameter optimization runs. Our results show a similar pattern. The Manhattan distance can work better than the Euclidean distance. However, both metrics have the problem that they do not always give reliable results and have multiple values less than one in the ratio calculation. In addition to investigating whether the nearest neighbor method works, we also wanted to find an architecture and metric that works stable, meaning that they are able to correctly detect all tested domain shifts. Three architectures and two metrics were found that work and are shown graphically in Fig. 5. The three architectures found have in common that they do not have a ratio with a value smaller than one in all experiments. It can be seen that the NVAE with the reconstruction error works best with strong domain shifts, resulting in large visual differences between images such as rain, snow, and night. If the domain shifts consist of different locations, resulting in a variety of only small dataset differences, the z-score with the VAE or JointVAE works better compared to the NVAE with the reconstruction error. Due to the additional categorical latent space of the JointVAE, the discrete latent space is built up better. This leads to the JointVAE functioning more stable with the z-score than the VAE. The VAE is significantly worse with the Oxford dataset and nuScenes. However, in rain and snow, the VAE works not so good, as can be seen in Fig. 5. The ratio 1.01 is so close to the in-domain/out-of-domain boundary that one can say that the VAE should rather not be used for weather domain shifts. The fact that the JointVAE works better than the



**Fig. 5** In this plot, the stability of the three combinations of VAE and metrics can be seen. These three combinations have a value greater than one for each of the five experiments and therefore give stable correct results. The red dashed line indicates the in-domain/out-of-domain boundary

**Table 7** The training time of the different VAE architectures with the nuScenes dataset (30013 images). Training was done on an Nvidia A100 with 50 epochs. Additionally, we also report the time needed to transform an image into the latent space

Architecture	Training Time (hh:mm:ss)	Image to Latent Space (in seconds)
VAE	00:49:17	0.0022
$\beta$ -VAE	00:50:16	0.0019
JointVAE	00:49:36	0.0022
NVAE	03:31:45	0.0101

VAE may not only be due to the latent space but also to the loss, which has more pressure on the latent spaces and is thus better structured for a nearest neighbor search. In Fig. 5, the  $\beta$ -VAE is missing. This is due to the A2D2 dataset, where the domain shift was not detected with each of the metrics. This could maybe be improved with a better choice of  $\beta$ . The optimal parametrization of each network structure was beyond the scope of this chapter. As the results show, at least in some experiments, the NVAE outperforms the other approaches. However, due to the more complex architecture, the duration of one training is four times higher than for the other architectures (see Table 7). Also, the inference time, here meaning the time to execute the encoder part, is significantly longer with factors between 4 and 5. This aspect should definitely be considered, since a practical application of such approaches probably would implicate frequent retraining and encoding of large amounts of images.

In summary, the JointVAE with the z-score works as a fast alternative for the NVAE with the reconstruction error.

## 6 Conclusions

This chapter investigated whether the latent space of a VAE can be used to find domain shifts in the (image) context of the automotive industry. One goal was to find a combination of metrics and VAE architectures that work in general, i.e., for different datasets and domain shifts.

The novelty of our chapter is the different types of VAE used to evaluate the shift of image data in different datasets and under different conditions.

To find out which combination is the most stable and best compared to the others, we conducted 120 experiments. In these experiments, four different VAE architectures were trained with three different datasets and five different domain shifts were evaluated. The VAE architectures differ in that three of them have the same layer structure and only the loss changes, and one architecture focuses on reconstructing the images.

Six different metrics were used to detect the domain shifts, including the reconstruction error and four different nearest neighbor searches.

The most stable results were obtained with the JointVAE and the z-score, and with the NVAE and the reconstruction error. With “stable results” we mean that some ratio of earth movers distances between the histograms of the nearest neighbor distances for in-domain and domain shift data is greater than one for all experiments. The VAE with the z-score was also above one in all experiments, but in the Oxford dataset, it was so minimally above one that this combination of metric and VAE cannot be recommended without limitations. In the experiments, the combination JointVAE and z-score performed better for more subtle domain shifts like the change of location, whereas the combination NVAE with reconstruction error performed better for visually present domain shifts like snow or rain.

The results of the experiments show that JointVAE with the latent spatial metric can be an alternative to NVAE with the reconstruction error.

In the current state, it is possible to support the function development with the selection of the best out-of-domain data. This selection can help to select useful data subsets with all relevant data present.

Better results can be expected if the function for which domain shifts are being searched is included in the training. If the information from the function, e.g., inner representations of a neural network is used to build the latent space, a performance drop of this neural network can probably be predicted better. However, this approach would be no longer function-agnostic, since it requires an initially trained function.

As a conclusion, the presented methods can be a useful tool to compare and optimize datasets with only few assumptions and can therefore be one component in a strategy to develop functions with better generalization capabilities. Yet, for a complete approach, it has to be complemented with other methods such as rule-based or function-specific dataset optimizations to mutually compensate each other’s weaknesses.

**Acknowledgements** The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project “Methoden und Maßnahmen zur Absicherung von KI-basierten Wahrnehmungsfunktionen für das automatisierte Fahren (KI Absicherung).” The authors would like to thank the consortium for the successful cooperation.

## References

- [AC15] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability. Technical report, SNU Data Mining Center (2015)
- [AHK01] C. Aggarwal, A. Hinneburg, D. Keim, On the surprising behavior of distance metrics in high dimensional spaces, in *Proceedings of the International Conference on Database Theory*, pp. 420–434, London, UK (2001)
- [BLO+17] J. Bach, J. Langner, S. Otten, E. Sax, M. Holzäpfel, Test scenario selection for system-level verification and validation of geolocation-dependent automotive control systems, in *Proceedings of the ICE/IEEE International Conference on Engineering, Technology and Innovation (ITMC)*, pp. 203–210, Madeira, Portugal (2017)
- [BR14] S. Banerjee, A. Roy, *Linear Algebra and Matrix Analysis for Statistics* (CRC Press, Boca Raton, 2014)
- [CA79] G.B. Coleman, H.C. Andrews, Image segmentation by clustering. Proc. IEEE **67**(5), 773–785 (1979)
- [CBL+20] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuScenes: a multimodal dataset for autonomous driving, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11621–11631, Virtual Conference (2020)
- [CHK21] J. Choi, G. Hwang, M. Kang, Discond-VAE: disentangling continuous factors from the discrete, pp. 1–14 (2021). [arXiv:2009.08039](https://arxiv.org/abs/2009.08039)
- [Dup18] E. Dupont, Learning disentangled joint continuous and discrete representations, pp. 1–16 (2018). [arXiv:1804.00104](https://arxiv.org/abs/1804.00104)
- [GKM+20] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A.S. Chung, L. Hauswald, V.H. Pham, M. Mühllegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, P. Schuberth, A2D2: Audi autonomous driving dataset, pp. 1–10 (2020). [arXiv:2004.06320](https://arxiv.org/abs/2004.06320)
- [GLGL18] Y.Y. Guo, T.G. Liu, M.S. Lew, A review of semantic segmentation using deep neural networks. Int. J. Multimed. Inf. Retr. **7**(2), 87–93 (2018)
- [HMP+17] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M.M. Botvinick, S. Mohamed, A. Lerchner,  $\beta$ -VAE: learning basic visual concepts with a constrained variational framework, in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–22, Toulon, France (2017)
- [KB15] D.P. Kingma, J. Ba, ADAM: a method for stochastic optimization, in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–15, San Diego, CA, USA (2015)
- [KW14] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–14, Banff, AB, Canada (2014)
- [LBR+18] J. Langner, J. Bach, L. Ries, S. Otten, M. Holzäpfel, E. Sax, Estimating the uniqueness of test scenarios derived from recorded real-world-driving-data using autoencoders, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pp. 1860–1866, Changshu, China (2018)

- [LFK+20] J. Löhdefink, J. Fehrling, M. Klingner, F. Hüger, P. Schlicht, N.M. Schmidt, T. Fingscheidt, Self-supervised domain mismatch estimation for autonomous perception, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1359–1368, Virtual Conference (2020)
- [LX18] Y. Lu, P. Xu, Anomaly detection for skin disease images using variational autoencoder, pp. 1–8 (2018). [arXiv:1807.01349](https://arxiv.org/abs/1807.01349)
- [MPLN17] W. Maddern, G. Pascoe, C. Linegar, P. Newman, 1 year, 1000 km: the Oxford RobotCar dataset. *Int. J. Robot. Res.* **36**(1), 3–15 (2017)
- [NB10] H.V. Nguyen, L. Bai, Cosine similarity metric learning for face verification, in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pp. 709–720, Queenstown, New Zealand (2010)
- [PGC+17] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS) Workshops*, pp. 1–4, Long Beach, CA, USA (2017)
- [RMM21] M. Rabe, S. Milz, P. Mäder, Development methodologies for safety critical machine learning applications in the automotive domain: a survey, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 129–141, Virtual Conference (2021)
- [RNFP19] F. Rosique, P.J. Navarro, C. Fernández, A. Padilla, A systematic review of perception system and simulators for autonomous vehicles research. *Sensors* **19**(3), 1–29 (2019)
- [RSBS20] L. Ries, M. Stumpf, J. Bach, E. Sax, Semantic comparison of driving sequences by adaptation of word embeddings, in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1–7, Virtual Conference (2020)
- [RTG00] Y. Rubner, C. Tomasi, L.J. Guibas, The Earth Mover’s distance as a metric for image retrieval. *Int. J. Comput. Vis. (IJCV)* **40**, 99–121 (2000)
- [SRR+20] V.K. Sundar, S. Ramakrishna, Z. Rahiminasab, A. Easwaran, A. Dubey, Out-of-distribution detection in multi-label datasets using latent space of  $\beta$ -VAE, in *Proceedings of the IEEE Symposium on Security and Privacy (SP) Workshops*, pp. 250–255, Virtual Conference (2020)
- [SSH20] T. Sämann, P. Schlicht, F. Hüger, Strategy to increase the safety of a DNN-based perception for HAD systems, pp. 1–14 (2020). [arXiv:2002.08935](https://arxiv.org/abs/2002.08935)
- [SSWS19] N.A. Stanton, P.M. Salmon, G.H. Walker, M. Stanton, Models and methods for collision analysis: a comparison study based on the Uber collision with a pedestrian. *Saf. Sci.* **120**(12), 117–128 (2019)
- [SWXS18] J. Sun, X. Wang, N. Xiong, J. Shao, Learning sparse representation with variational auto-encoder for anomaly detection. *IEEE Access* **6**, 33353–33361 (2018)
- [VGM+20] A. Vasilev, V. Golkov, M. Meissner, I. Lipp, E. Sgarlata, V. Tomassini, D.K. Jones, D. Cremers, Q-space novelty detection with variational autoencoders, in *Proceedings of the International Conference on Medical Image Computing & Computer-Assisted Intervention (MICCAI) Workshops*, pp. 113–124, Virtual Conference (2020)
- [VK21] A. Vahdat, J. Kautz, NVAE: a deep hierarchical variational autoencoder, pp. 1–21 (2021). [arXiv:2007.03898](https://arxiv.org/abs/2007.03898)
- [Ye11] J. Ye, Cosine similarity measures for intuitionistic fuzzy sets and their applications. *Math. Comput. Modelling* **53**(1–2), 91–97 (2011)
- [ZZXW19] Z.-Q. Zhao, P. Zheng, X. Shou-Tao, W. Xindong, Object detection with deep learning: a review. *IEEE Trans. Neural Netw. Learn. Syst. (TNNLS)* **30**(11), 3212–3232 (2019)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



# Optimized Data Synthesis for DNN Training and Validation by Sensor Artifact Simulation



Korbinian Hagn and Oliver Grau

**Abstract** Synthetic, i.e., computer-generated imagery (CGI) data is a key component for training and validating deep-learning-based perceptive functions due to its ability to simulate rare cases, avoidance of privacy issues, and generation of pixel-accurate ground truth data. Today, physical-based rendering (PBR) engines simulate already a wealth of realistic optical effects but are mainly focused on the human perception system. Whereas the perceptive functions require realistic images modeled with sensor artifacts as close as possible toward the sensor, the training data has been recorded. This chapter proposes a way to improve the data synthesis process by application of realistic sensor artifacts. To do this, one has to overcome the domain distance between real-world imagery and the synthetic imagery. Therefore, we propose a measure which captures the generalization distance of two distinct datasets which have been trained on the same model. With this measure the data synthesis pipeline can be improved to produce realistic sensor-simulated images which are closer to the real-world domain. The proposed measure is based on the Wasserstein distance (earth mover's distance, EMD) over the performance metric mean intersection-over-union (mIoU) on a per-image basis, comparing synthetic and real datasets using deep neural networks (DNNs) for semantic segmentation. This measure is subsequently used to match the characteristic of a real-world camera for the image synthesis pipeline which considers realistic sensor noise and lens artifacts. Comparing the measure with the well-established Fréchet inception distance (FID) on real and artificial datasets demonstrates the ability to interpret the generalization distance which is inherent asymmetric and more informative than just a simple distance measure. Furthermore, we use the metric as an optimization criterion to adapt a synthetic dataset to a real dataset, decreasing the EMD distance between a synthetic and the Cityscapes dataset from 32.67 to 27.48 and increasing the mIoU of our test algorithm (DeepLabV3+) from 40.36 to 47.63%.

---

K. Hagn (✉) · O. Grau

Intel Deutschland GmbH, Lilienthalstraße 15, 85579 Neubiberg, Germany  
e-mail: [korbinian.hagn@intel.com](mailto:korbinian.hagn@intel.com)

O. Grau

e-mail: [oliver.grau@intel.com](mailto:oliver.grau@intel.com)

## 1 Introduction

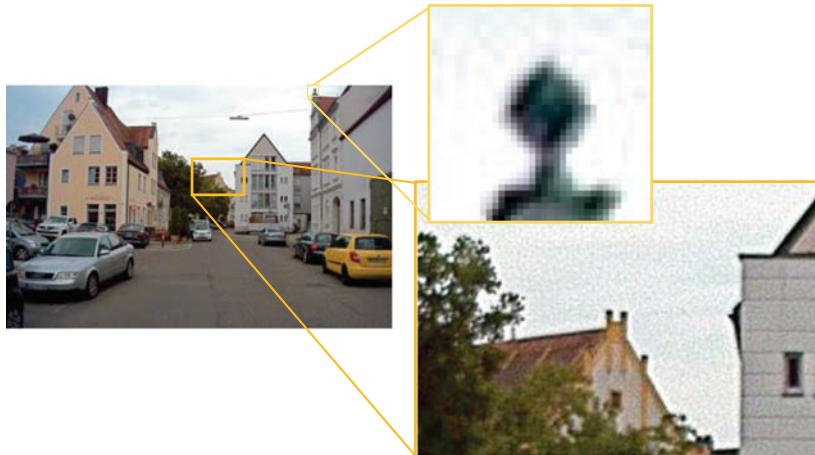
Validation of deep neural networks (DNNs) is increasingly resorting toward computer-generated imagery (CGI) due to its mitigation of certain issues. First, synthetic data can avoid privacy issues found with recordings of members of the public and, on the other hand, can automatically produce vast amounts of data at high quality with pixel-accurate ground truth data and reliability than costly manually labeled data. Moreover, simulations allow synthesis of rare cases and the systematic variation and explanation of critical constellations [SGH20]—a requirement for validation of products targeting safety-critical applications, such as automated driving. Here, the creation of corner cases and scenarios which otherwise could not be recorded in a real-world scenario without endangering other traffic participants is the key argument for the validation of perceptive AI with synthetic images.

Despite the advantages of CGI methods, training and validation with synthetic images still have challenges: Training with these images does not guarantee a similar performance on real-world images and validation is only valid if one can verify that the found weaknesses in the validation do not stem from the synthetic-to-real distribution shift seen in the input.

To measure and mitigate this domain shift, metrics have been introduced with various applications in the field of domain adaptation or transfer learning. In domain adaptation, the metrics such as FID, kernel inception distance (KID), and maximum mean discrepancy (MMD) are applied to train generative adversarial networks (GANs) to adapt on a target feature space [PTKY09] or to re-create the visual properties of a dataset [SGZ+16]. However, the problem of training and validation with synthetic imagery is directly related to the predictive performance of a perception algorithm on the target data, and these kinds of metrics struggle to correlate with the predictive performance [RV19]. Additionally, applications of domain adaptation methods often resort to specifically trained DNNs, e.g., GANs, which adapt one domain to the other and therefore add an extra layer of complexity and uncontrollability. This is especially unwanted if a validation goal is tested, e.g., to detect all pedestrians, and the domain adaption by a GAN would add additional objects into the scene (e.g., see [HTP+18]) making it even harder to attribute detected faults of the model to certain specifics of the tested scene. Here, the creation of images via a synthesis process allows to understand domain distance influence factors more directly as all parameters are under direct control.

Camera-recorded images inherently show visual imperfections or artifacts, such as sensor noise, blur, chromatic aberration, or image saturation, as can be seen in an image example from the A2D2 [GKM+20] dataset in Fig. 1. CGI methods, on the other hand, are usually based on idealized models; for example, the pinhole camera model [Stu14] which is free of sensor artifacts.

In this chapter, we present an approach to decrease the domain divergence of synthetic to real-world imagery for perceptive DNNs by realistically modeling sensor lens artifacts to increase the viability of CGI for training and validation. To achieve this, we first introduce a model of sensor artifacts whose parameters are extracted



**Fig. 1** Real-world images (here A2D2) exhibit sensor lens artifacts which have to be closely modeled by an image synthesis process to decrease the domain distance of synthetic to real-world datasets to make them viable for training and validation

from a real-world dataset and then apply it on a synthetic dataset for training and measuring the remaining domain divergence via validation. Therefore, a new interpretation of the domain divergence by generalization of the distance of two datasets by the per-image performance comparison over a dataset utilizing the Wasserstein or earth mover’s distance (EMD) is presented. Next, we demonstrate how this model is able to decrease the domain divergence further by optimization of the initial extracted sensor camera simulation parameters as depicted in Fig. 6. Additionally, we compare our results with randomly chosen parameters as well as with randomly chosen and optimized parameters. Last, we strengthen the case for the usability of our EMD domain divergence measure by comparison with the well-known Fréchet inception distance (FID) on a set of real-world and synthetic datasets and highlight the advantage of our asymmetric domain divergence against the symmetric distance.

## 2 Related Works

This chapter is related to two areas: domain distance measures, as used in the field of domain adaptation and synthetic data generation for training and validation.

**Domain distance measures:** A key challenge in domain adaptation approaches is the expression of a distance measure between datasets, also called domain shift. A number of methods were developed to mitigate this shift (e.g., see [LCWJ15, GL15, THSD17, THS+18]).

To measure the domain shift or domain distance, the inception score (IS) has been proposed [SGZ+16], where the classification output of an InceptionV3-

based [SVI+16] discriminator network trained on the ImageNet dataset [DDS+09] is used. The works of [HRU+17, BSAG21] rely on features extracted from the InceptionV3 network to tune domain adaptation approaches, i.e., the FID and KID. However, these metrics cannot predict if the classification performance increases when adapted data is applied as training data for a discriminator [RV19].

Therefore, to measure performance directly, it is essential to train with the adapted or synthetic data and validate on the target data, i.e., cross-evaluation as done by [RSM+16, WU18, SAS+18].

**Performance metrics:** The mean intersection-over-union (mIoU) is a widely used performance metric for benchmarking semantic segmentation [COR+16, VSN+18]. Adaptations and improvements of the mIoU have been proposed which put more weight on the segmentation contour as in [FMWR18, RTG+19]. Performance metrics as the mIoU are computed over the whole validation dataset, i.e., the whole confusion matrix, but there are propositions to apply the mIoU calculation on a per-image basis and compare the resulting empirical distributions [CLP13].

A per-image comparison mitigates several shortcomings of a single evaluation metric on the whole dataset when used for comparison of classifiers on the same dataset. First, one can distinguish multimodal and unimodal distributions, i.e., strong classification on one half and weak classification on the other half of a set can lead to the same mean as an average classification on all samples. Second, unimodal distributions with the same mean but different shape are also indiscernible under a single dataset averaged metric. This justification led to our choice of a per-image-based mIoU metric as it allows for deeper investigations which are especially helpful when one wants to understand the characteristics that increase or decrease a domain divergence.

**Sensor simulation for synthetic image training:** The use of synthesized data for development and validation is an accepted technique and has been also suggested for computer vision applications (e.g., [BB95]). Recently, specifically for the domain of driving scenarios, games engines have been adapted [RHK17, DRC+17].

Although game engines provide a good starting point to simulate environments, they usually only offer a closed rendering set-up with many trade-offs balancing between real-time constraints and a subjectively good visual appearance for human observers. Specifically the lighting computation in the rendering pipelines is transparent. Therefore, it does not produce a physically correct imagery; instead only a fixed rendering quality (as a function of lighting computation and tone mapping), resulting in output of images having a low dynamic range (LDR) (typically 8-bit per RGB color channel).

Recently, physical-based rendering techniques have been applied to the generation of data for training and validation, like Synscapes [WU18]. For our chapter we use a dataset in high dynamic range (HDR) created with the physical-based Blender Cycles renderer.<sup>1</sup> We implemented a customized tone mapping to 8-bit per color channel and sensor simulation, as described in the next section.

---

<sup>1</sup> Provided by the KI Absicherung project [KI 20].

While there is great interest in understanding the domain distance in the area of domain adaptation via generative strategies, i.e., GANs, there has been little research regarding sensor artifact influence on training and validation with synthetic images. Other works [[dCCN+16](#), [NdCCP18](#)] add different kinds of sensor noise to their training set and report a degradation of performance, compared to a model trained with no noise in the training set, due to training of a harder, i.e., noisier, visual task. Adding noise in training is a common technique for image augmentation and can be seen as a regularization technique [[Bis95](#)] to prevent overfitting.

Our task of modeling sensor artifacts for synthetic images extracted from camera images is not aimed at improving the generalization through random noise, but to tune the parameters of our sensor model to closely replicate the real-world images and improve generalization on the target data.

First results of modeling camera effects to improve synthetic data learning on the perceptive task of bounding box detection have been proposed by [[CSVJR18](#), [LLFW20](#)]. Lin et al. [[LLFW20](#)] additionally state that generalization is an asymmetric measure which should be considered when comparing with symmetric dataset distance measures from literature. Furthermore, Carlson et al. [[CSVJR19](#)] learned sensor artifact parameters from a real-world dataset and applied the learned parameters of their noise sources as image augmentation during training with synthetic data on the task of bounding box detection. However, contrasting our approach, they apply their optimization as style loss on a latent feature vector extracted from a VGG-16 network trained on ImageNet and evaluate the performance on the task of 2D object detection.

### 3 Methods

Given a synthetic (CGI) dataset of urban street scenes, our goal is to decrease the domain gap to a real-world dataset for semantic segmentation by realistic sensor artifact simulation. Therefore, we systematically analyze the image sensor artifacts of the real-world dataset and use this extracted parametrization for our sensor artifact simulation. To compare our synthetic dataset with the real-world dataset we contrive a novel per-image performance-based metric to measure the generalization distance between the datasets. We utilize a DeepLabV3+ [[CZP+18](#)] semantic segmentation model with a ResNet101 [[HZRS16](#)] backbone to train and evaluate on the different datasets throughout this paper. To show the valuable properties of our measure we compare it with the established domain distance, i.e., Fréchet inception distance (FID). Lastly, we use our measure as optimization criteria for adapting the parameters of our sensor artifact simulation with the extracted parameters as starting point and show that we can further decrease the domain distance from synthetic images to real-world images.

### 3.1 Sensor Simulation

We implemented a simple sensor model with the principle blocks depicted in Fig. 2: The module expects images in linear RGB space. Rendering engines like Blender Cycles<sup>2</sup> can provide these images as results in OpenEXR format.<sup>3</sup>

We simulate a simple model by applying *chromatic aberration*, *blur*, and *sensor noise*, as additive Gaussian noise (zero mean, variance is a free parameter), followed by a simple exposure control (linear tone mapping), finished by non-linear *gamma correction*.

First, we apply blur by a simple box filter with filter size  $F \times F$  and a chromatic aberration (CA). The CA is approximated using radial distortions ( $k_1$ , second order), e.g., [CV14], as defined in OpenCV. The CA is implemented as a per channel (red, green, blue) variation of the  $k_1$  radial distortion, i.e., we introduce an incremental parameter  $ca$  that affects the radial distortions:  $k_1(\text{blue}) = -ca$ ;  $k_1(\text{green}) = 0$ ;  $k_1(\text{red}) = +ca$ . As the next step, we apply Gaussian noise to the input image.

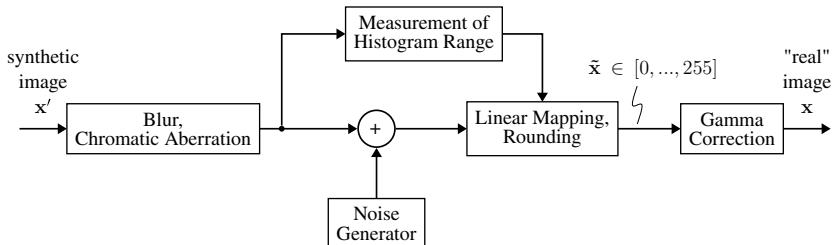
Applying a linear function, the pixel values are then mapped and rounded to the target output byte range  $[0, \dots, 255]$ .

The two parameters of the linear mapping are determined by a histogram evaluation of the input RGB values of the respective image, imitating an auto exposure of a real camera. In our experiments we have set it to saturate 2% (initially) of the brightest pixel values, as these are usually values of very high brightness, induced by sky or even the sun. Values below the minimum or above the set maximum are mapped to 0 or 255, respectively.

In the last step we apply gamma correction to achieve the final processed synthetic image:

$$\mathbf{x} = (\tilde{\mathbf{x}})^\gamma \quad (1)$$

The parameter  $\gamma$  is an approximation of the sensor non-linear mapping function. For media applications this is usually  $\gamma = 2.2$  for the sRGB color space [RD14]. However, for industrial cameras, this is not yet standardized and some vendors do



**Fig. 2** Sensor artifact simulation

<sup>2</sup> <https://www.blender.org/>.

<sup>3</sup> <https://www.openexr.com/>.



**Fig. 3** Left **a**: Synthetic images without lens artifacts. Right **b**: Applied sensor lens artifacts, including exposure control

not reveal it.<sup>4</sup> We therefore estimate the parameter as an approximation. Figure 3 depicts the difference of an image with and without simulated sensor artifacts.

### 3.2 Dataset Divergence Measure

Our proposed distance quantifies per image performance between models trained on different datasets but evaluated on the same dataset. Considering the task of semantic segmentation we chose the mIoU as our base metric. We then modify the mIoU to be calculated per image instead of the confusion matrix on the whole evaluated dataset. Next, we introduce the Wasserstein-1 or earth mover's distance (EMD) metric as our divergence measure between the per-image mIoU distribution of two classifiers trained on distinct datasets, i.e., synthetic and real-world datasets, but evaluated on the same real-world dataset the second classifier has been trained with.

The mIoU is defined as follows:

$$mIoU = \frac{1}{S} \sum_{s \in S} \frac{TP_s}{TP_s + FP_s + FN_s} \times 100\%, \quad (2)$$

---

<sup>4</sup> The providers of the Cityscapes dataset don't document the exact mapping.

with  $TP_s$ ,  $FN_s$ , and  $FP_s$  being the amount of true-positives, false-negatives, and false-positives of the  $s$ th class over all images of the evaluated dataset.

Here,  $\mathcal{S} = \{0, 1, \dots, S - 1\}$ , with  $S = 11$ , as we use the 11 classes defined in Table 1. These classes are the maximal overlap of common classes in the real and synthetic datasets considered for cross-evaluation and comparison of our measure with the Fréchet inception distance (FID), as can be seen later in Sect. 4.3, Tables 3 and 4.

A distribution over the per-image IoU takes the following form:

$$IoU_n = \frac{1}{S} \sum_{s \in \mathcal{S}} \frac{TP_{s,n}}{TP_{s,n} + FP_{s,n} + FN_{s,n}} \times 100\%, \quad (3)$$

where  $n$  denotes the  $n$ th image in the validation dataset. Here,  $IoU_n$  is measured in %. We want to compare the distributions of per-image IoU values from two different models; therefore, we apply the Wasserstein distance. The Wasserstein distance as an optimal mass transport metric from [KPT+17] is defined for density distributions  $p$  and  $q$  where  $\inf$  denotes the infimum, i.e., lowest transportation cost,  $\Gamma(p, q)$  denotes the set containing all joint distributions  $\pi$ , i.e., transportation maps, for  $(X, Y)$  which have the marginals  $p$  and  $q$  as follows:

$$W_r(p, q) = \left( \inf_{\pi \in \Gamma(p, q)} \int_{\mathbb{R} \times \mathbb{R}} |X - Y|^r d\pi \right)^{1/r}. \quad (4)$$

This distance formulation is equivalent to the following [RTC17]:

$$W_r(p, q) = \left( \int_{-\infty}^{\infty} |P(t) - Q(t)|^r dt \right)^{1/r}. \quad (5)$$

Here  $P$  and  $Q$  denote the respective cumulative distribution functions (CDFs) of  $p$  and  $q$ .

In our application we calculate the empirical distributions of  $p$  and  $q$ , which simplifies in this case to the function of the order statistics:

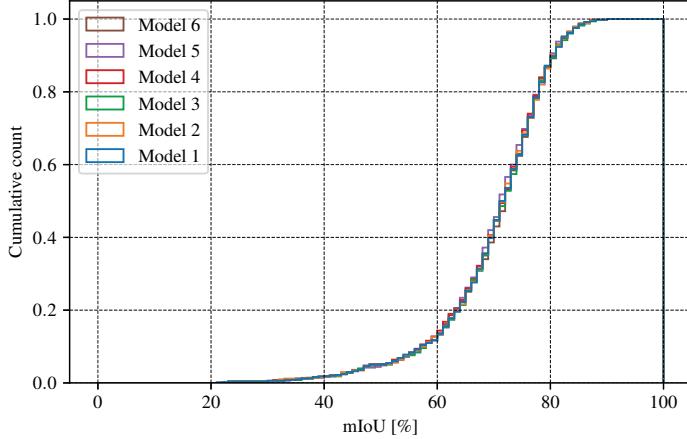
$$W_r(\hat{p}, \hat{q}) = \left( \sum_{i=1}^n |\hat{p}_i - \hat{q}_i|^r \right)^{1/r}, \quad (6)$$

where  $\hat{p}$  and  $\hat{q}$  are the empirical distributions of the marginals  $p$  and  $q$  sorted in ascending order. With  $r = 1$  and equal weight distributions we get the earth mover's distance (EMD) which, in other words, measures the area between the respective CDFs with  $L_1$  as ground distance.

We assume a sample size of at least 100 to be enough for the EMD calculation to be valid, as fewer samples might not guarantee a sufficient sampling of the domains. In our experiments we use sample sizes  $\geq 500$ .

**Table 1** Due to differences in label definition of real-world datasets, the class mapping for training and evaluation is decreased to 11 classes that are common in all considered datasets: A2D2 [GKM+20], Cityscapes [COR+16], Berkeley Deep Drive (BDD100K) [YCW+20], Mapillary Vistas (MV) [NOBK17], India Driving Dataset (IDD) [VSN+18], GTAV [RVRK16], our synthetic dataset [KI20], and Synscapes [WU18]

s	0	1	2	3	4	5	6	7	8	9	10
Label	road	sidewalk	building	pole	traffic light	traffic sign	vegetation	sky	human	car	truck



**Fig. 4** CDF of ensembles of DeeplabV3+ models trained on Cityscapes and evaluated on its validation set. Applying the 2-sample Kolmogorov-Smirnov test to each possible pair of the ensemble, we get a minimum  $p$ -value  $> 0.95$

The FID is a special case of the Wasserstein-2 distance derived from (6) with  $p = 2$  and  $\hat{\mu}$  and  $\hat{q}$  being normally distributed, leading to the following definition:

$$\text{FID} = \|\boldsymbol{\mu} - \boldsymbol{\mu}_w\|^2 + \text{tr}(\Sigma + \Sigma_w - 2(\Sigma \Sigma_w)^{1/2}), \quad (7)$$

where  $\boldsymbol{\mu}$  and  $\boldsymbol{\mu}_w$  are the means, and  $\Sigma$  and  $\Sigma_w$  are the covariance matrices of the multivariate Gaussian-distributed feature vectors of synthetic and real-world datasets, respectively.

Compared to distance metrics such as the FID which by definition is symmetric, our measure is a divergence, i.e., the distance from dataset A to dataset B can be different to the distance from dataset B to dataset A. Being a divergence also reflects the characteristic of a classifier having different generalization distance when trained on dataset A and evaluated on dataset B or the other way around.

Because the ground measure of the signatures, i.e., the IoU per image, is bounded to  $0 \leq IoU_n \leq 100$ , the EMD measure is then bounded to  $0 \leq EMD \leq 100^r$  with  $r$  being the Wasserstein norm. For  $r = 1$ , the measure is bound with  $0 \leq EMD \leq 100$ .

To verify whether the per-image IoU of a dataset is a good proxy of a dataset's domain distribution, we need to verify that the distribution stays (nearly) constant when training from different starting conditions. Therefore, we trained six models of the DeeplabV3+ network with the same hyperparameters but different random initialization on the Cityscapes dataset and evaluated them on the validation set calculating the mIoU per image. The resulting distributions of each model in the ensemble are converted into a CDF as is shown in Fig. 4. To have a stronger empirical evidence of the per-image mIoU performance distribution being constant for a dataset, we apply the two-sample Kolmogorov-Smirnov test on each pair of distri-

bution in the ensemble. The resulting p-values are at least  $> 0.95$ , hence supporting our hypothesis.

### 3.3 Datasets

For our sensor parameter optimization experiments we consider two datasets. First, the real-world Cityscapes dataset, which consists of 2,975 annotated images for training and 500 annotated images for validation. All images were captured in urban street scenes in German cities. Second, the synthetic dataset provided by the KI-A project [KI 20]. This dataset consists of 21,802 annotated training images and 5,164 validation images. The KI-A synthetic dataset comprises urban street scenes, similar to Cityscapes, and suburban to rural street scenes which are characterized by less traffic and less dense house placements, therefore more vegetation and terrain objects.

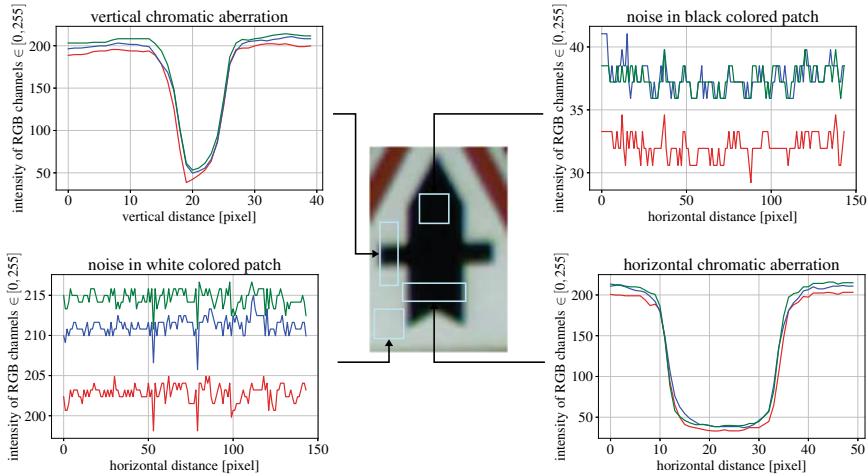
## 4 Results and Discussion

### 4.1 Sensor Parameter Extraction

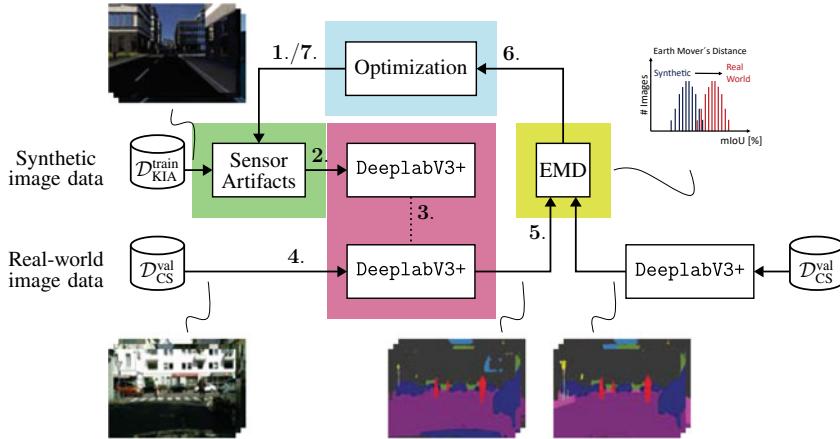
As a baseline for our sensor simulation, we analyzed images from the Cityscapes training data and measured the parameters. Sensor noise was extracted from about 10 images with uniformly colored areas ranging from dark to light colors. Chromatic aberration was extracted from 10 images with traffic signs on the outmost edges of the image, as can be seen in Fig. 5. The extracted values have been averaged over the count of images. The starting parameters of our optimization approach are then as follows: saturation = 2.0%, noise  $\sim \mathcal{N}(0, 3)$ ,  $\gamma = 0.8$ ,  $F = 4$ , and  $ca = 0.08$ .

### 4.2 Sensor Artifact Optimization Experiment

Utilizing the EMD as dataset divergence measure and the extracted sensor parameters from camera images of Cityscapes, we apply an optimization strategy to iteratively decrease the gap between the Cityscapes and the synthetic dataset [KI 20]. For optimization, we chose to use the trust region reflective (trf) method [SLA+15] as implemented in SciPy [VGO+20]. The trf is a least-squares minimization method to find the local minimum of a cost function given certain input variables. The cost function is the EMD from synthetic model and real-world model predictions on the same real-world validation dataset. The variables as input to the cost function are the parameters of the sensor artifact simulation. The trf method has the capability



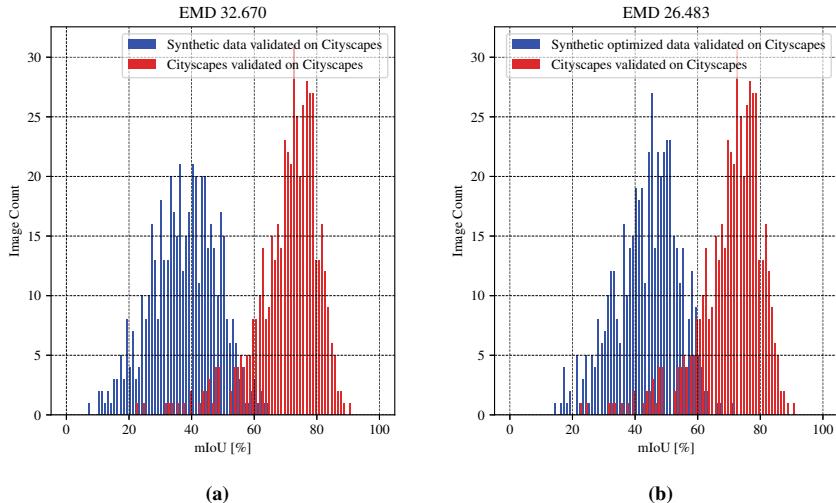
**Fig. 5** Exemplary manual extraction of sensor parameters from an extracted patch of a Cityscapes image on a traffic sign in the top right corner. Diagrams clockwise beginning top left: vertical chromatic aberration, noise level on black area, horizontal chromatic aberration, noise level on a plain white area



**Fig. 6** Optimization of sensor artifacts to decrease divergence between real and synthetic datasets

of bounding the variables to meaningful ranges. The stop criterion is met when the increase of parameter step size or decrease of the cost function is below  $10^{-6}$ .

The overall description of our optimization method is depicted in Fig. 6. *Step 1:* Initial parameters from the optimization method are applied in the sensor artifact simulation to the synthetic images. *Step 2:* The DeepLabV3+ model with ResNet101 backbone is pre-trained on 15 epochs on the original unmodified synthetic dataset and finetuned for one epoch on the synthetic dataset with applied sensor artifacts and



**Fig. 7** EMD domain divergence calculation of synthetic and optimized synthetic data to real-world images. **a** Comparison of synthetic data with Cityscapes data, **b** synthetic sensor artifact optimized dataset compared to the target dataset Cityscapes

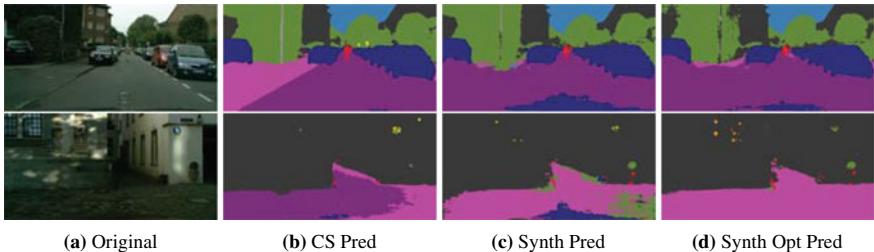
a learning rate of 0.1. *Step 3*: The model parameters are frozen and set to evaluate. *Step 4*: The model predicts on the validation set of the Cityscapes dataset. *Step 5*: The remaining domain divergence is measured by evaluation of the mIoU per image and calculation of the EMD to the evaluations of a model trained on Cityscapes. *Step 6*: The resulting EMD is fed as cost to the optimization method. *Step 7*: New parameters are set for the sensor artifact simulation, or the optimization ends if the stop criteria are met.

After iterating the parameter optimization with the trf method, we compare our optimized trained model with the unmodified synthetic dataset by their per-image mIoU distributions on the Cityscapes dataset. Figure 7 depicts the distributions resulting from this evaluation. The DeepLabV3+ model trained with the optimized sensor artifact simulation applied on the synthetic dataset outperforms the baseline and achieves an EMD score of 26.48, while decreasing the domain gap by 6.19. The resulting parameters are saturation = 2.11%, noise  $\sim \mathcal{N}(0, 3.0000005)$ ,  $\gamma = 0.800001$ ,  $F = 4$  and  $ca = 0.008000005$ . The parameters changed only slightly from the starting point, indicating the extracted parameters as good first choice.

An exemplary visual inspection of the results in Fig. 8 helps to understand the distribution shift and therefore the decreased EMD. While the best prediction performance image (top row) increased only slightly from the synthetic trained model (c) to the sensor artifact optimized model (d), the worst prediction case (bottom row) shows improved segmentation performance for the sensor-artifact-optimized model (d), in this case even better than the Cityscapes trained model (b).

**Table 2** Performance results as per-class mIoU, overall mIoU, and EMD domain divergence evaluated on Cityscapes with models trained on Cityscapes, our synthetic only, synthetic with random parameterized lens artifacts, and synthetic extracted parameterized lens artifacts datasets. For the latter two, there are models evaluated on Cityscapes with and without optimization of the parameters for the sensor lens artifact simulation. The model trained with optimized extracted parameters achieves the highest performance on the Cityscapes dataset

Model trained on	Optimized	road	sidewalk	building	pole	traffic light	traffic sign	vegetation	sky	human	car	truck	mIoU ←	EMD →
Cityscapes	no	97.50	81.21	92.22	54.54	57.28	70.26	92.33	93.98	82.55	93.67	81.62	81.56	—
w/o artifacts	no	60.46	23.51	71.99	10.70	26.00	13.47	75.72	74.70	51.27	34.46	1.74	40.37	32.67
w/ random artifacts	no	77.86	20.56	60.66	8.21	17.83	7.36	72.18	68.21	50.53	74.01	4.68	41.58	30.03
w/ extracted artifacts	no	80.65	27.17	66.54	10.43	21.64	11.87	68.82	67.99	59.22	70.06	<b>7.39</b>	44.71	29.84
w/ random artifacts	yes	<b>82.89</b>	<b>34.69</b>	71.11	11.29	16.66	9.12	69.81	<b>76.77</b>	<b>61.48</b>	<b>79.50</b>	5.30	45.76	26.58
w/ extracted artifacts	yes	79.62	30.30	<b>75.74</b>	<b>16.30</b>	<b>28.31</b>	<b>13.86</b>	<b>78.75</b>	70.88	59.74	64.06	6.42	<b>47.63</b>	<b>26.48</b>



**Fig. 8** Top row: Best performance predictions. Bottom row: Worst performance predictions. **a** Original, **b** Cityscapes-trained model prediction, **c** synthetically trained model prediction, and **d** sensor artifact optimized trained model prediction. While top performance increased only slightly, the optimization lead to more robust predictions in worst case, i.e., harder examples

We compare the overall mIoU performance on the Cityscapes datasets between models trained with the initial unmodified synthetic dataset, the synthetic dataset with random initialized lens artifact parameters, and the synthetic dataset with extracted parameters from Cityscapes with the baseline of a model trained on the Cityscapes dataset. Results are listed in Table 2 (rows 1–4). Additionally, for the random and the extracted parameters, we evaluate the performance with initial and optimized parameters, where the parameters have been optimized by our EMD minimization (rows 5 and 6). While the model without any sensor simulation achieves the lowest overall performance (row 2), the model with random parameter initialization achieves a slightly higher performance (row 3) and is surpassed by the model with the Cityscapes extracted parameters (row 4). Next, we take the models trained with optimized parameters into account (rows 5 and 6). Both models outperform all non-optimized experiment settings in terms of overall mIoU, with the model using optimized extracted parameters from Cityscapes showing the best overall mIoU (row 6). Concretely, the model trained with optimized random starting parameters achieves higher performance on classes road, sidewalk, human, and even significantly on the car class but still falls behind on five of the remaining classes and the overall performance on the Cityscapes dataset (row 5). Further, the random parameter optimized model took over 22 iterations to converge to its local minimum, whereas the optimization of extracted starting parameters only took six iterations until reaching a local minimum, making it more than three times faster to converge. Furthermore, it is shown that all models with applied sensor lens artifacts outperform the model trained without additional lens artifacts.

### 4.3 EMD Cross-evaluation

To get a deeper understanding of the implications of our EMD score, we evaluate our EMD results on a range of real-world and synthetic datasets for semantic segmentation. Including real-world datasets A2D2 [GKM+20], Cityscapes (CS) [COR+16],

**Table 3** Cross-domain divergence results of models trained on different real-world and synthetic datasets and evaluated on various validation or test sets of an average size of 1000 images. The domain divergence is measured with our proposed EMD measure; boldface values indicate the lowest divergence comparing our synthetic (Synth) and synthetic-optimized (SynthOpt) datasets, whereas underlined values indicate the lowest divergence values over all the datasets. The model trained with optimized lens artifacts applied to the synthetic images exhibits a smaller domain divergence than the model trained without lens artifacts

EMD ↓	A2D2	BDD100K	CS	GTAV	IDD	MV	SYNS	Synth	SynthOpt
A2D2	–	18.70	23.46	37.84	20.03	<b>10.72</b>	46.78	34.95	<b>29.32</b>
BDD100K	6.36	–	9.45	22.14	7.26	<b>1.42</b>	36.33	26.43	<b>21.54</b>
CS	10.90	12.09	–	36.42	13.01	<b>4.08</b>	20.62	32.66	<b>26.48</b>
GTAV	33.28	28.37	29.72	–	30.55	<b>23.30</b>	37.53	36.08	<b>32.94</b>
IDD	24.37	19.83	24.71	34.71	–	<b>12.81</b>	46.23	41.64	<b>36.95</b>
MV	10.63	10.36	14.34	28.35	<b>9.20</b>	–	35.97	30.35	<b>27.03</b>
SYNS	25.45	31.46	23.64	45.16	25.12	<b>23.45</b>	–	43.76	<b>43.56</b>

Berkeley Deep Drive (BDD100K) [YCW+20], Mapillary Vistas (MV) [NOBK17], India Driving Dataset (IDD) [VSN+18], as well as synthetic GTAV [RVRK16], our synthetic (Synth and SynthOpt) [KI 20], and Syncscapes (SYNS) [WU18] datasets. In Table 3 the results of cross-domain analysis measured with the EMD score are depicted. The columns denote that a DeepLabV3+ model has been trained on the corresponding dataset, i.e., the source dataset, whereas the rows denote the datasets it was evaluated on, i.e., the target datasets. Our optimized synthetic dataset achieves lower EMD scores, shown in boldface, than the synthetic baseline. While the domain divergence decrease is high on real datasets, the divergence decreased only marginally for the other synthetic datasets. Inspecting the EMD result on all datasets, the lowest divergence values are indicated by underline; the MV dataset shows to be closest to all the other evaluated datasets.

To set our measure in relation to established domain distance measures, we calculated the FID from each of our considered datasets to one another. The results are shown in Table 4. The FID, defined in (7), is the Wasserstein-2 distance of feature vectors from the InceptionV3 [SVI+16] network sampled on the two datasets to be compared with each other.

Again, boldface values indicate the lowest FID values between the synthetic (Synth) and synthetic-optimized (SynthOpt) datasets, whereas underlined values indicate the lowest values of all datasets. Here, only 4 out of the 7 datasets are closer, measured by the FID, to the synthetic-optimized dataset than to the original dataset. Furthermore, the FID sees the CS and the SYNS dataset closer to one another than the EMD divergence measure, while the MV dataset shows the lowest FID among the other evaluated datasets.

FID and EMD somewhat agree, if we evaluate the distance as minimum per-row in both tables, that the Mapillary Vistas dataset is in most cases the dataset that is closest to all other datasets.

**Table 4** Cross-domain distance results measured with the Fréchet inception distance (FID). Lowest FID between synthetic (Synth) and synthetic optimized (SynthOpt) datasets are in boldface, whereas the lowest FID values over all datasets are underlined

FID ↓	A2D2	BDD100K	CS	GTAV	IDD	MV	SYNS	Synth	SynthOpt
A2D2	–	60.16	98.46	78.16	58.75	<b>41.84</b>	109.35	<b>116.54</b>	121.55
BDD100K	<b>60.16</b>	–	59.90	62.42	52.15	<u>29.66</u>	74.871	115.51	<b>109.08</b>
CS	98.46	<u>59.90</u>	–	85.81	68.92	59.69	<u>43.87</u>	119.97	<b>112.42</b>
GTAV	78.16	62.42	85.81	–	74.08	<u>51.00</u>	89.62	92.513	<b>92.24</b>
IDD	58.75	52.15	68.92	74.08	–	<u>37.36</u>	64.09	<b>118.06</b>	125.30
MV	41.84	<u>29.66</u>	59.69	51.00	37.36	–	70.24	<b>74.46</b>	78.66
SYNS	109.35	74.87	<u>43.87</u>	89.62	64.09	70.24	–	113.77	<b>108.3</b>

Now, calculating the minimum per-column in both tables, the benefit of our asymmetric EMD comes to the light. The minimum per-column values of the FID are unchanged due to the diagonal symmetry of the cross-evaluation matrix stemming from the inherent symmetry of the measure. However, the EMD regards the BDD100K as the closest dataset. An intuitive explanation for the different minimum observations of the EMD is as follows: Training with many images exhibiting different geospatial and sensor properties of the Mapillary Vistas dataset covers a very broad domain and results in good generalization capability and therefore evaluation performance. Training with any of the other datasets cannot generalize well to the vast domain of Mapillary Vistas but to the rather constrained domain of BDD100K, which consists of lower resolution images with heavy compression artifacts, where even a model that has been trained on BDD100K does not generalize well on.

The asymmetric nature of our EMD allows for a more thorough and complex analysis of dataset discrepancies, when applied to the tasks of visual understanding, e.g., semantic segmentation, which otherwise cannot be captured by inherently symmetric distance metrics such as FID. Contrasting to [LLFW20], we could with our evaluation method not identify a consistency between FID and the generalization divergence, i.e., our EMD measure.

## 5 Conclusions

In this chapter, we could demonstrate that by utilizing the performance metric per image as a proxy distribution for a dataset and the earth mover’s distance (EMD) as a divergence measure between distributions, one can decrease visual differences of a synthetic dataset through optimization and increase the viability of CGI for training and validation purposes of perceptive AI. To reinforce our argument for per-image performance measures as proxy distributions, we showed that training an ensemble of a fixed model with different random starting conditions but with the same

hyperparameters leads to the same per-image performance distributions when these ensemble models are evaluated on the validation set of the training dataset. When utilizing synthetic imagery for validation, the domain gap, due to visual differences between real and computer-generated images, is hindering the applicability of these datasets. As a step toward decreasing the visual differences, we apply the proposed divergence measure as a cost function to an optimization which varies the parameters of the sensor artifact simulation, while trying to re-create the sensor artifacts that the real-world dataset exhibits. As starting point of the sensor artifact parameters, we extracted empirically the values from chosen images of the real-world dataset. The optimization improved the visual difference between the real-world and the optimized synthetic dataset measurably by the EMD and we could show that even when starting with random initialized parameters we can decrease the EMD and increase the mIoU on the target datasets. When measuring the divergence after parameter optimization to other real-world and synthetic datasets, we could show that the EMD decreases for all considered datasets but when measured by the FID only four of the datasets are closer. As the EMD is derived from the mIoU per image, it is an indicator of performance on the target dataset, whereas the FID fails to relate with performance. Effective minimization of the visual difference between synthetic and real-world datasets with the EMD domain divergence measure is one step further toward fully utilizing CGI for validation of perceptive AI functions.

**Acknowledgements** The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project “Methoden und Maßnahmen zur Absicherung von KI-basierten Wahrnehmungsfunktionen für das automatisierte Fahren (KI Absicherung)”. The authors would like to thank the consortium for the successful cooperation.

## References

- [BB95] W. Burger, M.J. Barth, Virtual reality for enhanced computer vision, in J. Rix, S. Haas, J. Teixeira (eds.), *Virtual Prototyping: Virtual Environments and the Product Design Process* (Springer, 1995), pp. 247–257
- [Bis95] M. Christopher Bishop, Training with noise is equivalent to Tikhonov regularization. *Neural Comput.* 7(1), 108–116 (1995)
- [BSAG21] M. Binkowski, D.J. Sutherland, M. Arbel, A. Gretton, *Demystifying MMD GANs*, Jan. 2021, pp. 1–36. [arxiv:1801.01401](https://arxiv.org/abs/1801.01401)
- [CLP13] G. Csurka, D. Larlus, F. Perronnin, What is a good evaluation measure for semantic segmentation? in *Proceedings of the British Machine Vision Conference (BMVC)*, Bristol, UK, Sept. 2013, pp. 1–11
- [COR+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The Cityscapes dataset for semantic urban scene understanding, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 3213–3223
- [CSVJR18] A. Carlson, K.A. Skinner, R. Vasudevan, M. Johnson-Roberson, Modeling camera effects to improve visual learning from synthetic data, in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, Munich, Germany, Aug. 2018, pp. 505–520

- [CSVJR19] A. Carlson, K.A. Skinner, R. Vasudevan, M. Johnson-Roberson, *Sensor Transfer: Learning Optimal Sensor Effect Image Augmentation for Sim-to-Real Domain Adaptation*, Jan. 2019, pp. 1–8. [arxiv:1809.06256](https://arxiv.org/abs/1809.06256)
- [CV14] V. Chari, A. Veeraraghavan, L. Distortion, Radial distortion, in *Computer Vision: A Reference Guide*, ed. by K. Ikeuchi (Springer, Boston, MA, 2014), pp. 443–445
- [CZP+18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-Decoder with atrous separable convolution for semantic image segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, Sept. 2018, pp. 833–851
- [dCCN+16] G.B.P. da Costa, W.A. Contato, T.S. Nazaré, J.E.S. do Batista Neto, M. Ponti, *An Empirical Study on the Effects of Different Types of Noise in Image Classification Tasks*, Sept. 2016, pp. 1–6. [arxiv:1609.02781](https://arxiv.org/abs/1609.02781)
- [DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, ImageNet: a large-scale hierarchical image database, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, June 2009, pp. 248–255
- [DRC+17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, V. Koltun, CARLA: an open urban driving simulator, in *Proceedings of the Conference on Robot Learning CORL*, Mountain View, CA, USA, Nov. 2017, pp. 1–16
- [FMWR18] E. Fernandez-Moral, R. Martins, D. Wolf, P. Rives, A new metric for evaluating semantic segmentation: leveraging global and contour accuracy, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China, June 2018, pp. 1051–1056
- [GKM+20] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A.S. Chung, L. Hauswald, V.H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, P. Schuberth, *A2D2: Audi Autonomous Driving Dataset*, April 2020, pp. 1–10. [arxiv:2004.06320](https://arxiv.org/abs/2004.06320)
- [GL15] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, July 2015, pp. 1180–1189
- [HRU+17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 6626–6637
- [HTP+18] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, CyCADA: cycle-consistent adversarial domain adaptation, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden, July 2018, pp. 1989–1998
- [HZRS16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 770–778
- [KI 20] KI Absicherung Consortium. KI Absicherung: Safe AI for Automated Driving (2020). Assessed 18 Nov. 2021
- [KPT+17] S. Kolouri, S.R. Park, M. Thorpe, D. Slepcev, G.K. Rohde, Optimal mass transport: signal processing and machine-learning applications. *IEEE Signal Process. Mag.* **34**(4), 43–59 (2017)
- [LCWJ15] M. Long, Y. Cao, J. Wang, M.I. Jordan, Learning transferable features with deep adaptation networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, July 2015, pp. 97–105
- [LLFW20] Z. Liu, T. Lian, J. Farrell, B. Wandell, Neural network generalization: the impact of camera parameters. *IEEE Access* **8**, 10443–10454 (2020)
- [NdCCP18] T.S. Nazaré, G.B.P. da Costa, W.A. Contato, M. Ponti, Deep convolutional neural networks and noisy images, in *Proceedings of the Iberoamerican Congress on Pattern Recognition (CIARP)*, Madrid, Spain, Nov. 2018, pp. 416–424

- [NOBK17] G. Neuhold, T. Ollmann, S. Rota Bulò, P. Kortschieder, The mapillary vistas dataset for semantic understanding of street scenes, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 4990–4999
- [PTKY09] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, Pasadena, CA, USA, July 2009, pp. 1187–1192
- [RD14] R. Ramanath, M.S. Drew, Color Spaces, in *Computer Vision: A Reference Guide*. ed. by K. Ikeuchi (Springer, Boston, MA, 2014), pp. 123–132
- [RHK17] S.R. Richter, Z. Hayder, V. Koltun, Playing for benchmarks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2232–2241
- [RSM+16] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A.M. Lopez, The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 3234–3243
- [RTC17] A. Ramdas, N. García Trillo, M. Cuturi, On Wasserstein two sample testing and related families of nonparametric tests. *Entropy* **19**(2), 47 (2017)
- [RTG+19] H. Rezatofighi, N. Tsoi, J.Y. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: a metric and a loss for bounding box regression, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 658–666
- [RV19] S.V. Ravuri, O. Vinyals, Seeing is not necessarily believing: limitations of BigGANs for data augmentation, in: *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, New Orleans, LA, USA, June 2019, pp. 1–5
- [RVRK16] S.R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for data: ground truth from computer games, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 102–118
- [SAS+18] F.S. Saleh, M.S. Aliakbarian, M. Salzmann, L. Petersson, J.M. Alvarez, Effective use of synthetic data for urban scene semantic segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, Sept. 2018, pp. 84–100
- [SGH20] Q.S. Sha, O. Grau, K. Hagn, DNN analysis through synthetic data variation, in *Proceedings of the ACM Computer Science in Cars Symposium (CSCS)*, virtual conference, Dec. 2020, pp. 1–10
- [SGZ+16] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, Xi Chen, *Improved Techniques for Training GANs*, June 2016, pp. 1–10. [arxiv:1606.03498](https://arxiv.org/abs/1606.03498)
- [SLA+15] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, July 2015, pp. 1889–1897
- [Stu14] P. Sturm, Pinhole Camera Model, in *Computer Vision: A Reference Guide*. ed. by K. Ikeuchi (Springer, Boston, MA, 2014), pp. 610–613
- [SVI+16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 2818–2826
- [THS+18] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, M. Chandraker, Learning to adapt structured output space for semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 7472–7481
- [THSD17] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017, pp. 2962–2971
- [VGO+20] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson,

- K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, I. Polat, Y. Feng, E.W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. Fundamental algorithms for scientific computing in python. *SciPy 1.0. Nat. Methods* **17**, 261–272 (2020)
- [VSN+18] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, C.V. Jawahar, *IDD: A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments*, Nov. 2018, pp. 1–9. [arxiv:1811.10200](https://arxiv.org/abs/1811.10200)
- [WU18] M. Wrenninge, J. Unger, *Synscapes: A Photorealistic Synthetic Dataset for Street Scene Parsing*, Oct. 2018, pp. 1–13. [arxiv:1810.08705](https://arxiv.org/abs/1810.08705)
- [YCW+20] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, T. Darrell, BDD100K: a diverse driving dataset for heterogeneous multitask learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference, June 2020, pp. 2636–2645

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



# Improved DNN Robustness by Multi-task Training with an Auxiliary Self-Supervised Task



Marvin Klingner and Tim Fingscheidt

**Abstract** While deep neural networks for environment perception tasks in autonomous driving systems often achieve impressive performance on clean and well-prepared images, their robustness under real conditions, i.e., on images being perturbed with noise patterns or adversarial attacks, is often subject to a significantly decreased performance. In this chapter, we address this problem for the task of semantic segmentation by proposing multi-task training with the additional task of depth estimation with the goal to improve the DNN robustness. This method has a very wide potential applicability as the additional depth estimation task can be trained in a self-supervised fashion, relying only on unlabeled image sequences during training. The final trained segmentation DNN is, however, still applicable on a single-image basis during inference without additional computational overhead compared to the single-task model. Additionally, our evaluation introduces a measure which allows for a meaningful comparison between different noise and attack types. We show the effectiveness of our approach on the Cityscapes and KITTI datasets, where our method improves the DNN performance w.r.t. the single-task baseline in terms of robustness against multiple noise and adversarial attack types, which is supplemented by an improved absolute prediction performance of the resulting DNN.

## 1 Introduction

**Motivation:** For a safe operation of highly automated driving systems, a reliable perception of the environment is crucial. Various perception tasks such as semantic segmentation [LSD15], [CPK+18], depth estimation [EPF14], [ZBSL17], or optical flow estimation [LLKX19] are often implemented by deep neural networks (DNNs). The output of these DNNs is then used to build a model of the environment, which

---

M. Klingner (✉) · T. Fingscheidt

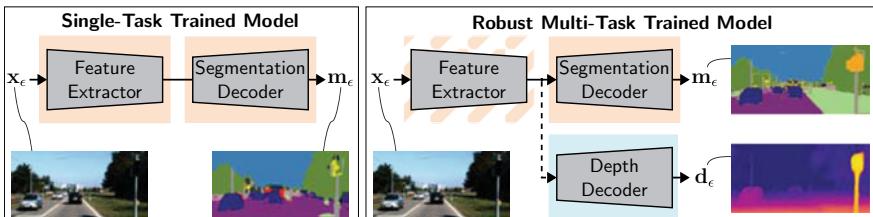
Institute for Communications Technology (IfN), Technische Universität Braunschweig,  
Schleinitzstr. 22, 38106 Braunschweig, Germany  
e-mail: [m.klingner@tu-bs.de](mailto:m.klingner@tu-bs.de)

T. Fingscheidt  
e-mail: [t.fingscheidt@tu-bs.de](mailto:t.fingscheidt@tu-bs.de)

is subsequently used for decision making in high-level planning systems. As many decisions are thereby executed upon the DNN predictions, these predictions need to be reliable under all kinds of environmental changes. This is, however, in contrast to typical DNN training, which is carried out on annotated datasets [COR+16], [NOBK17] covering only a small portion of possible environment variations as more diverse datasets are often not available. During deployment, the DNN performance can therefore drop significantly due to domain shifts not covered in the training dataset. These include, for example, noises induced by different optical sensors, brightness and weather changes, deployment in a different country, or even directed adversarial attacks [GSS15]. Therefore, for a safe deployment of DNNs in autonomous driving vehicles, their performance also needs to be *robust* w.r.t. these environment changes. In this chapter we aim at an improved robustness for the task of semantic segmentation.

**Robustness of DNNs:** Improving the robustness of DNNs is a highly relevant research topic for applications such as autonomous driving, where the perception system has to deal with varying environmental conditions and potentially even with adversarial attacks. As adversarial attacks usually rely on the computation of gradients on the input [GSS15], [MMS+18], it has been proposed to apply a non-differentiable pre-processing [GRCvdM18], [LLL+19] such that the impact of the perturbation on the DNN performance is reduced. However, the gradients of these pre-processings can be approximated [ACW18] so that these strategies usually only make an attack harder to calculate. Our approach therefore focuses on a more robust training of the DNN where often adversarial examples or image augmentations [GSS15], [MMS+18] are utilized already during training such that the DNN will be more robust to those during inference. While these approaches often induce a decreased performance on clean images, we aim at developing a method improving performance as well as robustness.

**Multi-task learning:** The training of several tasks in a single multi-task DNN, i.e., multi-task learning, is known to improve the absolute prediction performance of the single tasks as well as the efficiency of their computation due to the shared network parts [EF15], [KTMFs20]. In this chapter, as shown in Fig. 1, we use a multi-task



**Fig. 1** Robustness improvement through multi-task learning during training. When the input  $x_\epsilon$  is subject to perturbations of strength  $\epsilon$ , the output (segmentation  $m_\epsilon$ , depth  $d_\epsilon$ ) of a multi-task trained model (right-hand side) is still well predicted, while the output quality of the single-task trained model (left-hand side) is strongly impaired

network for semantic segmentation and depth estimation, for which these properties have also been shown in several works [KGC18], [KTMFs20]. Particularly, we follow [KBFs20] in focusing on the additional robustness-increasing properties of such a multi-task training. Moreover, it is important to note that the depth estimation can be trained in a self-supervised fashion, which only requires short unlabeled video sequences and thereby usually does not impose large additional requirements in terms of data. While it was still necessary to manually find a good weighting between the single tasks in [KBFs20], we apply the GradNorm task weighting strategy [CBLR18], which automatically sets and adapts the task weighting according to the task-specific training progress. Also, we show the method’s applicability across a wider range of datasets.

**Comparability of perturbations:** Up to now, a wide variety of different possible adversarial attack and noise types has been proposed [GW08], [GSS15], [CW17], [MMS+18]. However, each of these image perturbations is characterized by its own set of parameters, e.g., the standard deviation of the Gaussian noise distribution or the maximum noise value for many adversarial attacks. This makes it hard to compare perturbation strengths in a single mutual evaluation. To better compare these effects and to be able to draw conclusions between different noise and attack types, we employ a measure based on the signal-to-noise ratio, which enables a fair comparison between different perturbation types in terms of strength.

**Contributions:** To sum up, our contribution with this chapter is threefold. First, we propose a multi-task learning strategy with the task of self-supervised monocular depth estimation to improve a DNN’s robustness. Second, we provide a detailed analysis of the positive effects of this method on DNN performance and robustness to multiple input perturbations. Third, we employ a general measure for perturbation strength, thereby making different noise and attack perturbation types comparable.

## 2 Related Works

In this section, we give an overview of related multi-task learning approaches for depth estimation and semantic segmentation. Afterward, we discuss methods to improve the robustness of DNNs focusing on approaches for semantic segmentation.

**Multi-task learning** The performance of basic perception tasks such as semantic segmentation [LSD15] and depth estimation [EF15] has increased significantly by employing fully convolutional neural networks. Furthermore, these tasks can be learned jointly in a multi-task learning setup by employing a shared encoder, which was shown to be of mutual benefit for both tasks through the more extensively learned scene understanding [EF15], [KGC18]. This could be further facilitated by combining the loss functions to enforce cross-task consistency during optimization [KGC18], [XOWS18], [ZCX+19]. For depth estimation, the research

focus shifted from supervised to self-supervised training techniques [ZBSL17], [GMAB17], [GMAFB19], due to the more general applicability on unlabeled videos, which are more readily available than labeled datasets. Accordingly, such techniques have also been employed in multi-task setups with semantic segmentation [CLLW19], [GHL+20], [NVA19]. Usually, the focus of these works is to improve the absolute prediction performance of the single involved tasks. Thereby, it was proposed to exclude the influence of dynamic objects such as cars or pedestrians [KTMFs20], [RKKY+21b], [RKKY+21a] to employ pixel-adaptive convolution for improved semantic guidance [GHL+20], [RKKY+21b], or to enforce cross-task (edge) consistency between both tasks' outputs [CLLW19], [ZBL20], [YZS+18], [MLR+19]. The approaches have also been extended from simple pinhole camera models to more general fisheye camera models [RKKY+21b], [RKKY+21a]. For semantic segmentation, the depth estimation can also be beneficial in unsupervised domain adaptation approaches [KTMFs20], [LRLG19], [VJB+19].

In this chapter, we build upon these advances by employing a multi-task learning setup of self-supervised depth estimation and semantic segmentation as introduced by Klingner et al. [KTMFs20]. However, in contrast to [KTMFs20], we put the focus rather on the robustness instead of the absolute prediction performance of the resulting semantic segmentation DNN.

**Robustness of (semantic segmentation) DNNs** While DNNs can achieve an impressive performance on clean images, their performance is usually not robust w.r.t. additive noise [Bis95], [HK92]. For safety-critical application this is a particularly high risk, if this noise is calculated in a way that it is nearly not recognizable if added to the image, but still heavily impairs the performance, as shown by the works on adversarial examples by Szegedy et al. [SZS+14]. Consequently, subsequent works developed various kinds of targeted and non-targeted perturbations, calculated in an image-specific fashion and optimized to fool the DNN. These adversarial examples range from simple non-iterative methods such as the fast gradient sign method (FGSM) [GSS15] to more complex iteratively calculated methods such as the momentum iterative fast gradient sign method (MI-FGSM) [DLP+18], the Carlini and Wagner attack (CW) [CW17], or the projected gradient descent (PGD) method [MMS+18]. While these image-specific attacks may not be a relevant danger in real applications due to the high computational effort per image, the existence of image-agnostic adversarial perturbations (UAPs) has also been shown, e.g., by prior-driven uncertainty estimation (PD-UAP) [LJL+19], universal adversarial perturbation (UAP) [MDFFF17], or fast feature fool (FFF) [MGB17]. Although most of these works focus on the rather simple task of image classification, the applicability of these attacks to semantic segmentation is well known [AMT18], [MGR19], [BLK+21]. Furthermore, the attacks can be designed in a fashion such that the semantic segmentation outputs are completely wrong but still appear realistic [ASG+19], [MCKBF17].

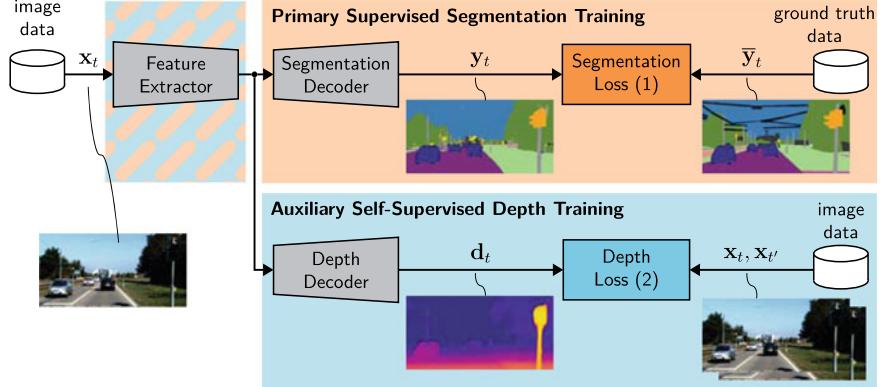
The vulnerability of DNNs to different kinds of adversarial attacks has encouraged research in defense methods, which improve the robustness against these perturbations. They can be roughly divided into three categories. First, for *gradient masking*, the idea is to prevent a potential attacker from calculating the gradients

[CBG+17], [MMS+18], [PMG+17]. However, Athalye et al. [ACW18] showed that the perturbations can be calculated from a different model. Also, they can be calculated in an image-agnostic (universal) fashion [MCKBF17], [MDFFF17], see also the Chapter ‘‘Improving Transferability of Generated Universal Adversarial Perturbations for Image Classification and Segmentation’’ [HAMFs22], such that gradient masking methods cannot prevent the network’s vulnerability to adversarial attacks under all conditions. Second, through *input transformations*, the perturbations can be removed from the input image [GR CvM18], [KBV+21], [LLL+19], e.g., by JPEG image compression [ATT18] or by incorporation of randomness [RSFM19], [XWZ+18]. While this improves the performance on attacked images, these transformations usually impair the performance on clean images, introducing an unfavorable trade-off. Third, *redundancy* among two or three networks serving the same task can be exploited, if networks are enforced to reveal some independence [BHSFs19], [BKV+20]. Fourth, *robust training* methods can be employed such that the DNN is less sensitive to fail because of attacks from the start. Common approaches include, e.g., adversarial training [GSS15], [MMS+18], robustness-oriented loss functions [BHSFs19], [BKV+20], [CLC+19], or self-supervision from auxiliary tasks [CLC+20], [HMKS19], [ZYC+20] during (pre-)training. Our chapter also focuses on robustness through self-supervision, where we introduce multi-task learning with the auxiliary task of depth estimation as a method to improve a semantic segmentation DNN’s robustness, which can be seen as an extension of [KBFs20]. Thereby, we achieve an improved performance, while also being more robust and even introducing a second useful task for scene perception. Compared to [KBFs20], we improve the multi-task learning strategy to reduce the need for manual task weighting and provide experiments across a wider range of datasets.

### 3 Multi-task Training Approach

Describing our multi-task learning approach, we start by defining our primary semantic segmentation task, followed by the auxiliary task of depth estimation. Finally, we describe how both tasks are trained in a multi-task learning setup as shown in Fig. 2.

**Primary semantic segmentation:** Semantic segmentation is defined as the pixel-wise classification of an input image  $\mathbf{x}_t = (\mathbf{x}_{t,i}) \in \mathcal{G}^{H \times W \times C}$  at time  $t$ , with height  $H$ , width  $W$ , number of channels  $C = 3$ , and  $\mathcal{G} = \{0, 1, \dots, 255\}$  (cf. top branch of Fig. 2). Accordingly, for each pixel  $\mathbf{x}_{t,i} \in \mathcal{G}^3$  with pixel index  $i \in \mathcal{I} = \{1, \dots, H \cdot W\}$  an output  $\mathbf{y}_{t,i} = (y_{t,i,s}) \in \mathbb{I}^{|\mathcal{S}|}$ ,  $\mathbb{I} = [0, 1]$  is predicted, which can be interpreted as the posterior probabilities that the pixel at index  $i$  belongs to a class  $s \in \mathcal{S}$  from the set of classes  $\mathcal{S} = \{1, 2, \dots, |\mathcal{S}|\}$ , with the number of classes  $|\mathcal{S}|$ . The final predicted semantic class  $m_{t,i} \in \mathcal{S}$  is determined by applying the argmax operation  $m_{t,i} = \text{argmax}_{s \in \mathcal{S}} y_{t,i,s}$ . Thereby, the network output  $\mathbf{y}_t \in \mathbb{I}^{H \times W \times |\mathcal{S}|}$  is converted to a segmentation mask  $\mathbf{m}_t = (m_{t,i}) \in \mathcal{S}^{H \times W}$ .



**Fig. 2** Multi-task training setup across domains for the joint learning of depth estimation and semantic segmentation. While the semantic segmentation is trained on labeled image pairs, the depth estimation is trained on unlabeled image sequences

As shown in Fig. 2, training of a semantic segmentation network requires ground truth labels  $\bar{y}_t \in \{0, 1\}^{H \times W \times |\mathcal{S}|}$ , which are derived from the ground truth segmentation mask  $\bar{\mathbf{m}} \in \mathcal{S}^{H \times W}$  represented by a one-hot encoding. These are utilized to train the network by a cross-entropy loss function as

$$J_t^{\text{seg}} = -\frac{1}{H \cdot W} \sum_{i \in \mathcal{I}} \sum_{s \in \mathcal{S}} w_s \bar{y}_{t,i,s} \cdot \log(y_{t,i,s}), \quad (1)$$

where  $w_s$  are class-wise weights obtained as outlined in [PCKC16], and  $\bar{y}_{t,i,s} \in \{0, 1\}$  are the single elements from the one-hot encoded ground truth tensor  $\bar{y}_t = (\bar{y}_{t,i,s})$ .

**Auxiliary depth estimation:** Aiming at a more robust feature representation, we employ the auxiliary depth estimation task, which can be trained on unlabeled image sequences, as shown in Fig. 2, bottom part. During training, the depth estimation DNN predicts a depth map  $\mathbf{d}_t = (d_{t,i}) \in \mathbb{D}^{H \times W}$ , representing the distance of each pixel from the camera plane, where  $\mathbb{D} = [d_{\min}, d_{\max}]$  represents the space of considered depth values constrained between the lower bound  $d_{\min}$  and the upper bound  $d_{\max}$ . We optimize the depth estimation DNN in a self-supervised fashion by considering two loss terms: First, we make use of the image reconstruction term  $J_t^{\text{ph}}$  (photometric loss), which essentially uses the depth estimation in conjunction with the relative pose between two images, to optimize the camera reprojection models between two consecutive frames of a video. Second, we apply a smoothness loss term  $J_t^{\text{sm}}$ , allowing high gradients in the depth estimate's values only in image regions with high color gradients, such that the total loss can be written as

$$J_t^{\text{depth}} = J_t^{\text{ph}} + \beta J_t^{\text{sm}}, \quad (2)$$

where  $\beta = 10^{-3}$  is adopted from previous works [CPMA19], [GMAFB19], [KTMFs20].

To optimize the network, we rely solely on sequential pairs of images  $\mathbf{x}_t, \mathbf{x}_{t'}$  with  $t' \in \mathcal{T}' = \{t-1, t+1\}$ , which are taken from a video. These pairs are passed to an additional pose network, which predicts the relative poses  $\mathbf{T}_{t \rightarrow t'} \in SE(3)$  between the image pairs, where  $SE(3)$  is the special Euclidean group representing the set of all possible rotations and translations [Sze10]. The network predicts this transformation in an axis-angle representation such that only the six degrees of freedom are predicted, which are canonically converted to a  $4 \times 4$  matrix for further processing. By letting the depth network predict the depth  $\mathbf{d}_t$ , both outputs, i.e., the depth  $\mathbf{d}_t$  and the poses  $\mathbf{T}_{t \rightarrow t'}$ , can be used to project the image frame  $\mathbf{x}_{t'}$  at time  $t'$  onto the pixel coordinates of the image frame  $\mathbf{x}_t$ , which results in two projected images  $\mathbf{x}_{t' \rightarrow t}, t' \in \mathcal{T}'$  (for a detailed description, we refer to [KTMFs20]). Conclusively, the reprojection model can be optimized by minimizing the pixel-wise distance between the projected images  $\mathbf{x}_{t' \rightarrow t} = (\mathbf{x}_{t' \rightarrow t, i})$  and the actual images  $\mathbf{x}_t = (\mathbf{x}_{t, i})$  as

$$J_t^{\text{ph}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \min_{t' \in \mathcal{T}'} \left( \frac{\gamma}{2} (1 - \text{SSIM}_i(\mathbf{x}_t, \mathbf{x}_{t' \rightarrow t})) + (1 - \gamma) \frac{1}{C} \|\mathbf{x}_{t, i} - \mathbf{x}_{t' \rightarrow t, i}\|_1 \right). \quad (3)$$

This so-called minimum reprojection loss or photometric loss [GMAFB19] utilizes a mixture of the structural similarity (SSIM) difference term  $\text{SSIM}_i(\cdot) \in \mathbb{I}$ , with  $\mathbb{I} = [0, 1]$ , and is computed on  $3 \times 3$  patches of the input, and an  $L_1$  difference term  $\|\cdot\|_1$  computed over all  $C = 3$  gray value channels. For optimal absolute prediction performance, the terms are weighted by a factor  $\gamma = 0.85$ , chosen as in previous approaches [CPMA19], [GMAFB19], [KTMFs20], [YS18]. The depth and pose networks are then implicitly optimized, as their outputs are the parameters of the projection model, used to obtain the projected image  $\mathbf{x}_{t' \rightarrow t}$ , whose distance to the actual image  $\mathbf{x}_t$  is minimized by (3).

As the photometric loss  $J_t^{\text{ph}}$  does not enforce a relationship in the depth map between depth values of neighboring pixels, we use a second smoothness loss term  $J_t^{\text{sm}}$  [GMAB17], allowing non-smoothness in the depth map  $\mathbf{d}_t$  only in image locations with strong color gradients. This loss is computed on the mean-normalized inverse depth  $\check{\rho}_t \in \mathbb{R}^{H \times W}$ , whose elements can be obtained from the depth map as  $\check{\rho}_{t, i} = \frac{\rho_{t, i}}{\frac{1}{HW} \sum_{j \in \mathcal{I}} \rho_{t, j}}$  with  $\rho_{t, i} = \frac{1}{d_{t, i}}$ . Thereby, the loss can be formulated as

$$J_t^{\text{sm}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \left( |\partial_h \check{\rho}_{t, i}| \exp \left( -\frac{1}{C} \|\partial_h \mathbf{x}_{t, i}\|_1 \right) + |\partial_w \check{\rho}_{t, i}| \exp \left( -\frac{1}{C} \|\partial_w \mathbf{x}_{t, i}\|_1 \right) \right), \quad (4)$$

with one-dimensional difference quotients  $\partial_h \check{\rho}_{t, i} = \partial_h \check{\rho}_{t, (h, w)} = \check{\rho}_{t, (h, w)} - \check{\rho}_{t, (h-1, w)}$  and  $\partial_w \check{\rho}_{t, i} = \partial_w \check{\rho}_{t, (h, w)} = \check{\rho}_{t, (h, w)} - \check{\rho}_{t, (h, w-1)}$  defined with respect to the height and width dimensions of the image, respectively. The indices  $h$  and  $w$  represent the exact pixel position in two dimensions, where  $h \in \{2, \dots, H\}$  and  $w \in \{2, \dots, W\}$ , where we exclude  $h = 1$  and  $w = 1$  to ensure the existence of a preceding pixel in height and width dimensions.

**Multi-task learning:** To train our model directly in a one-stage fashion without a pre-trained semantic segmentation network, we choose a multi-task setup with a shared encoder and two task-specific decoder heads as shown in Fig. 2. The decoders for semantic segmentation and depth estimation are optimized for their respective tasks according to the losses defined in (1) and (2), respectively, while the encoder is optimized for both tasks. As in [GL15], [KTMFs20], we let the task-specific gradients  $\mathbf{g}_t^{\text{depth}}$  and  $\mathbf{g}_t^{\text{seg}}$  propagate unscaled in the respective decoders, while their influence when reaching the encoder layers during backpropagation are scaled as

$$\mathbf{g}_t^{\text{total}} = \lambda^{\text{depth}} \mathbf{g}_t^{\text{depth}} + \lambda^{\text{seg}} \mathbf{g}_t^{\text{seg}}, \quad (5)$$

where the scalar weight  $\lambda^{\text{seg}}$  and  $\lambda^{\text{depth}}$  determine the weighting between the two tasks. In the other encoder layers, the backpropagation is then executed as usual. By scaling the gradients instead of the losses, the two decoders can learn optimal task-specific weights, while their influence on the shared encoder can be scaled to the optimal ratio using (5). Thereby, the encoder does not only learn optimal features for the semantic segmentation task, but can also take profit from the additional data accessible by the depth estimation, which can be trained on arbitrary videos.

In this chapter, we compare two kinds of task weightings: First, we apply the gradient weighting (GW) from [KBFs20], where we set  $\lambda^{\text{seg}} = \lambda$  and  $\lambda^{\text{depth}} = 1 - \lambda$  to scale the influence of both tasks. Here, the hyperparameter  $\lambda$  needs to be tuned to find the optimal weighting of both tasks. However, the results from [KBFs20] show that for a badly chosen hyperparameter  $\lambda$ , performance decreases drastically, which is why we apply the GradNorm (GN) multi-task learning technique [CBLR18], where the scale factors  $\lambda^{\text{seg}}$  and  $\lambda^{\text{depth}}$  are reformulated as learnable parameters  $\lambda^{\text{seg}}(\tau)$  and  $\lambda^{\text{depth}}(\tau)$  which are adapted at each learning step  $\tau$ . For simplicity, we henceforth abbreviate the tasks with an index  $k$  with  $k \in \mathcal{K} = \{\text{depth}, \text{seg}\}$ . Thereby, the loss function used for optimization of the scale factors, i.e., the task weights, is calculated as follows:

$$J_t^{\text{GN}}(\tau) = \sum_{k \in \mathcal{K}} \left| G_t^{(k)}(\tau) - \left( \frac{1}{|\mathcal{K}|} \sum_{\kappa \in \mathcal{K}} G_t^{(\kappa)}(\tau) \right) \cdot \left( r_t^{(k)}(\tau) \right)^{\alpha} \right|. \quad (6)$$

It depends on the magnitude  $G_t^{(k)}(\tau) = \|\lambda^{(k)}(\tau) \mathbf{g}_t^{(k)}(\tau)\|_2$  of the task-specific gradients  $\mathbf{g}_t^{(k)}(\tau)$  in the last shared layer and the task-specific training rates  $r_t^{(k)}(\tau) = \tilde{J}_t^{(k)}(\tau) \cdot \left( \frac{1}{|\mathcal{K}|} \sum_{\kappa \in \mathcal{K}} \tilde{J}_t^{(\kappa)}(\tau) \right)^{-1}$  with  $\tilde{J}_t^{(k)}(\tau) = J_t^{(k)}(\tau) \cdot (J_t^{(k)}(0))^{-1}$ , depending on the value of the task-specific losses  $J_t^{(k)}(\tau)$  taken from (1) and (2) at each step  $\tau$  in comparison to their values  $J_t^{(k)}(0)$  at step  $\tau = 0$ . These training rates can be interpreted as the convergence progress of the single tasks. Note that through the loss for the scaling factors in (6) a similar and stable convergence speed of both tasks is encouraged, which is essential for a successful multi-task training. The network is optimized with alternating updates of the scaling factors  $\lambda^{(k)}(\tau)$  by (6) and the network weights by (1) and (2). Although a more stable convergence is generally

expected, one can still optimize GradNorm (GN) with the hyperparameter  $\alpha$  in (6), controlling the restoring force back to balanced training rates. Also, after each step, the task weights are renormalized such that  $\lambda^{\text{seg}}(\tau) + \lambda^{\text{depth}}(\tau) = 1$ .

## 4 Evaluation Setup

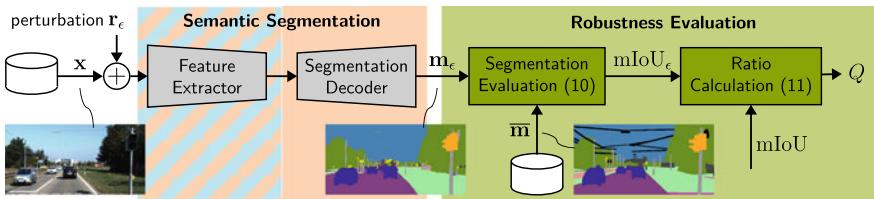
In this section, we present the input perturbations used to evaluate the model robustness, define the perturbation strength, and detail our chosen evaluation metrics. Finally, we provide an overview of our chosen databases and the implementation details of our method. The evaluation setup is depicted in Fig. 3.

**Image perturbations:** After the semantic segmentation network has been trained (either by single-task or multi-task training), we evaluate its robustness during inference by adding a perturbation  $\mathbf{r}_\epsilon$  of strength  $\epsilon$  to each input image  $\mathbf{x}$ , yielding a perturbed input image  $\mathbf{x}_\epsilon = \mathbf{x} + \mathbf{r}_\epsilon$ . For simplicity, we omit the subscript  $t$  as the evaluation is carried out on single images. We conduct experiments using two noise types and two adversarial attacks, for which we aim at imposing a perturbation strength  $\epsilon$  to make different perturbation types comparable. We measure this comparable strength by the signal-to-noise ratio (SNR) on the input image  $\mathbf{x}^{\text{adv}}$ , which is defined as

$$\text{SNR} = \frac{\mathbb{E} (\|\mathbf{x}\|_2^2)}{\mathbb{E} (\|\mathbf{r}_\epsilon\|_2^2)}. \quad (7)$$

Here,  $\mathbb{E} (\|\mathbf{x}\|_2^2)$  is the expectation value of the sum of the image's squared gray values, and  $\mathbb{E} (\|\mathbf{r}_\epsilon\|_2^2)$  is the expectation value of the sum of the squared noise pixels. As  $\mathbb{E} (\|\mathbf{x}\|_2^2)$  is always equal for different perturbation types, we define the perturbation's strength in dependency of  $\mathbb{E} (\|\mathbf{r}_\epsilon\|_2^2)$  as

$$\epsilon = \sqrt{\frac{1}{HWC} \mathbb{E} (\|\mathbf{r}_\epsilon\|_2^2)}. \quad (8)$$



**Fig. 3** Evaluation setup using additive perturbations to evaluate the robustness of a segmentation DNN. As perturbations, we use various noise and attack types to simulate deployment conditions

We consider Gaussian noise, salt and pepper noise, the fast gradient sign method (FGSM) attack [GSS15], and the projected gradient descent (PGD) attack [KGB17]. For Gaussian noise, the perturbation strength can be identified as the standard deviation of the underlying Gaussian distribution. For salt and pepper noise, on the other hand, some pixels are randomly set to 0 or 255. Therefore, first the input is perturbed, then the perturbation is obtained by  $\mathbf{r}_\epsilon = \mathbf{x}_\epsilon - \mathbf{x}$ , and finally the perturbation strength is computed by (8).

For the FGSM adversarial attack, the perturbation is calculated according to

$$\mathbf{x}_\epsilon = \mathbf{x} + \epsilon \cdot \mathbf{sign}(\nabla_{\mathbf{x}} J^{\text{ce}}(\bar{\mathbf{y}}, \mathbf{y}(\mathbf{x}))), \quad (9)$$

where  $\nabla_{\mathbf{x}}$  represents the derivative of the loss function with respect to the unperturbed image  $\mathbf{x}$ , and  $\mathbf{sign}(\cdot)$  represents the signum function applied to each element of its vectorial argument. As all elements  $r_{\epsilon,j}$  of the perturbation  $\mathbf{r}_\epsilon$  can only take on values  $r_{\epsilon,j} = \pm\epsilon$ ,  $j \in \{1, \dots, H \cdot W \cdot C\}$ , the perturbation variance is equal to  $\epsilon^2$  when applying (8). We also consider the PGD adversarial attack, due to its more advanced attack design, which is optimized over several optimization steps. This attack can be interpreted as an iterative version of (9) and allows investigations of the network robustness w.r.t. stronger adversarial attacks.

**Evaluation metrics:** To evaluate DNN performance on a perturbed input image  $\mathbf{x}_\epsilon$ , we pass this image to the DNN and generate the output  $\mathbf{m}_\epsilon$ , see Fig. 3. The output quality for a perturbation with  $\epsilon > 0$  is typically worse than the output generated from clean images  $\mathbf{x}_{\epsilon=0}$ . The absolute segmentation performance can then be obtained by calculating the mean intersection-over-union metric [EVGW+15] as

$$\text{mIoU}_\epsilon = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{\text{TP}_{\epsilon,s}}{\text{TP}_{\epsilon,s} + \text{FP}_{\epsilon,s} + \text{FN}_{\epsilon,s}}, \quad (10)$$

where the number of true positives  $\text{TP}_{\epsilon,s}$ , false positives  $\text{FP}_{\epsilon,s}$ , and false negatives  $\text{FN}_{\epsilon,s}$  for each class  $s$  are calculated over the entire evaluation subset before the application of (10).

As different models may have a different absolute prediction performance  $\text{mIoU}_{\epsilon=0}$  on clean images, a better performance on perturbed images can either mean that the model is better in general or more robust. Therefore, we rather compare the mIoU ratio

$$Q = \frac{\text{mIoU}_\epsilon}{\text{mIoU}_{\epsilon=0}} \quad (11)$$

obtained from the performance  $\text{mIoU}_\epsilon$  on perturbed images with a perturbation of strength  $\epsilon$ , normalized by the performance on clean images.

**Databases:** The additional training with the auxiliary task of depth estimation in a self-supervised fashion is applied jointly with the training of the primary semantic segmentation task. To accomplish this, for training, we always rely on an unlabeled image sequence dataset (bottom part of Table 1(a)) to train the depth estimation and

**Table 1** (a) Overview on the employed datasets with their respective subsets; (b) the employed image perturbations used to evaluate the robustness of the DNN (right-hand side)

(a) Databases; number of images				
Dataset	Symbol	Training	Validation	Test
Cityscapes [COR+16]	$\mathcal{X}^{\text{CS}}$	2,975	500	1525
KITTI [GLSU13, MG15]	$\mathcal{X}^{\text{KIT}}$	28,937	200	200
Cityscapes (seq.) [COR+16]	$\mathcal{X}^{\text{CS,seq}}$	83,300	–	–

(b) Perturbations				
Perturbation	Type			
Gaussian noise	Random noise			
Salt and pepper noise	Random noise			
FGSM [GSS15]	Adversarial attack			
PGD [KGB17]	Adversarial attack			

one labeled image dataset (top part of Table 1(a)) to train the semantic segmentation. Both datasets can be from different domains, as [KTMFs20] showed that a successful training for both tasks is also possible then. For segmentation training, we use Cityscapes [COR+16] ( $\mathcal{X}_{\text{train}}^{\text{CS}}$ ), while for depth estimation training we either use Cityscapes [COR+16] ( $\mathcal{X}_{\text{train}}^{\text{CS,seq}}$ ) or KITTI [GLSU13] ( $\mathcal{X}_{\text{train}}^{\text{KIT}}$ ), with the KITTI training split defined by [GMAB17]. Note that each image of the Cityscapes segmentation dataset contains 19 preceding and 10 succeeding unlabeled image frames, which we use for the depth estimation training. The number of training images for the depth estimation training splits deviates slightly from the original definitions due to the need of a preceding and a succeeding frame. For evaluation, we use the validation set from Cityscapes ( $\mathcal{X}_{\text{val}}^{\text{CS}}$ ), as the test sets are not publicly available. We also evaluate on the training set from the KITTI 2015 Stereo dataset [MG15] ( $\mathcal{X}_{\text{val}}^{\text{KIT}}$ ), which is disjoint from the training split as outlined in [GMAB17].

**Implementation details:** All models as well as training and evaluation protocols are implemented in PyTorch [PGM+19] and executed on an NVIDIA Tesla P100 graphics card. Same as [KTMFs20], we choose an encoder-decoder multi-task network architecture based on a ResNet18 feature extractor [HZRS16] pre-trained on ImageNet [RDS+15] and two task-specific decoder heads. These heads have an identical architecture except for the last layer: For depth estimation we employ a sigmoid output  $\sigma \in \mathbb{I}^{H \times W}$ , which is converted to depth values in a pixel-wise fashion as  $\frac{1}{a\sigma_i + b}$ , where  $a$  and  $b$  define the depth to the considered range [0.1, 100]. The segmentation output logits are comprised of  $S = |\mathcal{S}|$  feature maps which are converted to class probabilities via a softmax function. The pose network, which is required to train the depth estimation in a self-supervised fashion on videos, utilizes the same network architecture as in [GMAFB19].

We resize all images used for depth estimation training to a resolution of  $640 \times 192$ , while the images used for segmentation training are resized to  $512 \times 1024$  and cropped to the same resolution. We train the network for 40 epochs using the Adam optimizer [KB15] with a learning rate of  $10^{-4}$ , which is reduced to  $10^{-5}$  for the last 10 epochs. To ensure fairness, we use batch sizes of 12 and 6 for the single-task model and the two tasks of the multi-task model, respectively. Moreover, gradient scaling (5) is applied at all connections between the encoder and the decoder. For further training details the interested reader is referred to [KTMFs20].

## 5 Experimental Results and Discussion

In this section, we will first analyze the multi-task learning method w.r.t. the achieved absolute performance, where we will put the focus on how the performance can be improved without extensive hyperparameter tuning. Afterwards, the resulting models will be analyzed w.r.t. robustness against common image corruptions such as random noise or adversarial attacks.

**Absolute performance:** While the main focus of this chapter is the improved robustness of semantic segmentation DNNs, this robustness improvement should not come at the price of a significantly decreased absolute performance. In [KBFs20] it was proposed to scale the gradients using a manually tunable hyperparameter (GW). However, the results from Table 2 show that the absolute performance can decrease significantly, e.g., for GW ( $\lambda = 0.9$ ) on the KITTI dataset (right column in both tables), compared to the single-task baseline. As there is no general way to know which task weighting is optimal, we propose to use the GradNorm (GN) technique [CBLR18] instead of manual gradient weighting (GW). For this technique we observe that for all considered GN hyperparameters  $\alpha$  and on both dataset settings, the GradNorm technique improves the absolute performance over the single-task baseline. Interestingly, for this task weighting strategy, the task weights change over the course of the learning process. In particular, side experiments showed that the final task weights at the end of the training process do yield a decreased performance, if they are used constantly throughout the whole training process. This shows the importance of adapting them in dependence of the task-specific training progresses. Still, optimal performance is sometimes rather reached with manual gradient weighting (e.g., Table 2a), however, for robustness purposes an optimal absolute performance is not as important as a stable convergence of the multi-task training. We therefore use the GradNorm technique (GN) instead of the manual gradient weighting for all following experiments w.r.t. DNN robustness.

**Robustness to input noise:** As a simple next experiment, we compared the robustness w.r.t. Gaussian input noise between the baseline trained in a single-task fashion and models trained in a multi-task fashion with the GradNorm method. The results in Table 3a are obtained for a setting, where the segmentation and depth tasks are trained

**Table 2** Absolute segmentation performance measured by the mIoU [%] metric for models where the segmentation is trained on Cityscapes ( $\mathcal{X}_{\text{train}}^{\text{CS}}$ ) and the auxiliary depth estimation is either trained on KITTI ( $\mathcal{X}_{\text{train}}^{\text{KIT}}$ , top) or Cityscapes ( $\mathcal{X}_{\text{train}}^{\text{CS}, \text{seq}}$ , bottom). We report numbers on the Cityscapes ( $\mathcal{X}_{\text{val}}^{\text{CS}}$ ) and KITTI ( $\mathcal{X}_{\text{val}}^{\text{KIT}}$ ) validation sets for manual gradient weighting (GW) and the GradNorm (GN) multi-task training method. Best numbers are in boldface. Second best numbers are underlined.

(a) Segmentation: $\mathcal{X}_{\text{train}}^{\text{CS}}$ , depth: $\mathcal{X}_{\text{train}}^{\text{KIT}}$		
Method	mIoU on $\mathcal{X}_{\text{val}}^{\text{CS}}$	mIoU on $\mathcal{X}_{\text{val}}^{\text{KIT}}$
Baseline	63.5	43.0
GW ( $\lambda = 0.1$ )	<u>67.4</u>	<b>49.6</b>
GW ( $\lambda = 0.2$ )	<b>68.9</b>	47.7
GW ( $\lambda = 0.5$ )	<u>67.4</u>	34.7
GW ( $\lambda = 0.9$ )	67.7	29.8
GN ( $\alpha = 0.2$ )	66.8	44.0
GN ( $\alpha = 0.5$ )	67.0	47.0
GN ( $\alpha = 1.0$ )	66.4	<u>48.5</u>
GN ( $\alpha = 2.0$ )	65.7	45.6
(b) Segmentation: $\mathcal{X}_{\text{train}}^{\text{CS}}$ , depth: $\mathcal{X}_{\text{train}}^{\text{CS}, \text{seq}}$		
Method	mIoU on $\mathcal{X}_{\text{val}}^{\text{CS}}$	mIoU on $\mathcal{X}_{\text{val}}^{\text{KIT}}$
Baseline	63.5	43.0
GW ( $\lambda = 0.1$ )	65.8	<u>47.2</u>
GW ( $\lambda = 0.2$ )	66.6	46.0
GW ( $\lambda = 0.5$ )	65.1	44.9
GW ( $\lambda = 0.9$ )	66.1	40.5
GN ( $\alpha = 0.2$ )	66.1	46.1
GN ( $\alpha = 0.5$ )	<b>67.9</b>	<b>48.3</b>
GN ( $\alpha = 1.0$ )	<u>67.1</u>	45.5
GN ( $\alpha = 2.0$ )	66.5	<b>48.3</b>

on Cityscapes and KITTI, respectively. We clearly observe that all multi-task models (GN variants) exhibit a higher robustness measured by the mIoU ratio  $Q$  (11), e.g.,  $Q = 27.5\%$  to  $Q = 33.3\%$  for a perturbation strength of  $\epsilon = 16$ , compared to the single task baseline ( $Q = 18.4\%$ ). The model variant GN ( $\alpha = 0.5$ ) is furthermore shown in Fig. 4. When looking at the curves for the Cityscapes and KITTI validation sets and for both Gaussian and salt and pepper noise, we observe that the multi-task model is consistently either on par or better w.r.t. robustness, measured by the mIoU ratio  $Q$ .

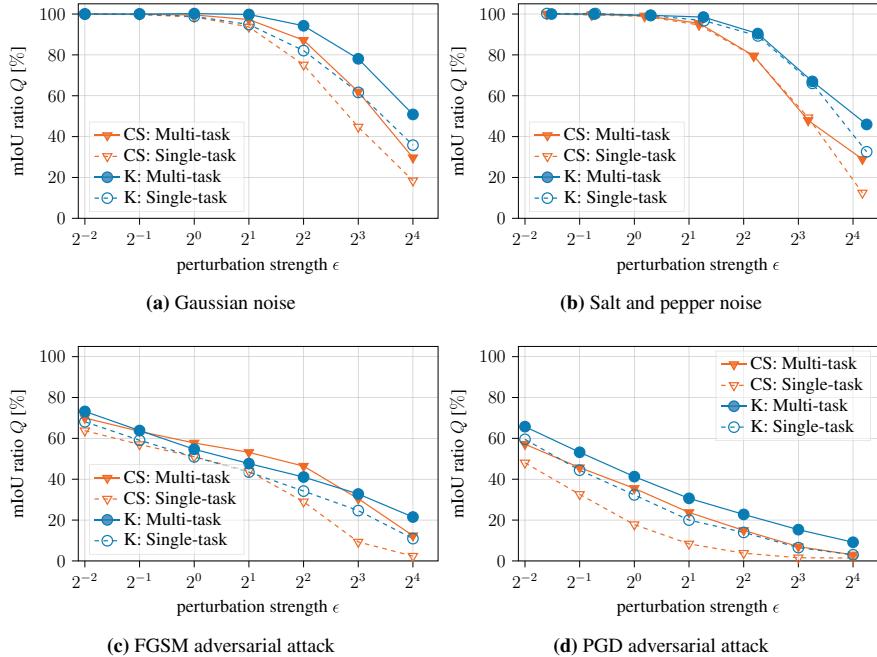
A slightly different behavior is observed when the auxiliary depth estimation task is trained on the same dataset as the semantic segmentation task, as shown in Table 3b. Here, for small perturbation strengths ( $\epsilon \leq 4$ ), the robustness is still improved, however, for larger perturbation strengths, the robustness is either similar or even decreased. This can be interpreted as an indication that the additional data

**Table 3** Robustness to Gaussian input noise measured by the mIoU ratio  $Q$  [%] (11) for models where the segmentation is trained on Cityscapes ( $\mathcal{X}_{\text{train}}^{\text{CS}}$ ) and the auxiliary depth estimation is either trained on KITTI ( $\mathcal{X}_{\text{train}}^{\text{KIT}}$ , top) or Cityscapes ( $\mathcal{X}_{\text{train}}^{\text{CS}, \text{seq}}$ , bottom). Best numbers are in boldface

(a) Segmentation: $\mathcal{X}_{\text{train}}^{\text{CS}}$ , depth: $\mathcal{X}_{\text{train}}^{\text{KIT}}$							
$\epsilon$	0.25	0.5	1	2	4	8	16
Baseline	<b>100.0</b>	99.7	98.6	93.7	75.1	44.7	18.4
GN ( $\alpha = 0.2$ )	<b>100.0</b>	99.9	99.0	96.1	85.5	60.8	28.1
GN ( $\alpha = 0.5$ )	<b>100.0</b>	99.9	<b>99.6</b>	<b>97.3</b>	87.2	61.9	29.7
GN ( $\alpha = 1.0$ )	<b>100.0</b>	<b>100.0</b>	99.5	97.1	<b>88.1</b>	<b>65.4</b>	<b>33.3</b>
GN ( $\alpha = 2.0$ )	<b>100.0</b>	99.8	99.2	96.2	86.2	60.7	27.5
(b) Segmentation: $\mathcal{X}_{\text{train}}^{\text{CS}}$ , depth: $\mathcal{X}_{\text{train}}^{\text{CS}, \text{seq}}$							
$\epsilon$	0.25	0.5	1	2	4	8	16
Baseline	<b>100.0</b>	99.7	98.6	93.7	75.1	44.7	<b>18.4</b>
GN ( $\alpha = 0.2$ )	<b>100.0</b>	99.8	<b>99.4</b>	<b>96.8</b>	82.0	<b>45.4</b>	13.0
GN ( $\alpha = 0.5$ )	99.9	99.7	99.1	95.3	79.4	41.8	10.2
GN ( $\alpha = 1.0$ )	<b>100.0</b>	<b>99.9</b>	<b>99.4</b>	96.3	<b>82.4</b>	42.5	15.6
GN ( $\alpha = 2.0$ )	99.8	99.7	99.1	96.2	82.0	44.5	8.0

from another domain is mainly responsible for the robustness improvement rather than the additional task itself. However, the self-supervised training technique of the auxiliary task is the precondition for being able to make use of additional unlabeled data.

**Robustness to adversarial attacks:** In addition to simple noise conditions, we also investigate adversarial attacks, where the noise pattern is optimized to fool the network. In Table 4 we show results on robustness w.r.t. the FGSM adversarial attack. We again observe that when the auxiliary task is trained in a different domain (left-hand side), the robustness is significantly improved regardless of the perturbation strength. In contrast to simple noise perturbations, the FGSM attack can degrade performance even for very small and visually hard-to-perceive perturbation strengths ( $\epsilon \leq 1$ ), for which robustness is still improved by our method. Moreover, we again observe that the robustness improvement is not as good, when the auxiliary depth task is trained in the same domain (right-hand side), as here the robustness improvement is not as high. For instance, for  $\epsilon = 8$  the robustness improves from  $Q = 9.6\%$  to  $Q = 33.4\%$  for the best multi-task model, when the depth is trained out-of-domain, while for in-



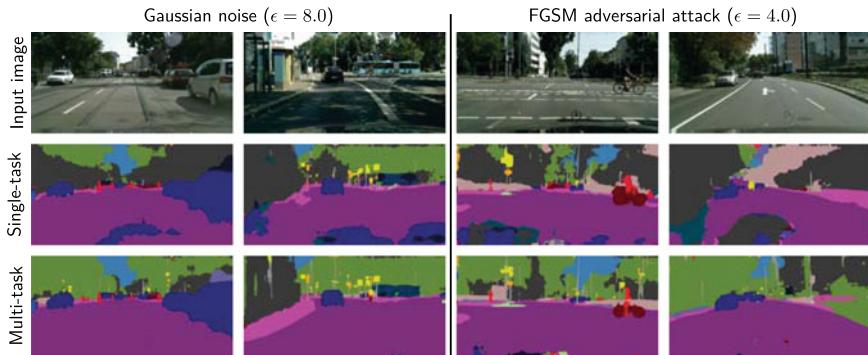
**Fig. 4** Robustness to several input perturbation types for models where the segmentation is trained on Cityscapes ( $X_{\text{train}}^{\text{CS}}$ ) and the auxiliary depth estimation on KITTI ( $X_{\text{train}}^{\text{KIT}}$ ). We report mIoU ratios  $Q$  [%] (11) on the Cityscapes ( $X_{\text{val}}^{\text{CS}}$ ) and KITTI ( $X_{\text{val}}^{\text{KIT}}$ ) validation sets for the GradNorm ( $\alpha = 0.5$ ) multi-task training method

domain training the improvement is only from  $Q = 9.6\%$  to  $Q = 21.8\%$ . Still, for the FGSM attack all multi-task models improve upon the baseline in terms of robustness, showing the general applicability of the GradNorm multi-task learning technique for robustness improvement.

To also investigate this effect on a wider range of datasets and perturbations, we show robustness results on the Cityscapes and KITTI validation sets and for the FGSM and PGD adversarial attack in Fig. 4, bottom. For all considered cases, the robustness of the multi-task model GN ( $\alpha = 0.5$ ) improves upon the single-task baseline. We also show qualitative results in Fig. 5 for Gaussian noise ( $\epsilon = 8.0$ ) and the FGSM attack ( $\epsilon = 4.0$ ) for the GN model ( $\alpha = 0.5$ ), where we also qualitatively observe a significant improvement over the single-task baseline.

**Table 4** Robustness to the FGSM adversarial attack measured by the mIoU ratio  $Q$  [%] (11) for models where the segmentation is trained on Cityscapes ( $\mathcal{X}_{\text{train}}^{\text{CS}}$ ) and the auxiliary depth estimation is either trained on KITTI ( $\mathcal{X}_{\text{train}}^{\text{KIT}}$ , top) or Cityscapes ( $\mathcal{X}_{\text{train}}^{\text{CS}, \text{seq}}$ , bottom). Best numbers are in boldface

(a) Segmentation: $\mathcal{X}_{\text{train}}^{\text{CS}}$ , depth: $\mathcal{X}_{\text{train}}^{\text{KIT}}$							
$\epsilon$	0.25	0.5	1	2	4	8	16
Baseline	63.8	57.0	51.0	43.9	29.6	9.6	2.4
GN ( $\alpha = 0.2$ )	<b>73.0</b>	<b>67.2</b>	<b>62.3</b>	<b>58.7</b>	<b>51.2</b>	32.6	14.4
GN ( $\alpha = 0.5$ )	70.0	63.4	57.8	53.1	46.4	30.4	12.4
GN ( $\alpha = 1.0$ )	70.3	63.9	59.3	55.8	49.2	<b>33.4</b>	13.3
GN ( $\alpha = 2.0$ )	72.1	66.1	61.5	57.5	44.5	31.1	<b>14.6</b>
(b) Segmentation: $\mathcal{X}_{\text{train}}^{\text{CS}}$ , depth: $\mathcal{X}_{\text{train}}^{\text{CS}, \text{seq}}$							
$\epsilon$	0.25	0.5	1	2	4	8	16
Baseline	63.8	57.0	51.0	43.9	29.6	9.6	2.4
GN ( $\alpha = 0.2$ )	71.4	65.7	61.0	<b>57.5</b>	<b>46.9</b>	<b>21.8</b>	<b>7.7</b>
GN ( $\alpha = 0.5$ )	<b>74.3</b>	66.0	61.1	56.7	44.5	17.7	5.6
GN ( $\alpha = 1.0$ )	72.6	<b>66.3</b>	<b>61.5</b>	57.2	40.8	16.4	<b>7.7</b>
GN ( $\alpha = 2.0$ )	72.2	66.0	<b>61.5</b>	57.1	44.5	16.2	2.9



**Fig. 5** Qualitative result comparison for models where the segmentation is trained on Cityscapes ( $\mathcal{X}_{\text{train}}^{\text{CS}}$ ) and the auxiliary depth estimation on KITTI ( $\mathcal{X}_{\text{train}}^{\text{KIT}}$ ). We show results for two different perturbations and compare between the single-task baseline and the GradNorm ( $\alpha = 0.5$ ) multi-task training method

## 6 Conclusions

In this chapter, we show how the robustness of a semantic segmentation DNN can be improved by multi-task training with the auxiliary task of depth estimation, which can be trained in a self-supervised fashion on arbitrary videos. We show this robustness improvement across two noise perturbations and two adversarial attacks, where we ensure comparability of different perturbations in terms of strength. By making use of the GradNorm task weighting strategy, we are able to remove the necessity for manual tuning of hyperparameters, thereby achieving a stable and easy-to-apply robustness improvement. Also, we show that in-domain training with the additional task of depth estimation already improves robustness to some degree, while out-of-domain training on additional unlabeled data enabled by the self-supervised training improves robustness even further. Moreover, our method is easy-to-apply, induces no computational overhead during inference, and even improves absolute performance, which can be of interest for applications such as autonomous driving, virtual reality, or medical imaging as long as additional video data is available. Our method thereby demonstrates various further advantages of multi-task training for semantic segmentation, which could be potentially generalizable to various further computer vision tasks.

## References

- [ACW18] A. Athalye, N. Carlini, D. Wagner, Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden, July 2018, pp. 274–283
- [AMT18] A. Arnab, O. Miksik, P.H.S. Torr, On the robustness of semantic segmentation models to adversarial attacks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 888–897
- [ASG+19] F. Assion, P. Schlicht, F. Greßner, W. Günther, F. Hüger, N.M. Schmidt, U. Rasheed, The attack generator: a systematic approach towards constructing adversarial attacks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA, June 2019, pp. 1370–1379
- [ATT18] A.E. Aydemir, A. Temizel, T.T. Temizel, *The Effects of JPEG and JPEG2000 Compression on Attacks Using Adversarial Examples*, Mar. 2018, pp. 1–4. [arxiv:1803.10418](https://arxiv.org/abs/1803.10418)
- [BHSFs19] A. Bär, F. Hüger, P. Schlicht, T. Fingscheidt, On the robustness of redundant teacher-student frameworks for semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA, June 2019, pp. 1380–1388
- [Bis95] C.M. Bishop, Training with noise is equivalent to Tikhonov regularization. *Neural Comput.* 7(1), 108–116 (1995)

- [BKV+20] A. Bär, M. Klingner, S. Varghese, F. Hüger, P. Schlicht, T. Fingscheidt, Robust semantic segmentation by redundant networks with a layer-specific loss contribution and majority vote, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1348–1358, virtual conference, June 2020
- [BLK+21] A. Bär, J. Löhdorfink, N. Kapoor, S.J. Varghese, F. Hüger, P. Schlicht, T. Fingscheidt, The vulnerability of semantic segmentation networks to adversarial attacks in autonomous driving: enhancing extensive environment sensing. *IEEE Signal Process. Mag.* **38**(1), 42–52 (2021)
- [CBG+17] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, N. Usunier, Parseval networks: improving robustness to adversarial examples, in *Proceedings of the International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, Aug. 2017, pp. 854–863
- [CBLR18] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, GradNorm: gradient normalization for adaptive loss balancing in deep multitask networks, in *Proceedings of the International Conference on Machine Learning (ICML)*, Stockholm, Sweden, July 2018, pp. 794–803
- [CLC+19] H.-Y. Chen, J.-H. Liang, S.-C. Chang, J.-Y. Pan, Y.-T. Chen, W. Wei, D.-C. Juan, Improving adversarial robustness via guided complement entropy, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea, Oct. 2019, pp. 4881–4889
- [CLC+20] T. Chen, S. Liu, S. Chang, Y. Cheng, L. Amini, Z. Wang, Adversarial robustness: from self-supervised pre-training to fine-tuning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference, June 2020, pp. 699–708
- [CLLW19] P.-Y. Chen, A.H. Liu, Y.-C. Liu, Y.-C.F. Wang, Towards scene understanding: unsupervised monocular depth estimation with semantic-aware representation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 2624–2632
- [COR+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 3213–3223
- [CPK+18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **40**(4), 834–848 (2018)
- [CPMA19] V. Casser, S. Pirk, R. Mahjourian, A. Angelova, Depth videos, in *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, HI, USA, Jan. 2019, pp. 8001–8008
- [CW17] N. Carlini, D.A. Wagner, Towards evaluating the robustness of neural networks, in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 39–57
- [DLP+18] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 9185–9193
- [EF15] D. Eigen, R. Fergus, Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2650–2658
- [EPF14] D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, in *Proceedings of the Conference on Neural Information*

- Processing Systems (NIPS/NeurIPS)*, Montréal, QC, Canada, Dec. 2014, pp. 2366–2374
- [EVGW+15] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: a retrospective. *Int. J. Comput. Vis. (IJCV)* **111**(1), 98–136 (2015)
- [GHL+20] V. Guizilini, R. Hou, J. Li, R. Ambrus, A. Gaidon, Semantically-Guided representation learning for self-supervised monocular depth, in *Proceedings of the International Conference on Learning Representations (ICLR)*, virtual conference, Apr. 2020, pp. 1–14
- [GL15] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, July 2015, pp. 1180–1189
- [GLSU13] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: the KITTI dataset. *Int. J. Robot. Res.* **32**(11), 1231–1237 (2013)
- [GMAB17] C. Godard, O.M. Aodha, G.J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017, pp. 270–279
- [GMAFB19] C. Godard, O.M. Aodha, M. Firman, G.J. Brostow, Digging into self-supervised monocular depth estimation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea, Oct. 2019, pp. 3828–3838
- [GRCvdM18] C. Guo, M. Rana, M. Cissé, L. van der Maaten, Countering adversarial images using input transformations, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, Apr. 2018, pp. 1–12
- [GSS15] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, pp. 1–11 (2015)
- [GW08] R.C. Gonzales, R.E. Woods, *Digital Image Processing* (Prentice Hall, 2008)
- [HAMFs22] A.S. Hashemi, B. Andreas, S. Mozaffari, T. Fingscheidt, Improving transferability of generated universal adversarial perturbations for image classification and segmentation, in *Deep Neural Networks and Data for Automated Driving—Robustness, Uncertainty Quantification, and Insights Towards Safety*, ed. by T. Fingscheidt, H. Gottschalk, S. Houben, (Springer, 2022), pp. 195–222
- [HK92] L. Holström, P. Koistinen, Using additive noise in backpropagation-training. *IEEE Trans. Neural Netw. (TNN)* **3**(1), 24–38 (1992). (January)
- [HMKS19] D. Hendrycks, M. Mazeika, S. Kadavath, D. Song, Using self-supervised learning can improve model robustness and uncertainty, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada, Dec. 2019, pp. 15637–15648
- [HZRS16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 770–778
- [KB15] D.P. Kingma, J. Ba, ADAM: a method for stochastic optimization, in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15
- [KBFs20] M. Klingner, A. Bär, T. Fingscheidt, Improved noise and attack robustness for semantic segmentation by using multi-task training with self-supervised depth estimation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, virtual conference, June 2020, pp. 1299–1309
- [KBV+21] N. Kapoor, A. Bär, S. Varghese, J.D. Schneider, F. Hüger, P. Schlicht, T. Fingscheidt, From a Fourier-domain perspective on adversarial examples to a Wiener filter defense for semantic segmentation, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, virtual conference, July 2021, pp. 1–8

- [KGB17] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, Toulon, France, Apr. 2017, pp. 1–14
- [KGC18] A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 7482–7491
- [KTMFs20] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, T. Fingscheidt, Self-Supervised monocular depth estimation: solving the dynamic object problem by semantic guidance, in *Proceedings of the European Conference on Computer Vision (ECCV)*, virtual conference, Aug. 2020, pp. 582–600
- [LJL+19] H. Liu, R. Ji, J. Li, B. Zhang, Y. Gao, Y. Wu, F. Huang, Universal adversarial perturbation via prior driven uncertainty approximation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seoul, Korea, Oct. 2019, pp. 2941–2949
- [LLKX19] P. Liu, M. Lyu, I. King, J. Xu, SelFlow: self-supervised learning of optical flow, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 4571–4580
- [LLL+19] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, W. Wen, Feature distillation: DNN-Oriented JPEG compression against adversarial examples, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 860–868
- [LRLG19] K.-H. Lee, G. Ros, J. Li, A. Gaidon, SPIGAN: privileged adversarial learning from simulation, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, Apr. 2019, pp. 1–14
- [LSD15] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015, pp. 3431–3440
- [MCKBF17] J.H. Metzen, M.C. Kumar, T. Brox, V. Fischer, Universal adversarial perturbations against semantic image segmentation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2774–2783
- [MDFFF17] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017, pp. 1765–1773
- [MG15] M. Menze, A. Geiger, Object scene flow for autonomous vehicles, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015, pp. 3061–3070
- [MGB17] K.R. Mopuri, U. Garg, R. Venkatesh Babu, Fast feature fool: a data independent approach to universal adversarial perturbations, in *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK, Sept. 2017, pp. 1–12
- [MGR19] K.R. Mopuri, A. Ganeshan, V.B. Radhakrishnan, Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **41**(10), 2452–2465 (2019)
- [MLR+19] Y. Meng, Y. Lu, A. Raj, S. Sunarjo, R. Guo, T. Javidi, G. Bansal, D. Bharadia, SIGNet: semantic instance aided unsupervised 3D geometry perception, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 9810–9820
- [MMS+18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, Apr. 2018, pp. 1–10
- [NOBK17] G. Neuhold, T. Ollmann, S.R. Bulò, P. Kortschieder, The mapillary vistas dataset for semantic understanding of street scenes, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 4990–4999

- [NVA19] J. Novosel, P. Viswanath, B. Arsenali, Boosting semantic segmentation with multi-task self-supervised learning for autonomous driving applications, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS) Workshops*, Vancouver, BC, Canada, Dec. 2019, pp. 1–11
- [PCKC16] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, *ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation*, pp. 1–10, June 2016. [arxiv:1606.02147](https://arxiv.org/abs/1606.02147)
- [PGM+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison et al., PyTorch: an imperative style, high-performance deep learning library, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Vancouver, BC, Canada, Dec. 2019, pp. 8024–8035
- [PMG+17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z.B. Celik, A. Swami, Practical black-box attacks against machine learning, in *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACSS)*, Abu Dhabi, UAE, Apr. 2017, pp. 506–519
- [RDS+15] O. Russakovsky, J. Deng, S. Hao, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis. (IJCV)* **115**(3), 211–252 (2015)
- [RKKY+21a] V.R. Kumar, M. Klingner, S. Yogamani, M. Bach, S. Milz, T. Fingscheidt, P. Mäder, SVDistNet: self-supervised near-field distance estimation on surround view fisheye cameras. *IEEE Trans. Intell. Transp. Syst. (TITS)* 1–10, June 2021, early access
- [RKKY+21b] V.R. Kumar, M. Klingner, S. Yogamani, S. Milz, T. Fingscheidt, P. Mäder, SynDistNet: self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving, in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, virtual conference, Jan. 2021, pp. 61–71
- [RSFM19] E. Raff, J. Sylvester, S. Forsyth, M. McLean, Barrage of random transforms for adversarially robust defense, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 6528–6537
- [Sze10] R. Szeliski. *Computer Vision: Algorithms and Applications* (Springer, 2010)
- [SZS+14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, AB, Canada, Dec. 2014, pp. 1–10
- [VJB+19] T.-H. Vu, H. Jain, M. Bucher, M. Cord, P. Perez, ADVENT: adversarial entropy minimization for domain adaptation in semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 2517–2526
- [XOWS18] D. Xu, W. Ouyang, X. Wang, N. Sebe, PAD-Net: multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 675–684
- [XWZ+18] C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, Mitigating adversarial effects through randomization, in *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 1–15, Vancouver, BC, Canada, Apr. 2018
- [YS18] Z. Yin, J. Shi, GeoNet: unsupervised learning of dense depth, optical flow and camera pose, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 1983–1992
- [YZS+18] G. Yang, H. Zhao, J. Shi, Z. Deng, J. Jia, SegStereo: exploiting semantic information for disparity estimation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, Sept. 2018, pp. 636–651

- [ZBL20] S. Zhu, G. Brazil, X. Liu, The edge of depth: explicit constraints between segmentation and depth, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference, June 2020, pp. 13116–13125
- [ZBSL17] T. Zhou, M. Brown, N. Snavely, D.G. Lowe, Unsupervised learning of depth and Ego-Motion from video, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017, pp. 1851–1860
- [ZCX+19] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, J. Yang, Pattern-Affinitive propagation across depth, surface normal and semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 4106–4115
- [ZYC+20] L. Zhang, M. Yu, T. Chen, Z. Shi, C. Bao, K. Ma, Auxiliary training: towards accurate and robust models, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference, June 2020, pp. 372–381

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



# Improving Transferability of Generated Universal Adversarial Perturbations for Image Classification and Segmentation



Atiye Sadat Hashemi, Andreas Bär, Saeed Mozaffari, and Tim Fingscheidt

**Abstract** Although deep neural networks (DNNs) are high-performance methods for various complex tasks, e.g., environment perception in automated vehicles (AVs), they are vulnerable to adversarial perturbations. Recent works have proven the existence of universal adversarial perturbations (UAPs), which, when added to most images, destroy the output of the respective perception function. Existing attack methods often show a low success rate when attacking target models which are different from the one that the attack was optimized on. To address such weak transferability, we propose a novel learning criterion by combining a low-level feature loss, addressing the similarity of feature representations in the first layer of various model architectures, with a cross-entropy loss. Experimental results on ImageNet and Cityscapes datasets show that our method effectively generates universal adversarial perturbations achieving state-of-the-art fooling rates across different models, tasks, and datasets. Due to their effectiveness, we propose the use of such novel generated UAPs in robustness evaluation of DNN-based environment perception functions for AVs.

---

A. S. Hashemi · A. Bär (✉) · T. Fingscheidt

Institute for Communications Technology (IfN), Technische Universität Braunschweig,  
Schleinitzstr. 22, 38106 Braunschweig, Germany

e-mail: [andreas.baer@tu-bs.de](mailto:andreas.baer@tu-bs.de)

A. S. Hashemi

e-mail: [atiye.hashemi1991@gmail.com](mailto:atiye.hashemi1991@gmail.com); [atiye.hashemi@semnan.ac.ir](mailto:atiye.hashemi@semnan.ac.ir)

T. Fingscheidt

e-mail: [t.fingscheidt@tu-bs.de](mailto:t.fingscheidt@tu-bs.de)

A. S. Hashemi · S. Mozaffari

Semnan University, Campus 1, 35131-19111 Semnan, Iran

e-mail: [mozaffari@semnan.ac.ir](mailto:mozaffari@semnan.ac.ir); [saeed\\_mozaffari@yahoo.com](mailto:saeed_mozaffari@yahoo.com)

S. Mozaffari

University of Windsor, 401 Sunset Ave, Windsor, ON N9B 3P4, Canada

## 1 Introduction

Reaching desired safety standards quickly is of utmost importance for automated vehicles (AVs), in particular, for their environment perception module. Recently, growing advancements in deep neural networks (DNNs) gave the means to researchers for solving real-world problems, specifically improving the state of the art in environment perception [GTCM20, NVM+19]. These networks can help AVs in understanding the environment, such as identifying traffic signs and detecting surrounding objects, by incorporating many sensors (e.g., camera, LiDAR, and RaDAR) to build an overall representation of the environment [WLH+20, FJGN20], or even to provide an end-to-end control for the vehicle [KBJ+20], see Fig. 1.

A large body of studies has addressed adversarial attacks [SZS+14, MDFF+18] and robustness enhancement [GRM+19, SOZ+18] of DNN architectures. In AVs, it was shown that adversarial signs could fool a commercial classification system in real-world driving conditions [MKMW19]. In addition to the vulnerability of image classifiers, Arnab et al. [AMT18] extensively analyzed the behavior of semantic segmentation architectures for AVs and illustrated their susceptibility to adversarial attacks. To reach the high level of automation, defined by the SAE standard J3016 [SAE18], car manufacturers have to consider various threats and perturbations targeting the AV systems.

In vision-related tasks, adversarial perturbations can be divided into two types: image-dependent (per-instance) adversarial perturbations and universal adversarial perturbations (UAPs). In image-dependent adversarial perturbations, a specific optimization has to be performed for each image individually to generate an adversarial example. In contrast, UAPs are more general perturbations in the sense that an additive single perturbation triggers all the images in a dataset to become adversarial examples. Therefore, they are much more efficient in terms of computation cost and time when compared to image-dependent adversarial attacks [CAB+20]. However, while showing a high success rate on the models they are optimized on, they lack transferability to other models.

In this chapter, we aim to specifically address the vulnerability of image classification and semantic segmentation systems as cornerstones of visual perception in automated vehicles. In particular, we aim at improving the transferability of task-specific universal adversarial perturbations. Our major contributions are as follows:

First, we present a comprehensive similarity analysis of features from several layers of different DNN classifiers.

Second, based on these findings, we propose a novel fooling loss function for generating universal adversarial perturbations. In particular, we combine the fast feature fool loss [MGR19], however, focusing on the first layer only, with the cross-entropy loss, to train an attack generator with the help of a source model for generating targeted and non-targeted UAPs for any other target model.

Third, we show that our UAPs not only exhibit remarkable transferability across multiple networks trained on the same dataset, but also they can be generated using a

reduced subset of the training dataset, while still having a satisfactory generalization power over unseen data.

Fourth, using our method, we are able to surpass state-of-the-art performance in both white-box and black-box settings.

Finally, by extensive evaluation of the generated UAPs on various image classification and semantic segmentation models, we demonstrate that our approach is generalizable across multiple vision tasks.

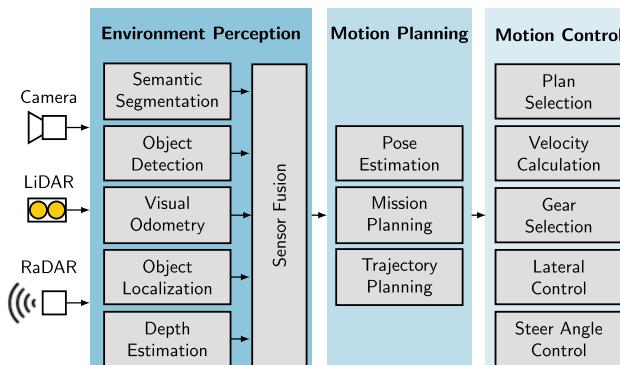
The remainder of this chapter is organized as follows. In Sect. 2, we present some background and related works. The proposed method, along with mathematical notations, is introduced in Sect. 3. Experimental results on image classification and semantic segmentation tasks are then presented in Sects. 4 and 5, respectively. Finally, in Sect. 6 we conclude the chapter.

## 2 Related Works

### 2.1 Environment Perception for Automated Vehicles

Figure 1 gives an overview of the major components comprised within an automated vehicle: Environment perception, motion planning, and motion control [GR20]. The environment perception module collects data using several sensors to obtain an overall representation of the surroundings. This information is then processed through the motion planning module to calculate a reasonable trajectory, which is executed via the motion control module at the end. In this chapter, we concentrate on the environment perception of automated vehicles.

The environment perception of automated vehicles comprises several sensors, including radio detection and ranging (RaDAR), light detection and ranging (LiDAR),



**Fig. 1** Environment perception and subsequent functions in an automated vehicle (AV), acc. to [GR20]

and camera [RT19]. RaDAR sensors supplement camera vision in times of low visibility, e.g., night driving, and are able to improve the detection accuracy [PTWA17]. LiDAR sensors are commonly used to make high-resolution maps, and they are capable of detecting obstacles [WXZ20]. The recent rise of deep neural networks puts a high interest to camera-based environment perception solutions, e.g., object detection and classification [ZWJW20], as well as semantic segmentation [RABA18].

In this chapter, we focus on camera-based environment perception with deep neural networks for image classification as well as for semantic segmentation in the context of AVs.

## 2.2 Adversarial Attacks

It is well known that deep neural networks are vulnerable to adversarial attacks [SZS+14]. While adversarial attacks for image classification have been widely studied, the influence of these attacks in other applications, such as semantic segmentation, has rarely been investigated [AM18, BHSFs19, BZIK20, BKV+20, KBFs20, ZBIK20a, BLK+21]. In this section, we review existing attack methods in both classification and semantic segmentation perception tasks.

There are several possible ways of categorizing an adversarial attack, e.g., targeted/non-targeted attack, and white-box/black-box attack [HM19]. In a targeted attack, the adversary generates an adversarial example to change the system's output to a maliciously chosen target label. In a non-targeted attack, on the other hand, the attacker wants to direct the result to a label that is different from the ground truth, no matter what it is. It should be noted that in semantic segmentation, targeted attacks can be divided into static and dynamic target segmentation [MCKBF17]. Static attacks are similar to the targeted attacks in image classification. Here, the aim is to enforce the model to always output the same target semantic segmentation mask. The dynamic attack follows the objective of replacing all pixels classified with a beforehand chosen target class by its spatial nearest neighbor class. Another way of attack categorization is based on the attackers' knowledge of the parameters and the architecture of the target model. While a white-box scenario refers to the case when an attacker has full knowledge about the underlying model, a black-box scenario implies that no information about the respective model is available. We address both targeted and non-targeted attacks as well as white-box and black-box scenarios in this chapter.

In the following, we will investigate two types of image-dependent and image-agnostic adversarial perturbations in classification and semantic segmentation models. We will also review the problem of transferability in adversarial attacks.

**Image-dependent adversarial perturbations:** There are various iterative, non-iterative, and generative methods for crafting image-dependent adversarial examples. Goodfellow et al. [GSS15] introduced the fast gradient sign method (FGSM), one of the first adversarial attacks. FGSM aims at computing the gradients of the

source model  $S$  with respect to the (image) input  $\mathbf{x}$  and a loss  $J$  to create an adversarial example. Iterative FGSM (I-FGSM) [KGB17] iteratively applies FGSM with a small step size, while momentum iterative FGSM [DLP+18] utilizes a momentum-based optimization algorithm for stronger adversarial attacks. Another iterative attack is projected gradient descent (PGD) [MMS+18], where the main difference to I-FGSM is random restarts in the optimization process. Kurakin et al. [KGB17] proposed the least-likely class method (LLCM), in which the target is set to the least-likely class predicted by the network. Xiao et al. [XZL+18] introduced the spatial transform attack (STM) for generating adversarial examples. In STM, instead of changing the pixel values in a direct manner, spatial filters are employed to substitute pixel values maliciously. Also, some works [HM19] have employed a generative adversarial network (GAN) [GPAM+14] for generating adversarial examples.

A basic analysis on the behavior of semantic segmentation DNNs against adversarial examples was performed by Arnab et al. [AMT18]. They applied three commonly known adversarial attacks for image classification tasks, i.e., FGSM [GSS15], Iterative-FGSM [KGB17], and LLCM [KGB17], to semantic segmentation models and illustrated the vulnerability of this task. There are also some works that concentrate on sophisticated adversarial attacks that lead to more reasonable outcomes in the context of semantic segmentation [XDL+18, PKGB18].

**Universal adversarial perturbations:** Universal adversarial perturbations (UAPs) were firstly introduced by Moosavi-Dezfooli et al. as image-agnostic perturbations [MDFFF17]. Similar to image-dependent adversarial attacks, there are some iterative, non-iterative, and generative techniques for creating UAPs. An iterative algorithm to generate UAPs for an image classifier was presented in [MDFFF18]. The authors provided an analytical analysis of the decision boundary in DNNs based on geometry and proved the existence of small universal adversarial perturbations. Some researchers focused on generative models that can be trained for generating UAPs [HD18, RMOGVB18]. Mopuri et al. presented a network for adversary generation (NAG) [RMOGVB18], which builds upon GANs. NAG utilizes fooling and diversity loss functions to model the distribution of UAPs for a DNN image classifier. Moreover, Poursaeed et al. [PKGB18] introduced the generative adversarial perturbation (GAP) algorithm for transforming noise drawn from a uniform distribution to adversarial perturbations to conduct adversarial attacks in classification and semantic segmentation tasks. Metzen et al. [MCKBF17] proposed an iterative algorithm for semantic segmentation, which led to more realistically looking false segmentation masks.

Contrary to the previous *data-dependent* methods, Mopuri et al. [MGR19] introduced fast feature fool (FFF), a *data-independent* algorithm for producing non-targeted UAPs. FFF aims at injecting maximal adversarial energy into each layer of the source model  $S$ . This is done by the following loss function:

$$J^{\text{FFF}}(\mathbf{r}) = \sum_{\ell=1}^L J_{\ell}^{\text{FFF}}(\mathbf{r}), \quad J_{\ell}^{\text{FFF}}(\mathbf{r}) = -\log(\|\mathbf{A}_{\ell}(\mathbf{r})\|_2), \quad (1)$$

where  $\mathbf{A}_\ell(\mathbf{r})$  is the mean of all feature maps of the  $\ell$ -th layer (after the activation function in layer  $\ell$ ), when *only* the UAP  $\mathbf{r}$  is fed into the model. Note that usually  $\mathbf{x}^{\text{adv}} = \mathbf{x} + \mathbf{r}$ , with clean image  $\mathbf{x}$ , is fed into the model. This algorithm starts with a random  $\mathbf{r}$  which is then iteratively optimized. For mitigating the absence of data in producing UAPs, Mopuri et al. [MUR18] proposed class impressions (CIs), a form of reconstructed images that are obtained via simple optimization from the source model. After finding multiple CIs in the input space for each target class, they trained a generator to create UAPs. Recently, Zhang et al. [ZBIK20b] proposed a targeted UAP algorithm using random source images (TUAP-RSI) from a proxy dataset instead of the original training dataset.

In this chapter, we follow Poursaeed et al. [PKGB18] by proposing an efficient generative approach that focuses on propagating adversarial energy in a source model to generate UAPs for the task of image classification and semantic segmentation.

**Transferability in black-box attacks:** The ability of an adversarial example to be effective against a different, potentially unknown, target model is known as transferability. Researchers have evaluated the transferability of adversarial examples on image classifiers [MGR19, MDFFF17, PXL+20, LBX+20] and semantic segmentation networks [PKGB18, AMT18].

Regarding the philosophy behind transferability, Goodfellow et al. [GSS15] demonstrated that estimating the size of adversarial subspaces is relevant to the transferability issue. Another potential perspective lies in the similarity of decision boundaries. Learning substitute models, approximating the decision boundaries of target models, is a famous approach to attack an unknown model [PMJ+16]. Wu et al. [WWX+20] considered DNNs with skip connections and found that using more gradients from the skip connections, rather than the residual modules, allows the attacker to craft more transferable adversarial examples. Wei et al. [WLCC18] proposed to manipulate feature maps, extracted by a separate feature network, to create more transferable image-dependent perturbations using a GAN. Li et al. [LBZ+20] introduced a virtual model known as a ghost network to apply feature-level perturbations to an existing model to produce a large set of diverse models. They showed that ghost networks, together with a coupled ensemble strategy, improve the transferability of existing techniques. Wu et al. [WZTE18] empirically investigated the dependence of adversarial transferability on model-specific attributes, including model capacity, architecture, and test accuracy. They demonstrated that fooling rates heavily depend on the similarity of the source model and target model architectures.

In this chapter, we increase the transferability of generated UAPs by including a loss term inspired by Mopuri et al. [MGR19] focusing on the adversarial energy in early layers.

### 3 Method

#### 3.1 Mathematical Notations

We first introduce notations for image classification and semantic segmentation in a natural domain without attacks and then extend this to the adversarial domain.

**Natural Domain:** Consider a *classifier*  $S$  trained on a training set  $\mathcal{X}^{\text{train}}$  having  $M$  different classes. This network assigns a label  $m = S(\mathbf{x}) \in \mathcal{M} = \{1, \dots, M\}$  to each input image  $\mathbf{x}$  from training set  $\mathcal{X}^{\text{train}}$  or test set  $\mathcal{X}^{\text{test}}$ . We assume that image  $\mathbf{x} \in \mathbb{I}^{H \times W \times C}$  is a clean image, meaning that it contains no adversarial perturbations, with height  $H$ , width  $W$ ,  $C = 3$  color maps, and  $\mathbb{I} = [0, 1]$ . Each image  $\mathbf{x}$  is tagged with a ground truth label  $\bar{m} \in \mathcal{M}$ .

In *semantic segmentation*, given an input image  $\mathbf{x}$ , we assign each pixel with a class label. In this case, the semantic segmentation network  $S$  outputs a label map  $\mathbf{m} = S(\mathbf{x}) \in \mathcal{M}^I$  for each input image  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_I)$ , with  $I = H \times W$ . Similar to before, each pixel  $\mathbf{x}_i \in \mathbb{I}^C$  of an image  $\mathbf{x}$  is tagged with a ground truth label  $\bar{m}_i \in \mathcal{M}$ , resulting in the ground truth label map  $\bar{\mathbf{m}}$  for the entire image  $\mathbf{x}$ .

**Adversarial domain:** Let  $\mathbf{x}^{\text{adv}}$  be an adversarial example that belongs to the adversarial space of the network; for example, for the network  $S$  this space is defined as  $\mathcal{X}_S^{\text{adv}}$ . In order to have a quasi-imperceptible perturbation  $\mathbf{r}$  when added to clean images to obtain adversarial examples, i.e.,  $\mathbf{x}^{\text{adv}} = \mathbf{x} + \mathbf{r}$ , it is bounded by  $\|\mathbf{r}\|_p \leq \epsilon$ , with  $\epsilon$  being the supremum of a respective  $p$ -norm  $\|\cdot\|_p$ .

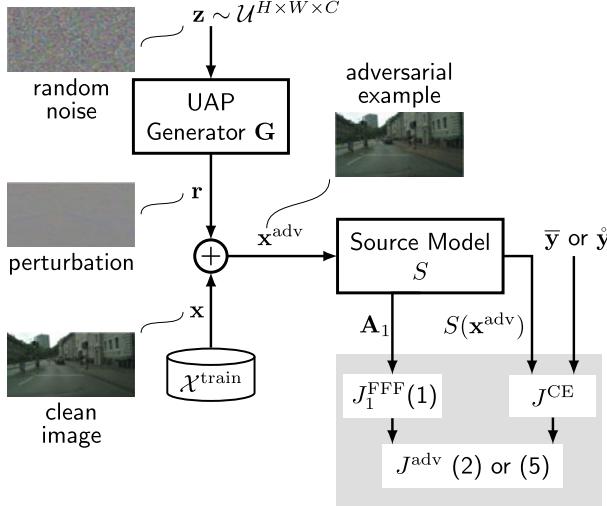
In case of *classification* networks, for each  $\mathbf{x}^{\text{adv}} \in \mathcal{X}_S^{\text{adv}}$ , the purpose of non-targeted attacks is to obtain  $S(\mathbf{x}^{\text{adv}}) \neq \bar{m}$ . In targeted attacks, the attacker tries to enforce  $S(\mathbf{x}^{\text{adv}}) = \hat{m} \neq \bar{m}$ , where  $\hat{m}$  denotes the target class the attacker aims at.

In case of *semantic segmentation* networks, for each  $\mathbf{x}^{\text{adv}} \in \mathcal{X}_S^{\text{adv}}$ , in non-targeted attacks we aim at  $S(\mathbf{x}^{\text{adv}}) = \mathbf{m} = (m_i)$  with  $m_i \neq \bar{m}_i$  for all  $i \in \mathcal{I} = \{1, \dots, I\}$ . On the other hand, in static targeted attacks, the goal is  $S(\mathbf{x}^{\text{adv}}) = \hat{\mathbf{m}} \neq \bar{\mathbf{m}}$ , where  $\hat{\mathbf{m}}$  denotes the target mask of the attacker.

Also, let  $T$  be the target model under attack, which is a deep neural network (DNN) with given (frozen) parameters, pretrained on some datasets to perform a specific environment perception task. We define  $\mathbf{z}$  as a random variable sampled from a distribution, which is fed to a UAP generator  $\mathbf{G}$  to produce a perturbation  $\mathbf{r} = \mathbf{G}(\mathbf{z})$ . Also,  $J$  stands for a loss function, which is differentiable with respect to the network parameters and the input.

#### 3.2 Method Introduction and Initial Analysis

In order to improve the robustness of DNNs for environment perception, knowledge of sophisticated image-agnostic adversarial attacks is needed, capable of working in both white-box and black-box fashions.

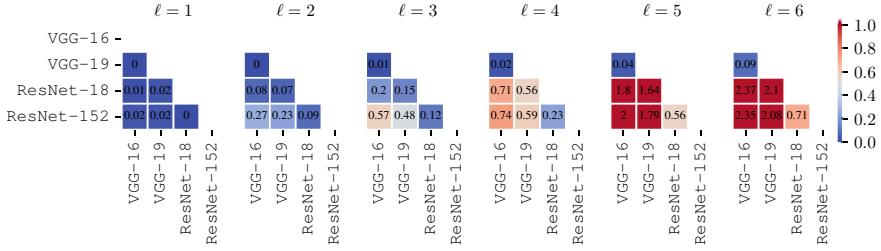


**Fig. 2** Proposed training approach for a UAP generator for non-targeted attacks (label  $\bar{y}$ ) and targeted attacks ( $\hat{y}$ )

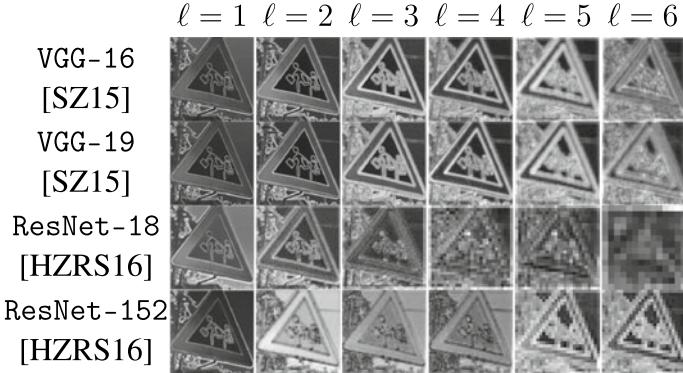
In this section, we present our proposed approach to generate UAPs. It builds upon the adversarial perturbation generator proposed by Poursaeed et al. [PKGB18]. Unlike [PKGB18], we focus on a fooling loss function for generating effective UAPs in both white-box and black-box scenarios. We employ a pretrained DNN as the source model  $S$ , which is exposed to UAPs during training the UAP generator. Our goal is to find a UAP  $\mathbf{r}$  by an appropriate loss function, which is able to not only deceive the source model  $S$  on a training or test dataset  $\mathcal{X}^{\text{train}}$  or  $\mathcal{X}^{\text{test}}$ , respectively, but also to effectively deceive a target model  $T$ , for which  $T \neq S$  holds.

Figure 2 gives an overview of our UAP generator training methodology. Let  $\mathbf{G}(\mathbf{z}) = \mathbf{r}$  be the UAP generator function mapping an unstructured, random multi-dimensional input  $\mathbf{z} \sim \mathcal{U}^{H \times W \times C}$  sampled from a prior uniform distribution  $\mathcal{U} = \mathbb{I}$ , onto a  $p$ -norm scaled  $\mathbf{r}$ . To obtain a  $p$ -norm scaled  $\mathbf{r}'$ , a preliminary obtained perturbation  $\mathbf{r}'$  is bounded by multiplying the generator network raw output  $\mathbf{r}' = \mathbf{G}'(\mathbf{z})$  with  $\min(1, \frac{\epsilon}{\|\mathbf{G}'(\mathbf{z})\|_p})$ . Next, the resulting adversarial perturbation  $\mathbf{r}$  is added to an image  $\mathbf{x} \in \mathcal{X}^{\text{train}}$  before being clipped to a valid range of RGB image pixel values, resulting in an adversarial example  $\mathbf{x}^{\text{adv}}$ . Finally, the generated adversarial example  $\mathbf{x}^{\text{adv}}$  is fed to a pretrained source model  $S$  to compute the adversarial loss functions based on targeted or non-targeted attack types.

To increase the model transferability of the generated UAPs, we seek similarities between different pretrained DNNs to take advantage of this property. For this, we selected some state-of-the-art classifiers such as VGG-16, VGG-19 [SZ15], ResNet-18, and ResNet-152 [HZRS16], all pretrained on ImageNet [RDS+15], to investigate their extracted feature maps in different levels. Then, we first measure the similarity of the mean feature maps of a layer between all networks over the entire



**Fig. 3** Mean squared error (MSE)  $\|\mathbf{A}_\ell^{\text{net}1}(\mathbf{x}) - \mathbf{A}_\ell^{\text{net}2}(\mathbf{x})\|_2^2$  between the mean of feature representations  $\mathbf{A}_\ell(\mathbf{x})$  in layers  $\ell \in \{1, 2, 3, 4, 5, 6\}$  of different DNN classifiers pretrained on the ImageNet dataset. The results are reported for images  $\mathbf{x}$  from the ImageNet validation dataset. All networks show a considerable similarity in terms of MSE in the first layer, whereas similarity in later layers is only seen among VGG-16 and VGG-19 [SZ15]



**Fig. 4** The layer-wise mean of feature representations  $\mathbf{A}_\ell(\mathbf{x})$  within different pretrained classifiers, computed for the first six layers for a random input image  $\mathbf{x}$  taken from the ImageNet validation dataset. High similarity is observed in the first layers, in the later layers only among the VGG-16 and the VGG-19 network

ImageNet [RDS+15] validation set, using the well-known and universally applicable mean squared error (MSE).<sup>1</sup> Figure 3 displays the resulting heat maps. In addition, Fig. 4 shows the mean of feature representations  $\mathbf{A}_\ell(\mathbf{x})$  for these four pretrained classifiers computed for layer  $\ell = 1$  up to  $\ell = 6$  (after each activation function) for a selected input image  $\mathbf{x}$ . Both figures, Figs. 3 and 4, show that the respective networks share a qualitatively and quantitatively high similarity in the first layer compared to all subsequent layers. Only for close relatives, such as VGG-16 and VGG-19, this similarity is found in later layers as well. We thus hypothesize that by applying the fast feature fool loss  $J_1^{\text{FFF}}$  (1) *only to the first layer* of the source model during train-

<sup>1</sup> We also evaluated the similarity of feature maps in different layers by the structural similarity index (SSIM) [WBSS04] and peak signal to noise ratio (PSNR) [HZ10]. The results are very similar to Fig. 3.

ing, we not only inject high adversarial energy into the first layer but also increase the transferability of the generated UAPs.

In the following, we formulate our fooling loss and consider the non-targeted and targeted attack cases (see Fig. 2).

### 3.3 Non-targeted Perturbations

In the non-targeted case, we want to fool the source model  $S$  so that its prediction  $S(\mathbf{x}^{\text{adv}})$  differs from the ground truth one-hot representation  $\bar{\mathbf{y}}$ . In the simplest and most sensible possible way, researchers define the negative cross-entropy as the fooling loss for non-targeted attacks, while Poursaeed et al. [PKGB18] proposed the logarithm of this loss function.

For *image classification*, we define the generator fooling loss for our non-targeted attacks as

$$J^{\text{adv, nontargeted}} = -\alpha \cdot J^{\text{CE}}(S(\mathbf{x}^{\text{adv}}), \bar{\mathbf{y}}) + (1-\alpha) \cdot J_1^{\text{FFF}}(\mathbf{x}^{\text{adv}}), \quad (2)$$

where  $J^{\text{CE}}$  denotes the cross-entropy loss,  $\bar{\mathbf{y}} = (\bar{y}_\mu) \in \{0, 1\}^M$  is the one-hot encoding of the ground truth label  $\bar{m}$  for image  $\mathbf{x}$ , and  $\mu$  being the class index, such that  $\bar{m} = \arg \max_{\mu \in \mathcal{M}} \bar{y}_\mu$ . Also, let  $J_1^{\text{FFF}}(\mathbf{x}^{\text{adv}})$  be the fast feature fool loss of layer  $\ell = 1$  (see (1)), when  $\mathbf{x}^{\text{adv}}$  is fed to the network  $S$ . Further, the cross-entropy loss is defined as

$$J^{\text{CE}}(\mathbf{y}, \bar{\mathbf{y}}) = - \sum_{\mu \in \mathcal{M}} \bar{y}_\mu \log(y_\mu) = -\log(y_{\bar{m}}), \quad (3)$$

where  $\mathbf{y} = S(\mathbf{x}^{\text{adv}}) = (y_\mu) \in \mathbb{I}^M$  is the network output vector of  $S$  with the predictions for each class  $\mu$ . For optimization, we utilize Adam [KB15] in standard configuration, following [PKGB18].

For *semantic segmentation*, in (2)  $\mathbf{y} = S(\mathbf{x}^{\text{adv}}) \in \mathbb{I}^{H \times W \times M}$  and  $\bar{\mathbf{y}} \in \{0, 1\}^{H \times W \times M}$  holds, and the cross-entropy loss in (3) is changed to

$$\begin{aligned} J^{\text{CE}}(\mathbf{y}, \bar{\mathbf{y}}) &= - \sum_{i \in \mathcal{I}} \sum_{\mu \in \mathcal{M}} \bar{y}_{i,\mu} \log(y_{i,\mu}) \\ &= - \sum_{i \in \mathcal{I}} \log(y_{i,\bar{m}_i}), \end{aligned} \quad (4)$$

with  $y_{i,\mu}$ ,  $\bar{y}_{i,\mu}$  being the prediction and ground truth for class  $\mu$  at pixel  $i$ , respectively, and  $y_{i,\bar{m}_i}$  being the prediction at pixel  $i$  for the ground truth class  $\bar{m}_i$  of pixel  $i$ .

### 3.4 Targeted Perturbations

Different from the non-targeted case, the goal of a targeted one is to let the DNN output  $S(\mathbf{x}^{\text{adv}})$  take on values  $\hat{\mathbf{y}}$  defined by the attacker, which usually differ from the ground truth (if source target labels align with the ground truth of a particular image, the UAP will output that ground truth label). Hence, the attacker aims to decrease the cross-entropy loss with respect to a target until the source model  $S$  predicts the selected target class with high confidence. As before, we add the fast feature fool loss in the first layer to boost the transferability of the targeted generated UAP.

For *image classification*, our generator fooling loss for targeted attacks is

$$J^{\text{adv, targeted}} = \alpha \cdot J^{\text{CE}}(S(\mathbf{x}^{\text{adv}}), \hat{\mathbf{y}}) + (1 - \alpha) \cdot J_1^{\text{FFF}}(\mathbf{x}^{\text{adv}}), \quad (5)$$

where  $\hat{\mathbf{y}} \in \{0, 1\}^M$  is the one-hot encoding of the target label  $\hat{m} \neq \bar{m}$ . Note that the sign of the cross entropy is flipped compared to the non-targeted case (2). Then, similar to the non-targeted attack, the Adam optimizer is utilized.

If we consider *semantic segmentation*, the ground truth  $\hat{\mathbf{y}}$  in (5) becomes a one-hot encoded semantic segmentation mask  $\hat{\mathbf{y}} \in \{0, 1\}^{H \times W \times M}$ .

## 4 Experiments on Image Classification

In this section, we will first describe the dataset, network architectures, and evaluation metrics, which are used to measure the performance of generated UAPs on *image classifiers*. Afterward, we will analyze the effectiveness of the proposed fooling method on state-of-the-art image classifiers and will compare it to other state-of-the-art attack methods.

### 4.1 Experimental Setup

**Dataset:** We use the ImageNet dataset [RDS+15], which is a large collection of human-annotated images. For all our experiments, a universal adversarial perturbation is computed for a set of 10,000 images taken from the ImageNet training set  $\mathcal{X}^{\text{train}}$  (i.e., 10 images per class) and the results are reported on the ImageNet validation set  $\mathcal{X}^{\text{val}}$  (50,000 images).

**Network architectures:** There are several design options regarding the architecture choices for generator  $G$  and source model  $S$ . For our generator, we follow [ZPIE17] and [PKGB18] and choose the ResNet generator from [JAFF16], which consists of some convolution layers for downsampling, followed by some residual blocks, before performing upsampling using transposed convolutions. As topology for the source model  $S$ , we utilize the same set of pretrained image classifiers as for the target

**Table 1** Fooling rates (%) of our proposed non-targeted UAPs in white-box attacks, for different values of  $\alpha$  and various target classifiers pretrained on ImageNet. Results are reported on a second training set of 10,000 images. The adversarial perturbation is bounded by  $L_\infty(\mathbf{r}) \leq \epsilon = 10$ . The highest fooling rates (%) are printed in boldface

$\alpha$	Source Model $S =$ Target Model $T$				Avg
	VGG-16	VGG-19	ResNet-18	ResNet-152	
0	8.52	8.29	7.24	4.04	7.02
0.6	90.49	93.48	88.93	84.41	89.32
0.7	<b>95.20</b>	<b>93.79</b>	<b>89.16</b>	87.05	<b>91.30</b>
0.8	90.03	93.24	89.07	<b>89.91</b>	90.56
0.9	95.13	92.14	88.34	89.37	91.24
1	92.87	71.88	88.88	85.34	84.74

**Table 2** Fooling rates (%) of our proposed non-targeted UAPs in white-box attacks, for various target classifiers pretrained on ImageNet. Results are reported on the ImageNet validation set. We report results for two  $L_p$  norms, namely  $L_2(\mathbf{r}) \leq \epsilon = 2000$  and  $L_\infty(\mathbf{r}) \leq \epsilon = 10$

$p$	$\epsilon$	$\alpha$	Source Model $S =$ Target Model $T$			
			VGG-16	VGG-19	ResNet-18	ResNet-152
2	2000	0.7	96.57	94.99	91.85	88.73
$\infty$	10	0.7	95.70	94.00	90.46	90.40

model  $T$ , i.e., VGG-16, VGG-19 [SZ15], ResNet-18, ResNet-152 [HZRS16], and also GoogleNet [SLJ+15].

**Evaluation metrics:** We use the fooling rate as our metric to assess the performance of our crafted UAPs on DNN image classifiers [MDFFF17, PKGB18, MUR18, MGR19]. In the case of non-targeted attacks, it is the percentage of input images for which  $T(\mathbf{x}^{\text{adv}}) \neq T(\mathbf{x})$  holds. For targeted attacks, we calculate the top-1 target accuracy, which can be understood as the percentage of adversarial examples, that is classified “correctly” as the target class as desired by the attacker.

## 4.2 Non-Targeted Universal Perturbations

According to Fig. 2, we train our model with the non-targeted fooling loss (2). For tuning the hyperparameter  $\alpha$ , the weight of our novel adversarial loss components, we utilized a second training set of 10,000 random images (again 10 images per class) taken from the ImageNet training set which is disjoint from both the training and the validation dataset. Table 1 shows that the best  $\alpha$  for non-targeted attacks, on average over all model topologies, is  $\alpha = 0.7$ .

For white-box attacks, where  $S = T$  holds, results on the ImageNet validation set for two different norms are given in Table 2. The maximum permissible  $L_p$  norm of

**Table 3** Fooling rates (%) of various non-targeted state-of-the-art methods on various target classifiers trained on ImageNet (white-box attacks). The results of other state-of-the-art methods are reported from the respective paper. <sup>+</sup>For comparison reasons, the average of our method leaves out the ResNet-18 model results. Highest fooling rates are printed in boldface

$p$	$\epsilon$	Method	$S = T$				Avg <sup>+</sup>
			VGG-16	VGG-19	ResNet-18	ResNet-152	
$\infty$	*10	FFF	47.10	43.62	-	29.78	40.16
		CIs	71.59	72.84	-	60.72	68.38
		UAP	78.30	77.80	-	84.00	80.03
		GAP	83.70	80.10	-	82.70	82.16
		NAG	77.57	83.78	-	87.24	82.86
		TUAP-RSI	94.30	<b>94.98</b>	-	90.08	93.12
		<b>Ours</b>	<b>95.70</b>	94.00	<b>90.46</b>	<b>90.40</b>	<b>93.36</b>
2	2000	UAP	90.30	84.50	-	88.50	87.76
		GAP	93.90	94.90	-	79.50	89.43
		<b>Ours</b>	<b>96.57</b>	<b>94.99</b>	<b>91.85</b>	<b>88.73</b>	<b>93.43</b>



(a) The UAP  $\mathbf{r}$  (left) and the respective adversarial examples  $\mathbf{x}^{\text{adv}} = \mathbf{x} + \mathbf{r}$  (right), with  $L_2(\mathbf{r}) \leq 2000$ .



(b) Original images  $\mathbf{x}$ .



(c) The UAP  $\mathbf{r}$  (left) and the respective adversarial examples  $\mathbf{x}^{\text{adv}} = \mathbf{x} + \mathbf{r}$  (right), with  $L_\infty(\mathbf{r}) \leq 10$ .

**Fig. 5** Examples of our non-targeted UAPs and adversarial examples. In **a** the universal adversarial perturbation is given on the left and eight different adversarial examples are shown on the right, where the  $L_2$  norm of the adversarial perturbation is bounded by  $\epsilon = 2000$ , i.e.,  $L_2(\mathbf{r}) \leq 2000$ . In **b** the respective original images are shown, whereas in **c** the  $L_\infty$  norm of the adversarial perturbation is bounded by  $\epsilon = 10$ , i.e.,  $L_\infty(\mathbf{r}) \leq 10$ ,  $\alpha = 0.7$ . In these experiments, the source model  $S$  is ResNet-18 [HZRS16]. The pixel values in UAPs are scaled for better visibility

the perturbations for  $p = 2$  and  $p = \infty$  is set to be  $\epsilon = 2000$  and  $\epsilon = 10$ , respectively, following [MDFFF17]. As Moosavi-Dezfooli et al. [MDFFF17] pointed out, these values are selected to acquire a perturbation whose norm is remarkably smaller than the average image norms in the ImageNet dataset to obtain quasi-imperceptible adversarial examples. The results in Table 2 show that the proposed method is successful in the white-box setting. For the  $L_\infty$  norm, all reported fooling rate numbers

**Table 4** Transferability of our proposed non-targeted UAPs (white-box and black-box attacks). Results are reported in the form of fooling rates (%) for various combinations of source models  $S$  and target models  $T$ , pretrained on ImageNet. The generator is trained to fool the source model (rows), and it is tested on the target model (columns). The adversarial perturbation is bounded by  $L_\infty(\mathbf{r}) \leq \epsilon = 10$ ,  $\alpha = 0.7$ . \*The average is computed *excluding* the easier white-box attacks (main diagonal)

		Target model $T$				Avg*
		VGG-16	VGG-19	ResNet-18	ResNet-152	
Source model $S$	VGG-16	<b>95.70</b>	86.67	49.98	36.34	57.66
	VGG-19	84.77	<b>94.00</b>	47.24	36.46	56.15
	ResNet-18	76.49	72.18	<b>90.46</b>	50.46	66.37
	ResNet-152	86.19	82.36	76.04	<b>90.40</b>	<b>81.53</b>

are above 90%. To illustrate that our adversarial examples are quasi-imperceptible to humans, we illustrate some adversarial examples with their respective scaled UAPs in Fig. 5.

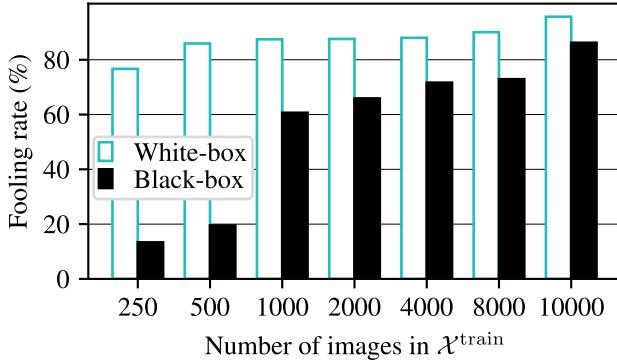
In Table 3, we compare our proposed approach in generating non-targeted UAPs with state-of-the-art methods, i.e., fast feature fool (FFF, as originally proposed, on all layers) [MGR19], class impressions (CIs) [MUR18], universal adversarial perturbation (UAP) [MDFFF17], generative adversarial perturbation (GAP) [PKGB18], network for adversary generation (NAG) [RMOGB18], and targeted UAP-random source image (TUAP-RSI) [ZBIK20b]. In these experiments, we again consider the white-box setting, i.e.,  $S = T$ . Our proposed approach achieves a new state-of-the-art performance on *almost all* models on *both*  $L_p$  norms, being on average 4% absolute better in fooling rate with the  $L_2$  norm, and at least 0.26% absolute better with the  $L_\infty$  norm.

In Table 4 we investigate the white-box ( $S = T$ ) and black-box ( $S \neq T$ ) capability of our proposed method through various combinations of source and target models. Note that the rightmost column represents an average over the fooling rates in the black-box settings and thus indicates the transferability of our proposed method. Overall, in the white-box and black-box settings, we achieve fooling rates of over 90% and 55%, respectively. We also compare the transferability of our produced UAPs with the same state-of-the-art methods as before. The results for these experiments are shown in Table 5, where VGG-16, ResNet-152, and GoogleNet are used as the source model in Table 5a–c, respectively. It turns out to be advisable to choose a deep network as the source model (ResNet-152); since then our performance on the unseen VGG-16 and VGG-19 target models is about 12% absolute better than earlier state of the art ( $L_\infty$  norm).

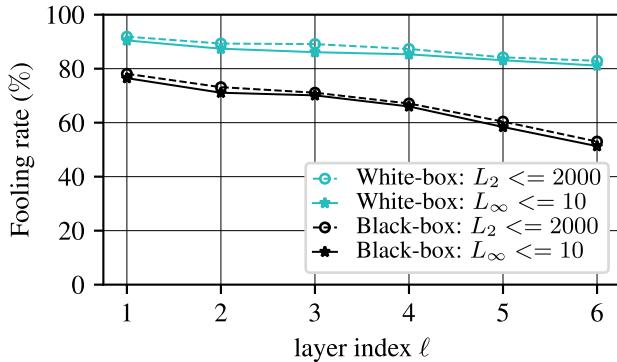
For investigating the generalization power of UAPs w.r.t. unseen data, we evaluate the influence of the size of the training dataset  $X^{\text{train}}$  on the quality of UAPs in conducting white-box and black-box attacks. Figure 6 shows the fooling rates obtained for VGG-16 as the target model  $T$ , on the ImageNet validation set for different sizes

**Table 5** Transferability of our proposed non-targeted UAPs compared to other methods, i.e., FFF [MGR19], CIs [MUR18], UAP [MDFFF17], GAP [PKGB18], and NAG [RMOGBV18], using different source models  $S$  and target models  $T$ . The UAP is bounded by  $L_\infty(\mathbf{r}) \leq \epsilon = 10$ . Values of our method are taken from Table 4 (except GoogleNet). \*Note that the results are reported from the respective paper. Highest fooling rates are printed in boldface

(a) $S$ : VGG-16 [SZ15]		
$T$	Method	Fooling Rate (%)
VGG-19	FFF*	41.98
	CIs*	65.64
	UAP*	73.10
	GAP	79.14
	NAG*	73.25
	<b>Ours</b>	<b>86.67</b>
ResNet-152	FFF*	27.82
	CIs*	45.33
	UAP*	<b>63.40</b>
	GAP	30.32
	NAG*	54.38
	<b>Ours</b>	36.34
ResNet-18	<b>Ours</b>	<b>49.98</b>
(b) $S$ : ResNet-152 [HZRS16]		
$T$	Method	Fooling Rate (%)
VGG-16	FFF*	19.23
	CIs*	47.21
	UAP*	47.00
	GAP	70.45
	NAG*	52.17
	<b>Ours</b>	<b>86.19</b>
VGG-19	FFF*	17.15
	CIs*	48.78
	UAP*	45.50
	GAP	70.38
	NAG*	53.18
	<b>Ours</b>	<b>82.36</b>
ResNet-18	<b>Ours</b>	<b>76.04</b>
(c) $S$ : GoogleNet [SLJ+15]		
$T$	Method	Fooling Rate (%)
VGG-16	FFF*	40.91
	CIs*	59.12
	UAP*	39.20
	GAP	71.14
	NAG*	56.40
	<b>Ours</b>	<b>75.35</b>
ResNet152	FFF*	25.31
	CIs*	47.81
	UAP*	45.50
	GAP	51.72
	NAG*	59.22
	<b>Ours</b>	<b>61.24</b>
ResNet-18	<b>Ours</b>	<b>67.30</b>



**Fig. 6** Fooling rates (%) of our non-targeted UAPs on the ImageNet validation set for different sizes of  $\mathcal{X}^{\text{train}}$  in both white-box and black-box settings. In the white-box setting, the source model  $S$  and the target model  $T$  are VGG-16, while in the black-box setting, the source model  $S$  is ResNet-152 and the target model  $T$  is again VGG-16. Results are reported for  $L_{\infty}(\mathbf{r}) \leq \epsilon = 10$  and  $\alpha = 0.7$



**Fig. 7** Fooling rates (%) of our non-targeted UAPs on the ImageNet validation set, in both white-box and black-box attacks for  $L_{\infty}$  and  $L_2$ , for different layers  $\ell$  in (1) applied in the loss function (2). In the white-box setting, the source model  $S$  and the target model  $T$  are ResNet-18. In the black-box setting, the source model  $S$  is again ResNet-18 and the target model  $T$  is VGG-16. Results are reported for  $\alpha = 0.7$

of  $\mathcal{X}^{\text{train}}$ . The results show that by using a dataset  $\mathcal{X}^{\text{train}}$  containing only 1000 images, our approach leads to a fooling rate of more than 60% on the ImageNet validation dataset in both the white-box and black-box settings. Additionally, the number of images in  $\mathcal{X}^{\text{train}}$  turns out to be more vital for the fooling rate of black-box attacks as compared to white-box attacks.

To examine the impact of the layer which is used in our loss function, we utilized different layers in (1), then applied them in the loss function (2) to train the source model for generating UAPs. In practice, we are interested in the impact the layer position has. Figure 7 shows the fooling rate in white-box and black-box settings for  $L_{\infty}$  and  $L_2$ , when different layers from  $\ell = 1$  to  $\ell = 6$  are applied in (1) and

(2). This figure shows that choosing deeper layers leads to a decreasing trend in the fooling rate. Also, this trend is stronger in black-box attacks, where a more than 20% drop in the attack success rate can be observed. This indicates that it is advisable to choose earlier layers, in particular the first layer, in generating UAPs to obtain both an optimal fooling rate and transferability.

### 4.3 Targeted Universal Perturbations

In this section, we apply the targeted fooling loss (5), again with  $\alpha = 0.7$ , for training the generator in Fig. 2. We assume the chosen  $\alpha$  is appropriate for targeted attacks as well and thus dispense further investigations. If results are good, then this supports some robustness w.r.t. the choice of  $\alpha$ . Figure 8 depicts two examples of our targeted UAPs, some original images and respective adversarial examples. In these experiments, the fooling rate (top-1 target accuracy) on the validation set for the target class  $\hat{m} = 919$  (street sign) and  $\hat{m} = 920$  (traffic light, traffic signal, stoplight) are 63.2 and 57.83%, respectively, which underlines the effectiveness of our approach.

For assessing the generalization power of our proposed method across different target classes and comparison with GAP [PKGB18], we followed Poursaeed et al. and used 10 randomly sampled classes. The resulting average top-1 target accuracy, when the adversarial perturbation is bounded by  $L_\infty(\mathbf{r}) \leq \epsilon = 10$ , is 66.57%, which is significantly higher than the one reported for GAP [PKGB18] with 52.0%.



(a) The UAP  $\mathbf{r}$  for target label traffic light, signal, stoplight (left) and the respective adversarial examples  $\mathbf{x}^{\text{adv}}$  (right).

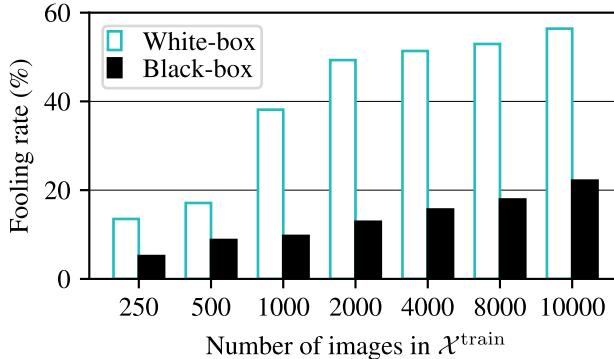


(b) Original images  $\mathbf{x}$ .



(c) The UAP  $\mathbf{r}$  for target label street sign (left) and the respective adversarial examples  $\mathbf{x}^{\text{adv}}$  (right).

**Fig. 8** Examples of our targeted UAPs and adversarial examples. In these experiments, the source model  $S$  is VGG-16 [SZ15], with  $L_\infty(\mathbf{r}) \leq \epsilon = 10$ ,  $\alpha = 0.7$ . The pixel values in UAPs are scaled for better visibility



**Fig. 9** Fooling rates (%) of our targeted UAPs on the ImageNet validation set for different sizes of  $\mathcal{X}^{\text{train}}$  in both white-box and black-box attacks. In the white-box setting, the source model  $S$  and the target model  $T$  are VGG-16, while in the black-box setting, the source model  $S$  is again VGG-16 and the target model  $T$  is VGG-19. Results are reported for  $L_\infty(\mathbf{r}) \leq \epsilon = 10$ ,  $\alpha = 0.7$ , and target class  $\hat{m} = 847$  (tank, army tank, armored combat vehicle, armoured combat vehicle)

To demonstrate the generalization power of our targeted UAPs w.r.t. unseen data, we visualize the attack success rate obtained by the source model VGG-16, in white-box and black-box settings, on the Image-Net validation dataset for different sizes of  $\mathcal{X}^{\text{train}}$  in Fig. 9. For instance, with  $\mathcal{X}^{\text{train}}$  containing 10,000 images, we are able to fool the target model on over 20% of the images in the ImageNet validation set. It should be noted that training the generator  $\mathbf{G}$  to produce a single UAP forcing the target model to output a specific target class  $\hat{m}$  is an extremely challenging task. However, we particularly observe that utilizing 10,000 training images again seems to be sufficient for a white-box attack.

## 5 Experiments on Semantic Segmentation

We continue our investigations by applying our method to the task of semantic segmentation to show its cross-task applicability. We start with the experimental setup, followed by an evaluation of non-targeted and targeted attacks.

### 5.1 Experimental Setup

**Dataset:** We conduct our experiments on the widely known Cityscapes dataset [COR+16]. It contains pixel-level annotations of 5,000 high-resolution images (2,975 training, 500 validation, and 1,525 test images) being captured in urban street scenes.

**Table 6** The mean intersection-over-union (mIoU) (%) of non-targeted UAP methods on the semantic segmentation model FCN-8 pretrained on Cityscapes. Our method is compared with GAP [PKGB18]. In these experiments, both the source model  $S$  and the target model  $T$  are either the same, i.e.,  $S = T = \text{FCN-8}$ , or different, i.e.,  $S = \text{ERFNet}$ ,  $T = \text{FCN-8}$ . Parameters are set to  $L_\infty(\mathbf{r}) \leq \epsilon$ ,  $\alpha = 0.7$ . Best results are printed in boldface

(a) $S = T = \text{FCN-8}$					
Method	$\epsilon$	2	5	10	20
GAP	18.1	12.8	4.0	2.1	
<b>Ours</b>	<b>16.2</b>	<b>9.8</b>	<b>2.0</b>	<b>0.3</b>	
(b) $S = \text{ERFNet}$ , $T = \text{FCN-8}$					
Method	$\epsilon$	2	5	10	20
GAP	27.3	16.4	7.0	4.3	
<b>Ours</b>	<b>26.4</b>	<b>14.8</b>	<b>4.1</b>	<b>1.5</b>	

The original images have a resolution of  $2048 \times 1024$  pixels, that we downsample to  $1024 \times 512$  for our experiments [MCKBF17, PKGB18].

**Network architecture:** Regarding the architecture of the source model  $S$ , we use FCN-8 [LSD15] for white-box attacks, and ERFNet [RABA18] for the black-box setting. FCN-8 consists of an encoder part which transforms an input image into a low-resolution semantic representation and a decoder part which recovers the high spatial resolution of the image by fusing different levels of feature representations together. ERFNet also consists of an encoder-decoder structure, but without any bypass connections between the encoder and the decoder. Additionally, residual units are used with factorized convolutions to obtain a more efficient computation.

In our experiments, we consider FCN-8 [LSD15] as our segmentation target model  $T$ , and use  $L_\infty$  norm, to be comparable with [MCKBF17, PKGB18].

**Evaluation metrics:** To assess the performance of a semantic segmentation network, we used the mean intersection-over-union (mIoU) [COR+16]. It is defined as

$$\text{mIoU} = \frac{1}{|\mathcal{M}|} \sum_{\mu \in \mathcal{M}} \frac{\text{TP}_\mu}{\text{TP}_\mu + \text{FP}_\mu + \text{FN}_\mu}, \quad (6)$$

with class  $\mu \in \mathcal{M}$ , class-specific true positives  $\text{TP}_\mu$ , false positives  $\text{FP}_\mu$ , and false negatives  $\text{FN}_\mu$ . For assessing the impact of non-targeted adversarial attacks on semantic segmentation, we compute the mIoU on adversarial examples [AMT18].

To measure the attack success of our targeted attack, we compute the pixel accuracy (PA) between the prediction  $\mathbf{m} = T(\mathbf{x}^{\text{adv}})$  and the target  $\hat{\mathbf{m}}$  [PKGB18]. In this chapter, we restrict our analysis to the static target segmentation scenario [MCKBF17] and use the same target mask as in [PKGB18, MCKBF17] (see also Fig. 11).

## 5.2 Non-targeted Universal Perturbations

For the non-targeted case, we train our model with the non-targeted adversarial loss function (2), where we use (4) as the cross-entropy loss. The maximum permissible  $L_p$  norm of the perturbations for  $p = \infty$  is set to  $\epsilon \in \{2, 5, 10, 20\}$ . We report results for both our method and GAP [PKGB18] on the Cityscapes validation set in Table 6a. It can be observed that across different values for  $\epsilon$  our method is superior to GAP in terms of decreasing the mIoU of the underlying target model.

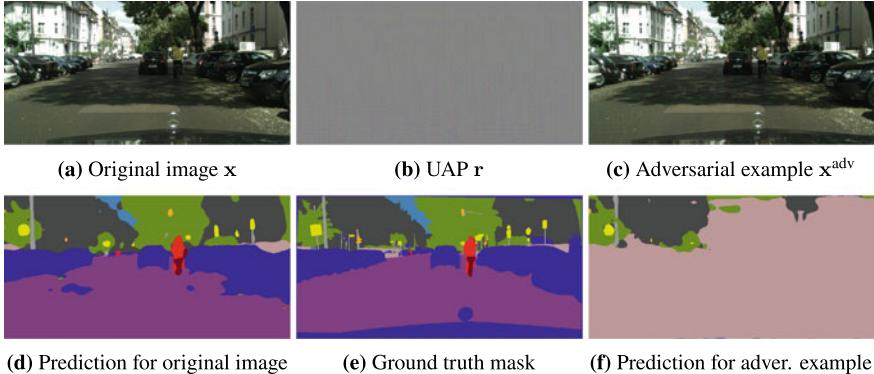
We visualize the effect of the generated non-targeted UAPs in Fig. 10 by illustrating the original image, the UAP, the resulting adversarial example, the prediction on the original image, the ground truth mask, and the resulting segmentation output. Here, the maximum permissible  $L_p$  norm of the perturbations  $p = \infty$  is set to  $\epsilon = 5$ .

For investigating the transferability of our generated UAPs, we use a UAP optimized on ERFNet as the source model  $S$  and test it on the target model  $T$  being FCN-8. Table 6b reports black-box attack results for our method compared to the attack method GAP [PKGB18]. Our non-targeted UAPs decrease the mIoU of the FCN-8 on the Cityscapes dataset more than GAP [PKGB18] does, in all different ranges of adversarial perturbations ( $\epsilon \in \{2, 5, 10, 20\}$ ). These results illustrate the effectiveness of the generated perturbation.

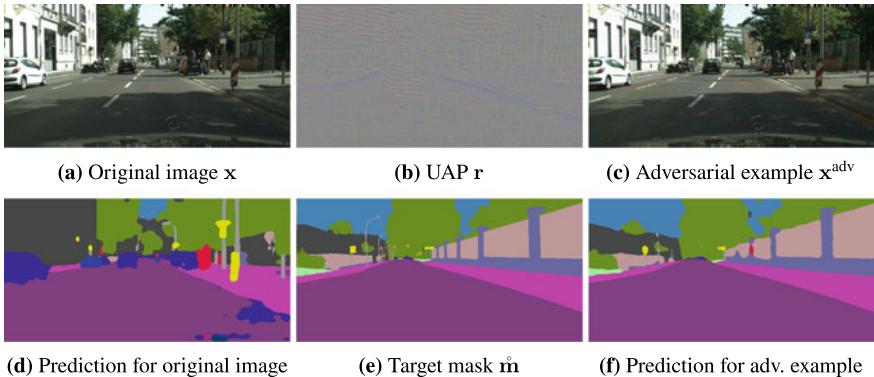
## 5.3 Targeted Universal Perturbations

In the targeted case, we aim at finding a UAP which forces the segmentation source model  $S$  and the segmentation target model  $T$  to predict a specific target mask  $\hat{\mathbf{m}}$ . We now train our model with the targeted adversarial loss function (5), using  $J^{\text{CE}}(\mathbf{y}, \hat{\mathbf{y}})$  according to (4) for the cross-entropy loss. We apply the same target  $\hat{\mathbf{y}}$  as in [PKGB18], see  $\hat{\mathbf{m}}$  in Fig. 11e.

Figure 11 depicts an original image, our generated targeted UAP, the respective adversarial example, the prediction on the original image, the target segmentation mask, and the prediction on the adversarial example. The results show that the generated UAP resembles the target mask and is able to fool the FCN-8 in a way that it now outputs the target segmentation mask.



**Fig. 10** An example of our non-targeted UAPs optimized for the task of semantic segmentation on the Cityscapes dataset. Results are displayed on the Cityscapes validation set. In these experiments, both the source model  $S$  and the target model  $T$  are FCN-8, with  $L_\infty(\mathbf{r}) \leq \epsilon = 5$ ,  $\alpha = 0.7$ . The pixel values in the UAP are scaled for better visibility



**Fig. 11** An example of our targeted UAPs optimized for the task of semantic segmentation on the Cityscapes dataset. Results are displayed on the Cityscapes validation set. In these experiments, both the source model  $S$  and the target model  $T$  are FCN-8, with  $L_\infty(\mathbf{r}) \leq \epsilon = 10$ ,  $\alpha = 0.7$ . The pixel values in the UAP are scaled for better visibility

We compare our targeted attack with two state-of-the-art methods in Table 7. While our method performs comparably well as state of the art for weak attacks, in medium to strong attacks we outperform both GAP [PKGB18] and UAP-Seg [MCKBF17].

**Table 7** Pixel accuracy (%) of targeted UAP methods (white-box attack) on the semantic segmentation model FCN-8 pretrained on the Cityscapes training set. Results are reported on the Cityscapes validation set. Our method is compared to UAP-Seg [MCKBF17] and GAP [PKGB18]. In these experiments, both the source model  $S$  and the target model  $T$  are FCN-8, with  $L_\infty(\mathbf{r}) \leq \epsilon$ ,  $\alpha = 0.7$ . Best results are printed in boldface

Method	$\epsilon$			
	2	5	10	20
GAP	<b>61.2</b>	79.5	92.1	97.2
UAP-Seg	60.9	80.3	91.0	96.3
<b>Ours</b>	61.0	<b>81.8</b>	<b>93.1</b>	<b>97.4</b>

## 6 Conclusions

We presented a novel method to effectively generate targeted and non-targeted universal adversarial perturbations (UAPs) in both white-box and black-box settings. Our proposed method shows new state-of-the-art fooling rates for targeted as well as non-targeted UAPs on different classifiers. Additionally, our non-targeted UAPs show a significantly higher transferability across models when compared to other methods, given that we generated our UAPs on the deepest network in the investigation. This is achieved by incorporating an additional loss term during training, which aims at increasing the activation of the first layer. Finally, we extended our method to the task of semantic segmentation to prove its applicability also in more complex environment perception tasks. Due to its state-of-the-art effectiveness for object classification and semantic segmentation, we strongly recommend to employ the proposed types of attacks in validation of automated vehicles’ environment perception.

## References

- [AM18] N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: a survey. *IEEE Access* **6**, 14410–14430 (2018)
- [AMT18] A. Arnab, O. Miksik, P.H.S. Torr, On the robustness of semantic segmentation models to adversarial attacks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 888–897
- [BHSFs19] A. Bär, F. Hüger, P. Schlicht, T. Fingscheidt, On the robustness of redundant teacher-student frameworks for semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Long Beach, CA, USA, June 2019, pp. 1380–1388
- [BKV+20] A. Bär, M. Klingner, S. Varghese, F. Hüger, P. Schlicht, T. Fingscheidt, Robust semantic segmentation by redundant networks with a layer-specific loss contribution and majority vote, in *Proceedings of the IEEE/CVF Conference on Computer*

- Vision and Pattern Recognition (CVPR) Workshops*, virtual conference, June 2020, pp. 1348–1358
- [BLK+21] A. Bär, J. Löhdefink, N. Kapoor, S.J. Varghese, F. Hüger, P. Schlicht, T. Fingscheidt, The vulnerability of semantic segmentation networks to adversarial attacks in autonomous driving: enhancing extensive environment sensing. *IEEE Signal Process. Mag.* **38**(1), 42–52 (2021)
- [BZIK20] P. Benz, C. Zhang, T. Imtiaz, I.-S. Kweon, Double targeted universal adversarial perturbations, in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, virtual conference, Nov. 2020, pp. 284–300
- [CAB+20] A. Chaubey, N. Agrawal, K. Barnwal, K.K. Guliani, P. Mehta, Universal Adversarial Perturbations: A Survey, pp. 1–20, May 2020. [arxiv:2005.08087](https://arxiv.org/abs/2005.08087)
- [COR+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 3213–3223
- [DLP+18] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 9185–9193
- [FJGN20] J. Fayyad, M.A. Jaradat, D. Gruyer, H. Najjaran, Deep learning sensor fusion for autonomous vehicle perception and localization: a review. *Sensors* **20**(15), 4220–4255 (2020)
- [GPAM+14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, Dec. 2014, pp. 2672–2680
- [GR20] N. Ghafoorianfar, M. Roopaei, Environmental perception in autonomous vehicles using edge level situational awareness, in *Proceedings of the IEEE Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, Jan. 2020, pp. 444–448
- [GRM+19] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F.A. Wichmann, Brendel, W. ImageNet-Trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1–22
- [GSS15] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–11
- [GTCM20] S. Grigorescu, B. Trasnea, T. Cocias, G. Macesanu, A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **37**(3), 362–386 (2020)
- [HD18] J. Hayes, G. Danezis, Learning universal adversarial perturbations with generative models, in *Proceedings of the IEEE Symposium on Security and Privacy (SP) Workshops*, San Francisco, CA, USA, May 2018, pp. 43–49
- [HM19] A.S. Hashemi, S. Mozaffari, Secure deep neural networks using adversarial image generation and training with Noise-GAN. *Comput. Secur.* **86**, 372–387 (2019)
- [HZ10] A. Hore, D. Ziou, Image quality metrics: PSNR versus SSIM, in *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey, Aug. 2010, pp. 2366–2369
- [HZRS16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016, pp. 770–778
- [JAFF16] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 694–711

- [KB15] D.P. Kingma, J. Ba, ADAM: a method for stochastic optimization, in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15
- [KBFs20] M. Klingner, A. Bär, T. Fingscheidt, Improved noise and attack robustness for semantic segmentation by using multi-task training with self-supervised depth estimation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, virtual conference, June 2020, pp. 1299–1309
- [KBJ+20] S. Kuutti, R. Bowden, Y. Jin, P. Barber, S. Fallah, A survey of deep learning applications to autonomous vehicle control. *IEEE Trans. Intell. Transp. Syst. (TITS)* **22**(2), 721–733 (2020)
- [KGB17] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, Toulon, France, Apr. 2017, pp. 1–14
- [LBX+20] Y. Li, S. Bai, C. Xie, Z. Liao, X. Shen, A. Yuille, Regional homogeneity: towards learning transferable universal adversarial perturbations against defenses, in *Proceedings of the European Conference on Computer Vision (ECCV)*, virtual conference, Aug. 2020, pp. 795–813
- [LBZ+20] Y. Li, S. Bai, Y. Zhou, C. Xie, Z. Zhang, A.L. Yuille, Learning transferable adversarial examples via ghost networks, in *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, USA, Feb. 2020, pp. 11458–11465
- [LSD15] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015, pp. 3431–3440
- [MCKBF17] J.H. Metzen, M.C. Kumar, T. Brox, V. Fischer, Universal adversarial perturbations against semantic image segmentation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2774–2783, Venice, Italy, Oct. 2017
- [MDFF+18] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, S. Soatto, Robustness of classifiers to universal perturbations: a geometric perspective, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, Apr. 2018, pp. 1–15
- [MDFFF17] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017, pp. 1765–1773
- [MGR19] K.R. Mopuri, A. Ganeshan, V.B. Radhakrishnan, Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **41**(10), 2452–2465 (2019)
- [MKMW19] N. Morgulis, A. Kreines, S. Mendelowitz, Y. Weisglass, Fooling a Real Car With Adversarial Traffic Signs, June 2019, pp. 1–19. [arxiv:1907.00374](https://arxiv.org/abs/1907.00374)
- [MMS+18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vancouver, BC, Canada, Apr. 2018, pp. 1–10
- [MUR18] K.R. Mopuri, P.K. Uppala, V.B. Radhakrishnan, Ask, Acquire, and Attack: Data-Free UAP generation using class impressions. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, Sept. 2018, pp. 19–34
- [NVM+19] A.M. Nascimento, L.F. Vismari, C.B.S.T. Molina, P.S. Cugnasca, J.B. Camargo, J.R. de Almeida, R. Inam, E. Fersman, M.V. Marquezini, A.Y. Hata, A systematic literature review about the impact of artificial intelligence on autonomous vehicle safety. *IEEE Trans. Intell. Transp. Syst. (TITS)* **21**(12), 4928–4946 (2019)
- [PKGB18] O. Poursaeed, I. Katsman, B. Gao, S. Belongie, Generative adversarial perturbations, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 4422–4431

- [PMJ+16] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. Berkay Celik, A. Swami, The limitations of deep learning in adversarial settings, in *Proceedings of the IEEE European Symposium on Security and Privacy (ESP)*, Saarbrücken, Germany, Mar. 2016, pp. 372–387
- [PTWA17] S.M. Patole, M. Torlak, D. Wang, M. Ali, Automotive radars: a review of signal processing techniques. *IEEE Signal Process. Mag.* **34**(2), 22–35 (2017)
- [PXL+20] H. Phan, Y. Xie, S. Liao, J. Chen, B. Yuan, CAG: a real-time low-cost enhanced-robustness high-transferability content-aware adversarial attack generator, in *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, USA, Feb. 2020, pp. 5412–5419
- [RABA18] E. Romera, J.M. Alvarez, L.M. Bergasa, R. Arroyo, ERFNet: efficient residual factorized convnet for real-time semantic segmentation. *IEEE Trans. Intell. Transp. Syst. (TITS)* **19**(1), 263–272 (2018)
- [RDS+15] O. Russakovsky, J. Deng, S. Hao, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis. (IJCV)* **115**(3), 211–252 (2015)
- [RMQGB18] K.R. Mopuri, U. Ojha, U. Garg, R. Venkatesh Babu, NAG: network for adversary generation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 742–751
- [RT19] A. Rasouli, J.K. Tsotsos, Autonomous vehicles that interact with pedestrians: a survey of theory and practice. *IEEE Trans. Intell. Transp. Syst. (TITS)* **21**(3), 900–918 (2019)
- [SAE18] SAE International, *SAE J3016: Surface Vehicle Recommended Practice – Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles* (SAE International, June 2018)
- [SLJ+15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015, pp. 1–9
- [SOZ+18] Z. Sun, M. Ozay, Y. Zhang, X. Liu, T. Okatani, Feature quantization for defending against distortion of images, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018, pp. 7957–7966
- [SZ15] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14
- [SZS+14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)*, Banff, AB, Canada, Dec. 2014, pp. 1–10
- [WBSS04] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
- [WLCC18] X. Wei, S. Liang, N. Chen, X. Cao, Transferable Adversarial Attacks for Image and Video Object Detection, pp. 1–7, May 2018. [arxiv:1811.12641](https://arxiv.org/abs/1811.12641)
- [WLH+20] Y. Wang, Z. Li, H. Hao, H. Yang, Y. Zheng, Research on visual perception technology of autonomous driving based on improved convolutional neural network. *J. Phys.: Conf. Ser.* **1550**(3), 21–27 (2020)
- [WWX+20] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, X. Ma, Skip Connections Matter: On the Transferability of Adversarial Examples Generated With ResNets, pp. 1–15, Feb. 2020. [arxiv:2002.05990](https://arxiv.org/abs/2002.05990)
- [WXZ20] W. Jiangqing, X. Hao, J. Zhao, Automatic lane identification using the roadside LiDAR sensors. *IEEE Intell. Transp. Syst. Mag.* **12**(1), 25–34 (2020)

- [WZTE18] L. Wu, Z. Zhu, C. Tai, E. Weinan, Understanding and Enhancing the Transferability of Adversarial Examples, pp. 1–15, Feb. 2018. [arxiv:1802.09707](https://arxiv.org/abs/1802.09707)
- [XDL+18] C. Xiao, R. Deng, B. Li, F. Yu, M. Liu, D. Song, Characterizing adversarial examples based on spatial consistency information for semantic segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, Sept. 2018, pp. 217–234
- [XZL+18] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, D. Song, Spatially Transformed Adversarial Examples, pp. 1–29, Jan. 2018. [arxiv:1801.02612](https://arxiv.org/abs/1801.02612)
- [ZBIK20a] C. Zhang, P. Benz, T. Imtiaz, I.-S. Kweon, CD-UAP: class discriminative universal adversarial perturbation, in *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, USA, Feb. 2020, pp. 6754–6761
- [ZBIK20b] C. Zhang, P. Benz, T. Imtiaz, I.-S. Kweon, Understanding adversarial examples from the mutual influence of images and perturbations, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference, June 2020, pp. 14521–14530
- [ZPIE17] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2242–2251
- [ZWJW20] K. Zhang, S.J. Wang, L. Ji, C. Wang, DNN based camera and LiDAR fusion framework for 3D object recognition. *J. Phys.: Conf. Ser.* **1518**(1), 12–44 (2020)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



# Invertible Neural Networks for Understanding Semantics of Invariances of CNN Representations



Robin Rombach, Patrick Esser, Andreas Blattmann, and Björn Ommer

**Abstract** To tackle increasingly complex tasks, it has become an essential ability of neural networks to learn abstract representations. These task-specific representations and, particularly, the invariances they capture turn neural networks into black-box models that lack interpretability. To open such a black box, it is, therefore, crucial to uncover the different semantic concepts a model has learned as well as those that it has learned to be invariant to. We present an approach based on invertible neural networks (INNs) that (i) recovers the task-specific, learned invariances by disentangling the remaining factor of variation in the data and that (ii) invertibly transforms these recovered invariances combined with the model representation into an equally expressive one with accessible semantic concepts. As a consequence, neural network representations become understandable by providing the means to (i) expose their semantic meaning, (ii) semantically modify a representation, and (iii) visualize individual learned semantic concepts and invariances. Our invertible approach significantly extends the abilities to understand black-box models by enabling post hoc interpretations of state-of-the-art networks without compromising their performance. Our implementation is available at <https://compvis.github.io/invariances/>.

## 1 Introduction

Key to the wide success of deep neural networks is end-to-end learning of powerful hidden representations that aim to (*i*) capture all task-relevant characteristics while (*ii*) being invariant to all other variabilities in the data [LeC12, AS18]. Deep learning

---

R. Rombach (✉) · P. Esser · A. Blattmann · B. Ommer  
HCI, IWR, Heidelberg University, Berliner Str. 43, 69120 Heidelberg, Germany  
e-mail: [robin.rombach@iwr.uni-heidelberg.de](mailto:robin.rombach@iwr.uni-heidelberg.de)

P. Esser  
e-mail: [patrick.esser@iwr.uni-heidelberg.de](mailto:patrick.esser@iwr.uni-heidelberg.de)

A. Blattmann  
e-mail: [bjoern.ommer@iwr.uni-heidelberg.de](mailto:bjoern.ommer@iwr.uni-heidelberg.de)

B. Ommer  
e-mail: [andreas.blattmann@iwr.uni-heidelberg.de](mailto:andreas.blattmann@iwr.uni-heidelberg.de)

can yield abstract representations that are perfectly adapted feature encodings for the task at hand. However, their increasing abstraction capability and performance comes at the expense of a lack in interpretability [BBM+15]: although the network may solve a problem, it does not convey an understanding of its predictions or their causes, often leaving the impression of a black box [Mil19]. In particular, users are missing an explanation of semantic concepts that the model has learned to *represent* and of those it has learned to *ignore*, i.e., its invariances.

Providing such explanations and an understanding of network predictions and their causes is thus crucial for transparent AI. Not only is this relevant to discover limitations and promising directions for future improvements of the AI system itself, but also for compliance with legislation [GF17, Eur20], knowledge distillation from such a system [Lip18], and post hoc verification of the model [SWM17]. Consequently, research on interpretable deep models has recently gained a lot of attention, particularly methods that investigate latent representations to understand what the model has learned [SWM17, SZS+14, BZK+17, FV18, ERO20].

**Challenges and aims:** Assessing these latent representations is challenging due to two fundamental issues: (*i*) To achieve robustness and generalization despite noisy inputs and data variability, hidden layers exhibit a distributed coding of semantically meaningful concepts [FV18]. Attributing semantics to a single neuron via backpropagation [MLB+17] or synthesis [YCN+15] is thus impossible without altering the network [MSM18, ZKL+16], which typically degrades performance. (*ii*) End-to-end learning trains deep representations toward a goal task, making them invariant to features irrelevant for this goal. Understanding these characteristics that a representation has abstracted away is challenging, since we essentially need to portray features that have been discarded.

These challenges call for a method that can interpret existing network representations by recovering their invariances without modifying them. Given these recovered invariances, we seek an invertible mapping that translates a representation *and* the invariances onto understandable semantic concepts. The mapping disentangles the distributed encoding of the high-dimensional representation and its invariances by projecting them onto separate multi-dimensional factors that correspond to human-understandable semantic concepts. Both this translation and the recovering of invariances are implemented with invertible neural networks (INNs) [Red93, DSB17, KD18]. For the translation, this guarantees that the resulting understandable representation is equally expressive as the model representation combined with the recovered invariances (no information is lost). Its invertibility also warrants that feature modifications applied in the semantic domain correctly adjust the recovered representation.

**Our contributions:** Our contributions to a comprehensive understanding of deep representations are as follows: (*i*) We present an approach, which, by utilizing invertible neural networks, improves the understanding of representations produced by existing network architectures with no need for re-training or otherwise compromising their performance. (*ii*) Our generative approach is able to recover the invariances that result from the non-injective projection (of input onto a latent representation) which

deep networks typically learn. This model then provides a probabilistic visualization of the latent representation and its invariances. (iii) We bijectively translate an arbitrarily abstract representation and its invariances via a non-linear transformation into another representation of equal expressiveness, but with accessible semantic concepts. (iv) The invertibility also enables manipulation of the original latent representations in a semantically understandable manner, thus facilitating further diagnostics of a network.

## 2 Background

Two main approaches to interpretable AI can be identified, those which aim to incorporate interpretability directly into the design of models, and those which aim to provide interpretability to existing models [MSM18]. Approaches from the first category range from modifications of network architectures [ZKL+16], over regularization of models encouraging interpretability [LBMO19, PASC+20], toward combinations of both [ZNWZ18]. However, these approaches always involve a trade-off between model performance and model interpretability. Being of the latter category, our approach allows to interpret representations of existing models without compromising their performance.

To better understand what an existing model has learned, its representations must be studied [SWM17]. Szegedy et al. [SZS+14] show that both random directions and coordinate axes in the feature space of networks can represent semantic properties and conclude that they are not necessarily represented by individual neurons. Different works attempt to select groups of neurons which have a certain semantic meaning, such as based on scenes [ZKL+15], objects [SR15] and object parts [SRD14]. [BZK+17] studied the interpretability of neurons and found that a rotation of the representation space spanned by the neurons decreases its interpretability. While this suggests that the neurons provide a more interpretable basis compared to a random basis, [FV18] shows that the choice of basis is not the only challenge for interpretability of representations. Their findings demonstrate that learned representations are distributed, i.e., a single semantic concept is encoded by an activation pattern involving multiple neurons, and a single neuron is involved in the encoding of multiple different semantic concepts. Instead of selecting a set of neurons directly, [ERO20] learns an INN that transforms the original representation space to an interpretable space, where a single semantic concept is represented by a known group of neurons and a single neuron is involved in the encoding of just a single semantic concept. However, to interpret not only the representation itself but also its invariances, it is insufficient to transform only the representation itself. Our approach therefore transforms the latent representation space of an autoencoder, which has the capacity to represent its inputs faithfully, and subsequently translates a model representation and its invariances into this space for semantic interpretation and visualization.

A large body of works approach interpretability of existing networks based on visualizations. Selvaraju et al. [SCD+20] use gradients of network outputs

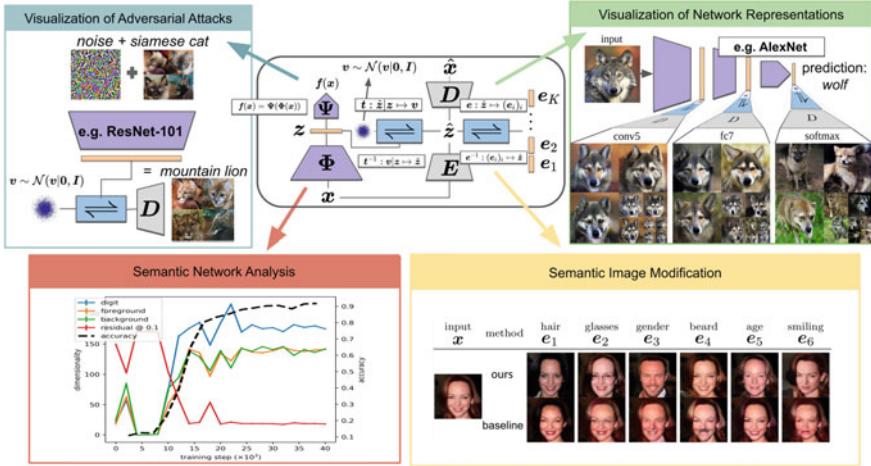
with respect to a convolutional layer to obtain coarse localization maps. Bach et al. [BBM+15] propose an approach to obtain pixel-wise relevance scores for a specific class of models which is generalized in [MLB+17]. To obtain richer visual interpretations, [ZF14, SVZ14, YCN+15, MV16] reconstruct images which maximally activate certain neurons. Nguyen et al. [NDY+16] use a generator network for this task, which was introduced in [DB16] for reconstructing images from their feature representation. Our key insight is that these existing approaches do not explicitly account for the invariances learned by a model. Invariances imply that feature inversion is a one-to-many mapping and thus they must be recovered to solve the task. Recently, [SGM+20] introduced a GAN-based approach that utilizes features of a pretrained classifier as a semantic pyramid for image generation. Nash et al. [NWKW19] used samples from an autoregressive model of images conditioned on a feature representation to gain insights into the representation’s invariances. In contrast, our approach recovers an explicit representation of the invariances, which can be recombined with modified feature representations, and thus makes the effect of modifications to representations, e.g., through adversarial attacks, visible.

Other works consider visual interpretations for specialized models. Santurkar et al. [SIT+19] showed that the quality of images which maximally activate certain neurons is significantly improved when activating neurons of an adversarially robust classifier. Bau et al. [BZS+19] explore the relationship between neurons and the images produced by a generative adversarial network. For the same class of models, [GAOI19] finds directions in their input space which represent semantic concepts corresponding to certain cognitive properties. Such semantic directions have previously also been found in classifier networks [UGP+17] but requires aligned data. All of these approaches require either special training of models, are limited to a very special class of models which already provide visualizations, or depend on special assumptions on model and data. In contrast, our approach can be applied to arbitrary models without re-training or modifying them, and provides both visualizations and semantic explanations, for both the model’s representation and its learned invariances.

### 3 Method

Common tasks of computer vision can be phrased as a mapping from an input image  $\mathbf{x}$  to some output  $f(\mathbf{x})$  such as a classification of the image, a regression (e.g., of object locations), a (semantic) segmentation map, or a re-synthesis that yields another image. Deep learning utilizes a hierarchy of intermediate network layers that gradually transform the input into increasingly more abstract representations. Let  $\mathbf{z} = \Phi(\mathbf{x}) \in \mathbb{R}^{N_z}$  be the representation extracted by one such layer (without loss of generality we consider  $\mathbf{z}$  to be an  $N_z$ -dim vector, flattening it if necessary) and  $f(\mathbf{x}) = \Psi(\mathbf{z}) = \Psi(\Phi(\mathbf{x}))$  the mapping onto the output.

An essential characteristic of a deep feature encoding  $\mathbf{z}$  is the increasing abstractness of higher feature encoding layers and the resulting reduction of information.



**Fig. 1** Proposed architecture. We provide post hoc interpretation for a given deep network  $f = \Psi \circ \Phi$ . For a deep representation  $z = \Phi(x)$  a conditional INN  $t$  recovers  $\Phi$ 's invariances  $v$  from a representation  $\hat{z}$  which contains entangled information about *both*  $z$  and  $v$ . The INN  $e$  then translates the representation  $\hat{z}$  into a factorized representation with accessible semantic concepts. This approach allows for various applications, including visualizations of network representations of natural (green box) and altered inputs (blue box), semantic network analysis (red box) and semantic image modifications (yellow box)

This reduction generally causes the feature encoding to become invariant to those properties of the input image, which do not provide salient information for the task at hand [CWG+18]. To explain a latent representation, we need to recover such invariances  $v$  and make  $z$  *and*  $v$  interpretable by learning a bijective mapping onto understandable semantic concepts, see Fig. 1. Section 3.1 describes our INN  $t$  to recover an encoding  $v$  of the invariances. Due to the generative nature of  $t$ , our approach can correctly sample visualizations of the model representation and its invariances without leaving the underlying data distribution and introducing artifacts. With  $v$  then available, Sect. 3.2 presents an INN  $e$  that translates  $t$ 's encoding of  $z$  and  $v$  without losing information onto disentangled semantic concepts. Moreover, the invertibility allows modifications in the semantic domain to correctly project back onto the original representation or into image space.

### 3.1 Recovering the Invariances of Deep Models

**Learning an encoding to help recover invariances:** Key to a deep representation is not only the information  $z$  captures, but also what it has learned to abstract away. To learn what  $z$  misses with respect to  $x$ , we need an encoding  $\hat{z}$ , which, in contrast to  $z$ , includes the invariances exhibited by  $z$ . Without making prior assumptions about

the deep model  $f$ , autoencoders provide a generic way to obtain such an encoding  $\hat{z}$ , since they ensure that their input  $x$  can be recovered from their learned representation  $\hat{z}$ , which hence also comprises the invariances.

Therefore, we learn an autoencoder with an encoder  $E$  that provides the data representation  $\hat{z} = E(x)$  and a decoder  $D$  producing the data reconstruction  $\hat{x} = D(\hat{z})$ . Section 3.2 will utilize the decoding from  $\hat{z}$  to  $\hat{x}$  to visualize both  $z$  and  $v$ . The autoencoder is trained to reconstruct its inputs by minimizing a perceptual metric between input and reconstruction,  $\|x - \hat{x}\|$ , as in [DB16]. The details of the architecture and training procedure can be found in Sect. 3.3, autoencoder  $E, D$ . It is crucial that the autoencoder only needs to be trained once on the training data. Consequently, the same  $E$  can be used to interpret different representations  $z$ , e.g., different models or layers within a model, thus ensuring fair comparisons between them. Moreover, the complexity of the autoencoder can be adjusted based on the computational needs, allowing us to work with much lower dimensional encodings  $\hat{z}$  compared to reconstructing the invariances directly from the images  $x$ . This reduces the computational demands of our approach significantly.

**Learning a conditional INN that recovers invariances:** Due to the reconstruction task of the autoencoder,  $\hat{z}$  not only contains the invariances  $v$ , but also the representation  $z$ . Thus, we must disentangle [EHO19, LSOL20, KSLO19]  $v$  and  $z$  using a mapping  $t(\cdot|z) : \hat{z} \mapsto v = t(\hat{z}|z)$  which, depending on  $z$ , extracts  $v$  from  $\hat{z}$ .

Besides extracting the invariances from a given  $\hat{z}$ ,  $t$  must also enable an inverse mapping from given model representations  $z$  to  $\hat{z}$  to support a further mapping onto semantic concepts Sect. 3.2 and visualization based on  $D(\hat{z})$ . There are many different  $x$  with  $\Phi(x) = z$ , namely, all those  $x$  which differ only in properties that  $\Phi$  is invariant to. Thus, there are also many different  $\hat{z}$  that this mapping must recover. Consequently, the mapping from  $z$  to  $\hat{z}$  is set-valued. However, to understand  $f$  we do not want to recover all possible  $\hat{z}$ , but only those which are likely under the training distribution of the autoencoder. In particular, this excludes unnatural images such as those obtained by DeepDream [MOT15] or adversarial attacks [SZS+14]. In conclusion, we need to sample  $\hat{z} \sim p(\hat{z}|z)$ .

To avoid a costly inversion process of  $\Phi$ ,  $t$  must be invertible (implemented as an INN) so that a change of variables

$$p(\hat{z}|z) = \frac{p(v|z)}{|\det \nabla(t^{-1})(v|z)|} , \text{ where } v = t(\hat{z}|z), \quad (1)$$

yields  $p(\hat{z}|z)$  by means of the distribution  $p(v|z)$  of invariances, given a model representation  $z$ . Here, the denominator denotes the absolute value of the determinant of Jacobian  $\nabla(t^{-1})$  of  $v \mapsto t^{-1}(v|z) = \hat{z}$ , which is efficient to compute for common invertible network architectures. Consequently, we obtain  $\hat{z}$  for given  $z$  by sampling from the invariant space  $v$  given  $z$  and then applying  $t^{-1}$ ,

$$\hat{z} \sim p(\hat{z}|z) \iff v \sim p(v|z), \hat{z} = t^{-1}(v|z). \quad (2)$$

Since  $\mathbf{v}$  is the invariant space for  $z$ , both are complementary thus implying independence  $p(\mathbf{v}|z) = p(\mathbf{v})$ . Because a powerful transformation  $\mathbf{t}^{-1}$  can transform between two arbitrary densities, we can assume without loss of generality a Gaussian prior  $p(\mathbf{v}) = \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I})$ , where  $\mathbf{I}$  is the identity matrix. Given this prior, our task is then to learn the transformation  $\mathbf{t}$  that maps  $\mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I})$  onto  $p(\hat{z}|z)$ . To this end, we maximize the log-likelihood of  $\hat{z}$  given  $z$ , which results in a per-example loss of

$$J_e(\hat{z}, z) = -\log p(\hat{z}|z) = -\log \mathcal{N}(\mathbf{t}(\hat{z}|z)|\mathbf{0}, \mathbf{I}) - \log |\det \nabla \mathbf{t}(\hat{z}|z)|. \quad (3)$$

Minimizing this loss over the training data distribution  $p(\mathbf{x})$  gives  $\mathbf{t}$ , a bijective mapping between  $\hat{z}$  and  $(z, \mathbf{v})$ ,

$$J(\mathbf{t}) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [J_e(\mathbf{E}(\mathbf{x}), \Phi(\mathbf{x}))] \quad (4)$$

$$= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[ \frac{1}{2} \|\mathbf{t}(\mathbf{E}(\mathbf{x})|\Phi(\mathbf{x}))\|^2 + N_{\hat{z}} \log 2\pi - \log |\det \nabla \mathbf{t}(\mathbf{E}(\mathbf{x})|\Phi(\mathbf{x}))| \right]. \quad (5)$$

Note that both  $\mathbf{E}$  and  $\Phi$  remain fixed during minimization of  $J$ .

### 3.2 Interpreting Representations and Their Invariances

**Visualizing representations and invariances:** For an image representation  $z = \Phi(\mathbf{x})$ , (2) presents an efficient approach (a single forward pass through the INN  $\mathbf{t}$ ) to sample an encoding  $\hat{z}$ , which is a combination of  $z$  with a particular realization of its invariances  $\mathbf{v}$ . Sampling multiple realizations of  $\hat{z}$  for a given  $z$  highlight what remains constant and what changes due to different  $\mathbf{v}$ : information preserved in the representation  $z$  remains constant over different samples and information discarded by the model ends up in the invariances  $\mathbf{v}$  and shows changes over different samples. Visualizing the samples  $\hat{z} \sim p(\hat{z}|z)$  with  $\hat{\mathbf{x}} = \mathbf{D}(\hat{z})$  portrays this constancy and changes due to different  $\mathbf{v}$ . To complement this visualization, in the following, we learn a transformation of  $\hat{z}$  into a semantically meaningful representation which allows to uncover the semantics captured by  $z$  and  $\mathbf{v}$ .

**Learning an INN to produce semantic interpretations:** The autoencoder representation  $\hat{z}$  is an equivalent representation of  $(z, \mathbf{v})$  but its feature dimensions do not necessarily correspond to semantic concepts [FV18]. More generally, without supervision, we cannot reliably discover semantically meaningful, explanatory factors of  $\hat{z}$  [LBL+19]. In order to explain  $\hat{z}$  in terms of given semantic concepts, we apply the approach of [ERO20] and learn a bijective transformation of  $\hat{z}$  to an interpretable representation  $\mathbf{e}(\hat{z})$  where different groups of components, called factors, correspond to semantic concepts.

To learn the transformation  $\mathbf{e}$ , we parameterize  $\mathbf{e}$  by an INN and assume that semantic concepts are defined implicitly by pairs of images, i.e., for each semantic

concept we have access to training pairs  $\mathbf{x}^a, \mathbf{x}^b$  that have the respective concept in common. For example, the semantic concept “smiling” is defined by pairs of images, where either both images show smiling persons or both images show non-smiling persons. Applying this formulation, input pairs which are similar in a certain semantic concept are similar in the corresponding factor of the interpretable representation  $\mathbf{e}(\hat{\mathbf{z}})$ .

Following [ERO20], the loss for training the invertible network  $\mathbf{e}$  is then given by

$$\begin{aligned} J(\mathbf{e}) = & \mathbb{E}_{\mathbf{x}^a, \mathbf{x}^b} \left[ -\log p(\mathbf{e}(\mathbf{E}(\mathbf{x}^a)), \mathbf{e}(\mathbf{E}(\mathbf{x}^b))) \right. \\ & \left. - \log |\det \nabla \mathbf{e}(\mathbf{E}(\mathbf{x}^a))| - \log |\det \nabla \mathbf{e}(\mathbf{E}(\mathbf{x}^b))| \right]. \end{aligned} \quad (6)$$

**Interpretation by applying the learned INNs:** After training, the combination of  $\mathbf{e}$  with  $\mathbf{t}$  from Sect. 3.1 provides semantic interpretations given a model representation  $\mathbf{z}$ : (2) gives realizations of the invariances  $\mathbf{v}$  which are combined with  $\mathbf{z}$  to produce  $\hat{\mathbf{z}} = \mathbf{t}^{-1}(\mathbf{v}|\mathbf{z})$ . Then  $\mathbf{e}$  transforms  $\hat{\mathbf{z}}$  without loss of information into a semantically accessible representation  $(\mathbf{e}_i)_i = \mathbf{e}(\hat{\mathbf{z}}) = \mathbf{e}(\mathbf{t}^{-1}(\mathbf{v}|\mathbf{z}))$  consisting of different semantic factors  $\mathbf{e}_i$ . Comparing the  $\mathbf{e}_i$  for different model representations  $\mathbf{z}$  and invariances  $\mathbf{v}$  allows us to observe which semantic concepts the model representation  $\mathbf{z} = \Phi(\cdot)$  is sensitive to, and which it is invariant to.

**Semantic Modifications of Latent Representations:** The transformations  $\mathbf{t}^{-1}$  and  $\mathbf{e}$  not only interpret a representation  $\mathbf{z}$  in terms of accessible semantic concepts  $(\mathbf{e}_i)_i$ . Given  $\mathbf{v} \sim p(\mathbf{v})$ , they also allow to modify  $\hat{\mathbf{z}} = \mathbf{t}^{-1}(\mathbf{v}|\mathbf{z})$  in a semantically meaningful manner by altering its corresponding  $(\mathbf{e}_i)_i$  and then applying the inverse translation  $\mathbf{e}^{-1}$ ,

$$\hat{\mathbf{z}} \xrightarrow{\mathbf{e}} (\mathbf{e}_i) \xrightarrow{\text{modification}} (\mathbf{e}_i^*) \xrightarrow{\mathbf{e}^{-1}} \hat{\mathbf{z}}^*. \quad (7)$$

The modified representation  $\hat{\mathbf{z}}^*$  is then readily transformed back into image space  $\hat{\mathbf{x}}^* = \mathbf{D}(\hat{\mathbf{z}}^*)$ . Besides visual interpretation of the modification,  $\hat{\mathbf{x}}^*$  can be fed into the model  $\Psi(\Phi(\hat{\mathbf{x}}^*))$  to probe for sensitivity to certain semantic concepts.

### 3.3 Implementation Details

In this section, we provide implementation details about the exact training procedure and architecture of all components of our approach. This is only for the sake of clarity and completeness. For readers who are already familiar with INNs or people rather interested in the higher level ideas of our approach than in its technical details, this section can be safely skipped.

**Autoencoder  $\mathbf{E}, \mathbf{D}$ :** In Sect. 3.1, we introduced an autoencoder to obtain a representation  $\hat{\mathbf{z}}$  of  $\mathbf{x}$ , which includes the invariances abstracted away by a given model representation  $\mathbf{z}$ . This autoencoder consists of an encoder  $E(\mathbf{x})$  and a decoder  $D(\hat{\mathbf{z}})$ .

**Table 1** Autoencoder architecture on datasets with images of resolution  $28 \times 28$ 

Encoder: FC and Norm refer to fully connected layer and batch normalization, respectively. The term “down” denotes downsampling by factor 2. Leaky ReLU (LReLU) uses a slope parameter of 0.2.

RGB image $\mathbf{x} \in \mathbb{I}^{28 \times 28 \times 3}$ , $\mathbb{I} = [0, 1]$
Conv down, Norm, LReLU $\rightarrow \mathbb{R}^{14 \times 14 \times 64}$
Conv down, Norm, LReLU $\rightarrow \mathbb{R}^{7 \times 7 \times 128}$
FC $\mapsto (\boldsymbol{\mu}, \boldsymbol{\sigma}\boldsymbol{\sigma}^T) \in \mathbb{R}^{64} \times \mathbb{R}^{64}$

Decoder: FC and Norm refer to fully connected layer and batch normalization, respectively. Leaky ReLU (LReLU) uses a slope parameter of 0.2.

$\hat{\mathbf{z}} \in \mathbb{R}^{64} \sim \mathcal{N}(\boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}\boldsymbol{\sigma}^T))$
FC $\rightarrow \mathbb{R}^{7 \times 7 \times 128}$
Conv transpose, Norm, LReLU $\rightarrow \mathbb{R}^{14 \times 14 \times 64}$
Conv transpose, Tanh $\rightarrow \mathbb{I}^{28 \times 28 \times 3}$ , $\mathbb{I} = [0, 1]$

Because the INNs  $\mathbf{t}$  and  $\mathbf{e}$  transform the distribution of  $\hat{\mathbf{z}}$ , we must ensure a strictly positive density for  $\hat{\mathbf{z}}$  to avoid degenerate solutions. This is readily achieved with a stochastic encoder, i.e., we predict mean  $\mathbf{E}(\mathbf{x})_{\boldsymbol{\mu}}$  and diagonal  $\mathbf{E}(\mathbf{x})_{\boldsymbol{\sigma}^2}$  of a Gaussian distribution, and obtain the desired representation as  $\hat{\mathbf{z}} \sim \mathcal{N}(\hat{\mathbf{z}} | \mathbf{E}(\mathbf{x})_{\boldsymbol{\mu}}, \text{diag}(\mathbf{E}(\mathbf{x})_{\boldsymbol{\sigma}^2}))$ . Following [DW19], we train this autoencoder as a variational autoencoder using the reparameterization trick [KW14, RMW14] to match the encoded distribution to a standard normal distribution, and jointly learn the scalar output variance  $\gamma$  under an image metric  $\|\mathbf{x} - \hat{\mathbf{x}}\|$  to avoid blurry reconstructions. The resulting loss function is thus

$$\begin{aligned} J(\mathbf{E}, \mathbf{D}, \gamma) = & \mathbb{E}_{\substack{\mathbf{x} \sim p(\mathbf{x}) \\ \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon} | 0, \mathbf{I})}} \left[ \frac{1}{\gamma} \|\mathbf{x} - \mathbf{D}(\mathbf{E}(\mathbf{x})_{\boldsymbol{\mu}} + \boldsymbol{\epsilon} \sqrt{\text{diag}(\mathbf{E}(\mathbf{x})_{\boldsymbol{\sigma}^2})})\| + \log \gamma \right. \\ & \left. + \frac{1}{2} \sum_{i=1}^{N_z} \{(\mathbf{E}(\mathbf{x})_{\boldsymbol{\mu}})_i^2 + (\mathbf{E}(\mathbf{x})_{\boldsymbol{\sigma}^2})_i - 1 - \log(\mathbf{E}(\mathbf{x})_{\boldsymbol{\sigma}^2})_i\} \right]. \end{aligned} \quad (8)$$

Note that  $\sqrt{(\cdot)}$  and  $\log(\cdot)$  on multi-dimensional entities are applied element-wise. In the experiments shown in this chapter, we use images of spatial resolutions  $28 \times 28$  and  $128 \times 128$ , resulting in different architectures for the autoencoder, summarized in Tables 1 and 2, respectively. For the encoder  $\mathbf{E}$  processing images of spatial resolution  $128 \times 128$ , we use an architecture based on ResNet-101 [HZRS16], and for the corresponding decoder  $\mathbf{D}$  we use an architecture based on BigGAN [BDS19], where we include a small fully connected network to replace the class conditioning used in BigGAN by a conditioning on  $\hat{\mathbf{z}}$ .

In the  $28 \times 28$  case, we use a squared  $L_2$  loss for the image metric, which corresponds to the first term in (8). For our  $128 \times 128$ -models, we further use an improved metric as in [DB16], which includes additional perceptual [ZIE+18] and discriminator losses. The perceptual loss consists of  $L_1$  feature distances obtained from different

**Table 2** Autoencoder architecture for datasets with images of resolution  $128 \times 128$ 

Encoder based on Resnet-101: Bottleneck, FC, and Norm refer to bottleneck residual unit, fully connected layer, and batch normalization, respectively. The term “down” denotes downsampling by factor 2.

Decoder based on BigGAN: Embed, FC, Norm, and ResBlock refer to embedding layers, fully connected layers, batch normalization, and residual block, respectively. The term “up” denotes upsampling by factor 2. Leaky ReLU (LReLU) uses a slope parameter of 0.2.

RGB image $x \in \mathbb{I}^{128 \times 128 \times 3}$ , $\mathbb{I} = [0, 1]$
Conv down $\rightarrow \mathbb{R}^{64 \times 64 \times 64}$
Norm, ReLU, MaxPool $\rightarrow \mathbb{R}^{32 \times 32 \times 64}$
$3 \times$ BottleNeck $\rightarrow \mathbb{R}^{32 \times 32 \times 256}$
$4 \times$ BottleNeck down $\rightarrow \mathbb{R}^{16 \times 16 \times 512}$
$23 \times$ BottleNeck down $\rightarrow \mathbb{R}^{8 \times 8 \times 1024}$
$3 \times$ BottleNeck down $\rightarrow \mathbb{R}^{4 \times 4 \times 2048}$
AvgPool, FC $\mapsto (\mu, \sigma\sigma^T) \in \mathbb{R}^{128} \times \mathbb{R}^{128}$

$\hat{z} \in \mathbb{R}^{128} \sim \mathcal{N}(\mu, \text{diag}(\sigma\sigma^T))$
$3 \times (\text{FC}, \text{LReLU}) \rightarrow \mathbb{R}^{256}$
FC, Softmax $\rightarrow \mathbb{R}^{1000}$
Embed $\mapsto h \in \mathbb{R}^{128}$
FC( $\hat{z}$ ) $\rightarrow \mathbb{R}^{4 \times 4 \times 16 \cdot 96}$
ResBlock( $\hat{z}, h$ ) up $\rightarrow \mathbb{R}^{8 \times 8 \times 16 \cdot 96}$
ResBlock( $\hat{z}, h$ ) up $\rightarrow \mathbb{R}^{16 \times 16 \times 8 \cdot 96}$
ResBlock( $\hat{z}, h$ ) up $\rightarrow \mathbb{R}^{32 \times 32 \times 4 \cdot 96}$
ResBlock( $\hat{z}, h$ ) up $\rightarrow \mathbb{R}^{64 \times 64 \times 2 \cdot 96}$
Non-Local Block $\rightarrow \mathbb{R}^{64 \times 64 \times 2 \cdot 96}$
ResBlock( $\hat{z}, h$ ) up $\rightarrow \mathbb{R}^{64 \times 64 \times 96}$
Norm, ReLU, Conv up $\rightarrow \mathbb{R}^{128 \times 128 \times 3}$
Tanh $\mapsto \hat{x} \in \mathbb{I}^{128 \times 128 \times 3}$ , $\mathbb{I} = [0, 1]$

layers of a fixed, pretrained network. We use a VGG-16 network pretrained on ImageNet and weighted distances of different layers as in [ZIE+18]. The discriminator is trained along with the autoencoder to distinguish reconstructed images from real images using a binary classification loss, and the autoencoder maximizes the log-probability that reconstructed images are classified as real images. The architectures of VGG-16 and the discriminator are summarized in Table 3.

**Details on the INNe for Revealing Semantics of Deep Representations:** Previous works have successfully applied INNs for density estimation [DSB17], inverse problems [AKW+19], and on top of autoencoder representations [ERO20, XYA19, DMB+21, BMDO21b, BMDO21a] for a wide range of applications such as video synthesis [DMB+21, BMDO21b] and translation between pretrained, powerful networks [REO20]. This section provides details on how we embed the approach of [ERO20] to reveal the semantic concepts of autoencoder representations  $\hat{z}$ , cf. Sect. 3.2.

Since we will never have examples for all relevant semantic concepts, we include a residual concept that captures the remaining variability of  $\hat{z}$ , which is not explained by the given semantic concepts.

Following [ERO20], we learn a bijective transformation  $e(\hat{z})$ , which translates the non-interpretable representation  $\hat{z}$  invertibly into a factorized representation

**Table 3** Architectures used to compute image metrics for the autoencoder, which were used for training the autoencoder  $E$ ,  $D$  on datasets with images of resolution  $128 \times 128$ .  
 VGG-16 pretrained on ImageNet for feature extraction. Output of bold layers are used to compute feature distances.

RGB image $\mathbf{x} \in \mathbb{R}^{128 \times 128 \times 3}$ $2 \times \mathbf{Conv, ReLU} \rightarrow \mathbb{R}^{128 \times 128 \times 64}$ $\text{MaxPool} \rightarrow \mathbb{R}^{64 \times 64 \times 64}$ $2 \times \mathbf{Conv, ReLU} \rightarrow \mathbb{R}^{64 \times 64 \times 128}$ $\text{MaxPool} \rightarrow \mathbb{R}^{32 \times 32 \times 128}$ $3 \times \mathbf{Conv, ReLU} \rightarrow \mathbb{R}^{32 \times 32 \times 256}$ $\text{MaxPool} \rightarrow \mathbb{R}^{16 \times 16 \times 256}$ $3 \times \mathbf{Conv, ReLU} \rightarrow \mathbb{R}^{16 \times 16 \times 512}$ $\text{MaxPool} \rightarrow \mathbb{R}^{8 \times 8 \times 512}$ $3 \times \mathbf{Conv, ReLU} \rightarrow \mathbb{R}^{8 \times 8 \times 512}$	Discriminator: All convolutions use kernel size 4. Norm refers to batch normalization, leaky ReLU uses a slope parameter 0.2. RGB image $\mathbf{x} \in \mathbb{R}^{128 \times 128 \times 3}$ $\text{Conv down, LReLU} \rightarrow \mathbb{R}^{64 \times 64 \times 64}$ $\text{Conv down, Norm, LReLU} \rightarrow \mathbb{R}^{32 \times 32 \times 128}$ $\text{Conv down, Norm, LReLU} \rightarrow \mathbb{R}^{16 \times 16 \times 256}$ $\text{Conv down, Norm, LReLU} \rightarrow \mathbb{R}^{8 \times 8 \times 512}$ $\text{Conv, Norm, LReLU} \rightarrow \mathbb{R}^{8 \times 8 \times 512}$ $\text{Conv} \rightarrow \mathbb{R}^{8 \times 8 \times 1}$
--	--

$(\mathbf{e}_i(\hat{\mathbf{z}}))_{i=0}^K = \mathbf{e}(\hat{\mathbf{z}})$ , where each factor  $\mathbf{e}_i \in \mathbb{R}^{N_{e_i}}$  represents one of the given semantic concepts for  $i = 1, \dots, K$ , and  $\mathbf{e}_0 \in \mathbb{R}^{N_{e_0}}$  is the residual concept.

The INN  $\mathbf{e}$  establishes a one-to-one correspondence between an encoding and different semantic concepts and, conversely, enables semantic modifications to correctly alter the original encoding (see next section). Being an INN,  $\mathbf{e}(\hat{\mathbf{z}})$  and  $\hat{\mathbf{z}}$  need to have the same dimensionality and we set  $N_{e_0} = N_{\hat{\mathbf{z}}} - \sum_{i=1}^K N_{e_i}$ . We denote the indices of concept  $i$  with respect to  $\mathbf{e}(\hat{\mathbf{z}})$  as  $\mathcal{I}_i \subset \{1, \dots, N_{\hat{\mathbf{z}}}\}$  such that we can write  $\mathbf{e}_i = (\mathbf{e}(\hat{\mathbf{z}})_k)_{k \in \mathcal{I}_i}$ .

In the following, we emphasize on deriving a loss function for training the semantic INN. Let  $\mathbf{e}_i$  be the factor representing some semantic concept, e.g., gender, that the contents of two images  $\mathbf{x}^a, \mathbf{x}^b$  share. Then the projection of their encodings  $\hat{\mathbf{z}}^a, \hat{\mathbf{z}}^b$  onto this semantic concept must be similar [ERO20, KWKT15],

$$\mathbf{e}_i(\hat{\mathbf{z}}^a) \simeq \mathbf{e}_i(\hat{\mathbf{z}}^b) \quad \text{where } \hat{\mathbf{z}}^a = E(\mathbf{x}^a), \hat{\mathbf{z}}^b = E(\mathbf{x}^b). \quad (9)$$

Moreover, to interpret  $\hat{\mathbf{z}}$  we are interested in the separate contribution of different semantic concepts  $\mathbf{e}_i$  that explain  $\hat{\mathbf{z}}$ . Hence, we seek a mapping  $\mathbf{e}(\cdot)$  that strives to disentangle different concepts,

$$\mathbf{e}_i(\hat{\mathbf{z}}) \perp \mathbf{e}_j(\hat{\mathbf{z}}) \quad \forall i \neq j, \mathbf{x} \quad \text{where } \hat{\mathbf{z}} = E(\mathbf{x}). \quad (10)$$

The objectives in (9), (10) imply a correlation in  $\mathbf{e}_i$  for pairs  $\hat{\mathbf{z}}^a$  and  $\hat{\mathbf{z}}^b$  and no correlation between concepts  $\mathbf{e}_i, \mathbf{e}_j$  for  $i \neq j$ . This calls for a Gaussian distribution with a covariance matrix that reflects these requirements.

Let  $\mathbf{e}^a = (\mathbf{e}_i^a) = (\mathbf{e}_i(\mathbf{E}(\mathbf{x}^a)))$  and  $\mathbf{e}^b$  likewise, where  $\mathbf{x}^a, \mathbf{x}^b$  are samples from a training distribution  $p(\mathbf{x}^a, \mathbf{x}^b)$  for the  $i$ th semantic concept. The distribution of pairs  $\mathbf{e}^a$  and  $\mathbf{e}^b$  factorizes into a conditional and a marginal,

$$p(\mathbf{e}^a, \mathbf{e}^b) = p(\mathbf{e}^b | \mathbf{e}^a) p(\mathbf{e}^a). \quad (11)$$

Objective (10) implies a diagonal covariance for the marginal distribution  $p(\mathbf{e}^a)$ , i.e., a standard normal distribution, and (9) entails a correlation between  $\mathbf{e}_i^a$  and  $\mathbf{e}_i^b$ . Therefore, the correlation matrix is  $\Sigma^{ab} = \rho \operatorname{diag}((\delta_{\mathcal{I}_i}(k))_{k=1}^{N_z})$ , where

$$\delta_{\mathcal{I}_i}(k) = \begin{cases} 1 & \text{if } k \in \mathcal{I}_i, \\ 0 & \text{else.} \end{cases}$$

By symmetry,  $p(\mathbf{e}^b) = p(\mathbf{e}^a)$ , which gives

$$p(\mathbf{e}^b | \mathbf{e}^a) = \mathcal{N}(\mathbf{e}^b | \Sigma^{ab} \mathbf{e}^a, \mathbf{I} - (\Sigma^{ab})^2). \quad (12)$$

Inserting (12) and a standard normal distribution for  $p(\mathbf{e}^a)$  into (11) yields the negative log-likelihood for a pair  $\mathbf{e}^a, \mathbf{e}^b$ .

Given pairs  $\mathbf{x}^a, \mathbf{x}^b$  as training data, another change of variables from  $\hat{\mathbf{z}}^a = \mathbf{E}(\mathbf{x}^a)$  to  $\mathbf{e}^a = \mathbf{e}(\hat{\mathbf{z}}^a)$  gives the training loss function for  $\mathbf{e}$  as the negative log-likelihood of  $\hat{\mathbf{z}}^a, \hat{\mathbf{z}}^b$ ,

$$\begin{aligned} J(\mathbf{e}) = \mathbb{E}_{\mathbf{x}^a, \mathbf{x}^b} & \left[ -\log p(\mathbf{e}(\mathbf{E}(\mathbf{x}^a)), \mathbf{e}(\mathbf{E}(\mathbf{x}^b))) \right. \\ & \left. - \log |\det \nabla \mathbf{e}(\mathbf{E}(\mathbf{x}^a))| - \log |\det \nabla \mathbf{e}(\mathbf{E}(\mathbf{x}^b))| \right]. \end{aligned} \quad (13)$$

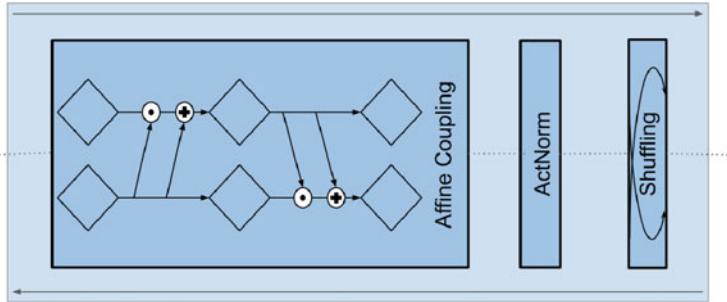
For simplicity, we have derived the loss for a single semantic concept  $\mathbf{e}_i$ . Simply summing over the losses of different semantic concepts yields their joint loss function and allows us to learn a joint translator  $\mathbf{e}$  for all of them.

In the following, we focus on the log-likelihood of pairs. The loss for  $\mathbf{e}$  in (13) contains the log-likelihood of pairs  $\mathbf{e}^a, \mathbf{e}^b$ . Inserting (12) and a standard normal distribution for  $p(\mathbf{e}^a)$  into (11) yields

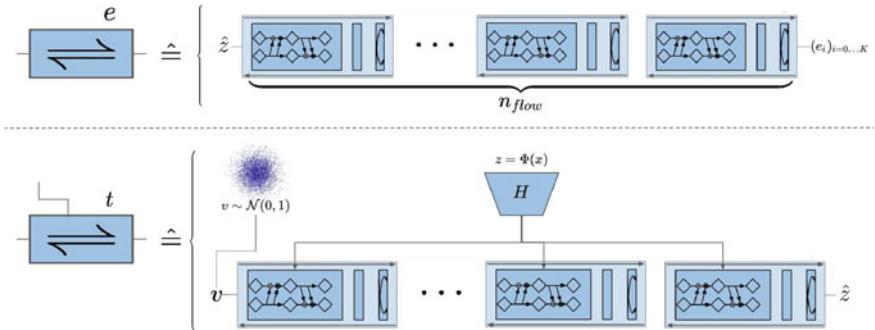
$$-\log p(\mathbf{e}^a, \mathbf{e}^b) = \frac{1}{2} \left( \sum_{k \in \mathcal{I}_i} \frac{(\mathbf{e}_k^b - \rho \mathbf{e}_k^a)^2}{1 - \rho^2} + \sum_{k \in \mathcal{I}_i^c} (\mathbf{e}_k^b)^2 + \sum_{k=1}^{N_z} (\mathbf{e}_k^a)^2 \right) + C, \quad (14)$$

where  $C = C(\rho, N_z)$  is a constant that can be ignored for the optimization process.  $\rho \in (0, 1)$  determines the relative importance of loss terms corresponding to the similarity requirement in (9) and the independence requirement in (10). We use a fixed value of  $\rho = 0.9$  for all experiments.

In the following, we describe the architecture of the semantic INN. In our implementation,  $\mathbf{e}$  is built by stacking invertible blocks, see Fig. 2, which consist of three



**Fig. 2** A single invertible block used to build our invertible neural networks



**Fig. 3** Architectures of our INN models. *top:* The semantic INN  $e$  consists of stacked invertible blocks. *bottom:* The conditional INN  $t$  is composed of a embedding module  $H$  that downsamples (upsamples if necessary) a given model representation  $\mathbf{h} = H(z) = H(\Phi(x))$ . Subsequently,  $\mathbf{h}$  is concatenated to the inputs of each block of the invertible model

invertible layers: coupling blocks [DSB17], actnorm layers [KD18], and shuffling layers. The final output is split into the factors  $(\mathbf{e}_i)$ , see Fig. 3.

Coupling blocks split their input  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$  along the channel dimension and use fully connected neural networks  $s_i$  and  $\tau_i$  to perform the following computation:

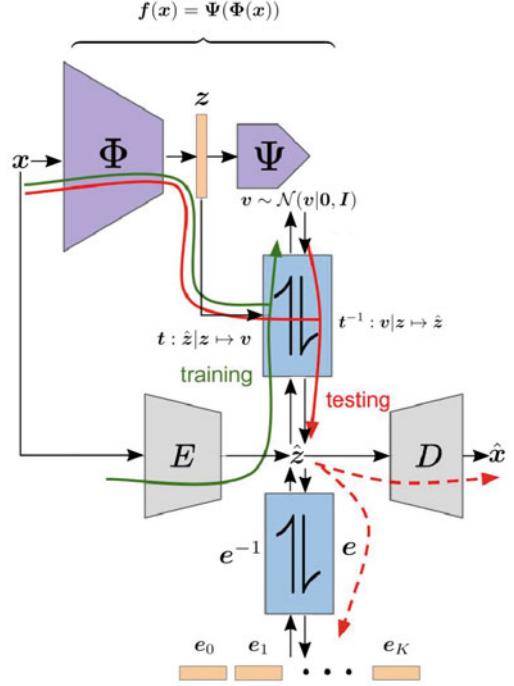
$$\tilde{\mathbf{x}}_1 = \mathbf{x}_1 \odot s_1(\mathbf{x}_2) + \tau_1(\mathbf{x}_2), \quad (15)$$

$$\tilde{\mathbf{x}}_2 = \mathbf{x}_2 \odot s_2(\tilde{\mathbf{x}}_1) + \tau_2(\tilde{\mathbf{x}}_1), \quad (16)$$

with the element-wise multiplication operator  $\odot$ . Actnorm layers consist of learnable shift and scale parameters for each channel, which are initialized to ensure activations with zero mean and unit variance on the first training batch. Shuffling layers use a fixed, randomly initialized permutation to shuffle the channels of its input, which provides a better mixing of channels for subsequent coupling layers.

**Conditional INN  $t$  for recovering invariances of deep representations:** We first elaborate on the architecture of the conditional INN. We build the conditional invert-

**Fig. 4** Graphical distinction of information flow during training and inference. During training of  $t$ , the encoder  $E$  provides an (approximately complete) data representation, which is used to learn the invariances of a given model’s representations  $z$ . At inference, the encoder is not necessarily needed anymore: given a representation  $z = \Phi(x)$ , invariances can be sampled from the prior distribution and decoded into data space through  $t^{-1}$



ible neural network  $t$  by expanding the semantic model  $e$  as follows: Given a model representation  $z$ , which is used as the conditioning of the INN, we first calculate its embedding

$$\mathbf{h} = \mathbf{H}(z) \quad (17)$$

which is subsequently fed into the affine coupling block:

$$\tilde{\mathbf{x}}_1 = \mathbf{x}_1 \odot s_1(\mathbf{x}_2, \mathbf{h}) + \tau_1(\mathbf{x}_2, \mathbf{h}), \quad (18)$$

$$\tilde{\mathbf{x}}_2 = \mathbf{x}_2 \odot s_2(\tilde{\mathbf{x}}_1, \mathbf{h}) + \tau_2(\tilde{\mathbf{x}}_1, \mathbf{h}), \quad (19)$$

with  $\odot$  again being an element-wise multiplication operator, where  $s_i$  and  $\tau_i$  are modified from (16) such that they are capable of processing a concatenated input  $(\mathbf{x}_i, \mathbf{h})$ . The embedding module  $\mathbf{H}$  is usually a shallow convolutional neural network used to down-/upsample a given model representation  $z$  to a size that the networks  $s_i$  and  $\tau_i$  are able to process. This means that  $t$ , analogous to  $e$ , consists of stacked invertible blocks, where each block is composed of coupling blocks, actnorm layers, and shuffling layers, cf. Sect. 3.3, details on the INN  $e$  for revealing semantics of deep representations, and Fig. 2. The complete architectures of both  $t$  and  $e$  are depicted in Fig. 3. Additionally, Fig. 4 provides a graphical distinction of the training and testing process of  $t$ . During training, the autoencoder  $D \circ E$  provides a representation of the

data that contains both the invariances and the representation of some model w.r.t. the input  $\mathbf{x}$ . After training of  $t$ , the encoder may be discarded and visual decodings and/or semantic interpretations of a model representation  $\mathbf{z}$  can be obtained by sampling and transforming  $\mathbf{v}$  as described in (2).

## 4 Experiments

To explore the applicability of our approach, we conduct experiments on several models: SqueezeNet [IHM+16], which provides lightweight classification, FaceNet [SKP15], a baseline for face recognition and clustering, trained on the VGGFace2 dataset [CSX+18], and variants of ResNet [HZRS16], a popular architecture often used when fine-tuning a classifier on a specific task and dataset.

Experiments are conducted on the following datasets: CelebA [LLWT15], AnimalFaces [LHM+19], Animals (containing carnivorous animals), ImageNet [DDS+09], and ColorMNIST, which is an augmented version of the MNIST dataset [LCB98], where both background and foreground have random, independent colors. Evaluation details follow in Sect. 4.5.

### 4.1 Comparison to Existing Methods

A key insight of our chapter is that reconstructions from a given model’s representation  $\mathbf{z} = \Phi(\mathbf{x})$  are impossible if the invariances the model has learned are not considered. In Fig. 5, we compare our approach to existing methods that either try to reconstruct the image via gradient-based optimization [MV16] or by training a reconstruction network directly on the representations  $\mathbf{z}$  [DB16]. By conditionally sampling images  $\hat{\mathbf{x}} = \mathbf{D}(\hat{\mathbf{z}})$ , where we obtain  $\hat{\mathbf{z}}$  via the INN  $t$  as described in (2) based on the invariances  $\mathbf{v} \sim p(\mathbf{v}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ , we bypass this shortcoming and obtain natural images without artifacts for any layer depth. The increased image quality is further confirmed by the Fréchet inception distance (FID) scores [HRU+17] reported in Table 4.

### 4.2 Understanding Models

**Interpreting a face recognition model:** FaceNet [SKP15] is a widely accepted baseline in the field of face recognition. This model embeds input images of human faces into a latent space where similar images have a small  $L_2$ -distance. We aim to understand the process of face recognition within this model by analyzing and visualizing learned invariances for several layers explicitly; see Table 6 for a detailed breakdown of the various layers of FaceNet. For the experiment, we use a pre-

Method	Reconstructions $\hat{x}$ from representations $z = \Phi(x)$ of different layers				
	input	conv5	fc6	fc7	fc8
<b>ours</b>					
[DB16]					
[MV16]					

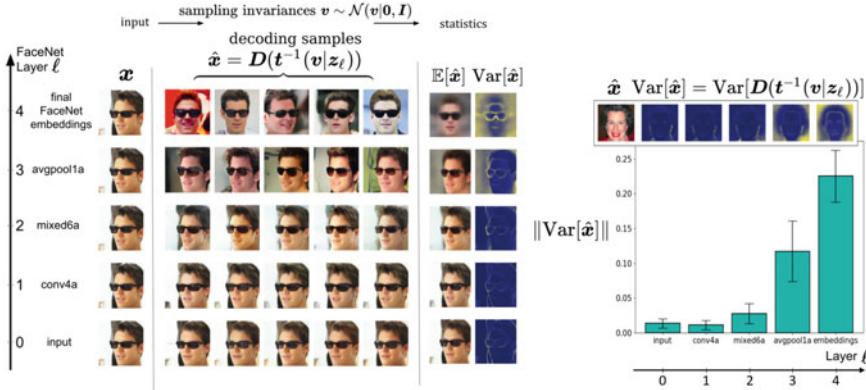
**Fig. 5** Comparison to existing network inversion methods for AlexNet [KSH12]. In contrast to the methods of [DB16] (D&B) and [MV16] (M&V), our invertible method explicitly samples the invariances of  $\Phi$  w.r.t. the data, which circumvents a common cause for artifacts and produces natural images independent of the depth of the layer which is reconstructed

**Table 4** FID scores for layer visualizations of AlexNet, obtained with our method and [DB16] (D&B). Scores are calculated on the Animals dataset

Layer	conv5	fc6	fc7	fc8	Output
<b>ours</b>	<b><math>23.6 \pm 0.5</math></b>	<b><math>24.3 \pm 0.7</math></b>	<b><math>24.9 \pm 0.4</math></b>	<b><math>26.4 \pm 0.4</math></b>	<b><math>27.4 \pm 0.3</math></b>
D&B	25.2	<b>24.9</b>	27.2	36.1	352.6

trained FaceNet and train the generative model presented in (2) by conditioning on various layers. Figure 6 depicts the amount of variance present in each selected layer when generating  $n = 250$  samples for each of the 100 different input images. This variance serves as a proxy for the amount of abstraction capability FaceNet has learned in its respective layers: more abstract representations allow for a rich variety of corresponding synthesized images, resulting in a large variance in image space when being decoded. We observe an approximate exponential growth of learned invariances with increasing layer depth, suggesting abstraction mainly happens in the deepest layers of the network. Furthermore, we are able to synthesize images that correspond to the given model representation for each selected layer.

**How does relevance of different concepts emerge during training?** Humans tend to provide explanations of entities by describing them in terms of their semantics, e.g., size or color. In a similar fashion, we want to semantically understand how a network (here: SqueezeNet) learns to solve a given problem.



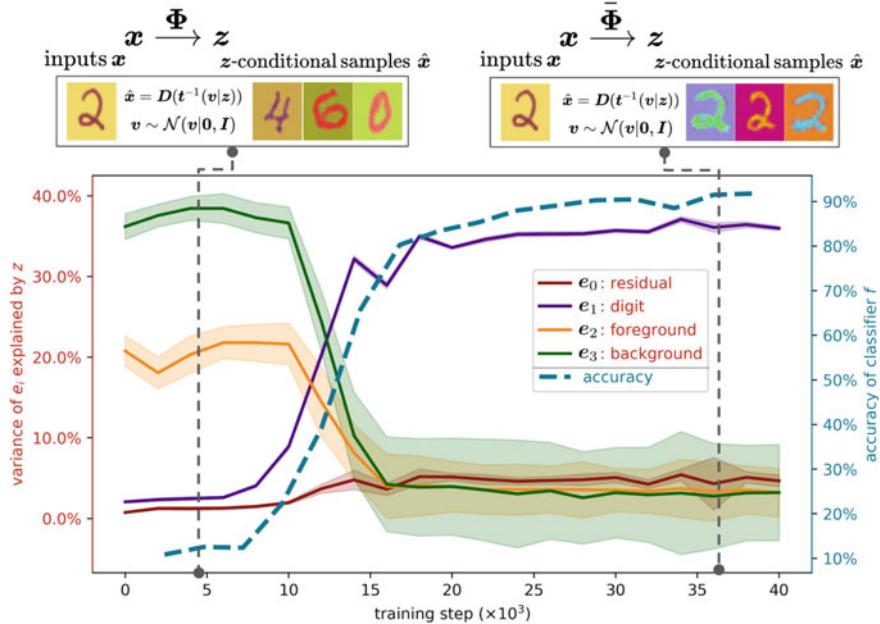
**Fig. 6** Left: Visualizing FaceNet representations and their invariances. Sampling multiple reconstructions  $\hat{x} = \mathbf{D}(\mathbf{t}^{-1}(\mathbf{v}|z_\ell))$  shows the degree of invariance learned by different layers  $\ell$ . The invariance w.r.t. pose increases for deeper layers as expected for face identification. Surprisingly, FaceNet uses glasses as an identity feature throughout all its layers as evident from the spatial mean and variance plots, where the glasses are still visible. This reveals a bias and weakness of the model. Right: spatially averaged variances over multiple  $x$  and layers

Intuitively, a network should, for example, be able to solve a given classification problem by focusing on the relevant information while discarding task-irrelevant information. To build on this intuition, we construct a toy problem: digit classification on ColorMNIST. We expect the model to ignore both the random background and foreground colors of the input data, as it does not help making a classification decision. Thus, we apply the invertible approach presented in Sect. 3.2 and recover three distinct factors: digit class, background color, and foreground color. To capture the semantic changes occurring over the course of training of this classifier, we couple 20 instances of the invertible interpretation model on the last convolutional layer, each representing a checkpoint between iteration 0 and iteration 40000 (equally distributed). The result is shown in Fig. 7: we see that the digit factor becomes increasingly more relevant, with its relevance being strongly correlated to the accuracy of the model.

### 4.3 Effects of Data Shifts on Models

This section investigates the effects that altering input data has on the model we want to understand. We examine these effects by manipulating input data through adversarial attacks or image stylization.

**How do adversarial attacks affect network representations?** Here, we experiment with the fast gradient sign method (FGSM) [GSS15], which manipulates the input image by maximizing the objective of a given classification model. To understand



**Fig. 7** Analyzing the degree to which different semantic concepts are captured by a network representation changes as training progresses. For SqueezeNet on ColorMNIST, we measure how much the data varies in different semantic concepts  $e_i$  and how much of this variability is captured by  $z$  at different training iterations. Early on,  $z$  is sensitive to foreground and background color, and later on it learns to focus on the digit attribute. The ability to encode this semantic concept is proportional to the classification accuracy achieved by  $z$ . At training iterations 4k and 36k, we apply our method to visualize model representations and thereby illustrate how their content changes during training

how such an attack modifies representations of a given model, we first compute the image's invariances with respect to the model as  $v = t(E(x)|\Phi(x))$ . For an attacked image  $x^*$ , we then compute the attacked representation as  $z^* = \Phi(x^*)$ . Decoding this representation with the original invariance  $v$  allows us to precisely visualize what the adversarial attack changed. This decoding,  $\hat{x}^* = D(t(v|z^*))$ , is shown in Fig. 8. We observe that, over layers of the network, the adversarial attack gradually changes the representation toward its target. Its ability to do so is strongly correlated with the amount of invariances, quantified as the total variance explained by  $v$ , for a given layer as also observed in [JBZB19].

**How does training on different data affect the model?** Geirhos et al. [GRM+19] proposed the hypothesis that classification networks based on convolutional blocks mainly focus on texture patterns to obtain class probabilities. We further validate this hypothesis by training our invertible network  $t$  conditioned on pre-logits  $z = \Phi(x)$  (i.e., the penultimate layer) of two ResNet-50 realizations. As shown in Fig. 9, a ResNet architecture trained on standard ImageNet is susceptible to the

Perturbation	$x$	Visualizing perturbed representation at					prediction
		input	conv	fc	logits		
None							Siamese cat
Random							Siamese cat
Attack							Mountain lion
Variance of $\hat{z}$ explained by $v$		11.82% ( $\pm 0.52$ )	7.22% ( $\pm 0.16$ )	49.59% ( $\pm 2.00$ )	84.77% ( $\pm 5.77$ )		

**Fig. 8** Visualizing FGSM adversarial attacks on ResNet-101. To the human eye, the original image and its attacked version are almost indistinguishable. However, the input image is correctly classified as “Siamese cat”, while the attacked version is classified as “mountain lion”. Our approach visualizes how the attack spreads throughout the network. Reconstructions of representations of attacked images demonstrate that the attack targets the semantic content of deep layers. The variance of  $\hat{z}$  explained by  $v$  combined with these visualizations shows how increasing invariances cause vulnerability to adversarial attacks

so-called “texture-bias”, as samples generated conditioned on representation of pure texture images consistently show valid images of corresponding input classes. We furthermore visualize that this behavior can indeed be removed by training the same architecture on a stylized version of ImageNet<sup>1</sup>; the classifier does focus on shape. Rows 10–12 of Fig. 9 show that the proposed approach can be used to generate sketch-based content with the texture-agnostic network.

#### 4.4 Modifying Representations

Invertible access to semantic concepts enables targeted modifications of representations  $\hat{z}$ . In combination with a decoder for  $\hat{z}$ , we obtain semantic image editing capabilities. We provide an example in Fig. 10, where we modify the factors hair color, glasses, gender, beard, age, and smile. We infer  $\hat{z} = \mathbf{E}(x)$  from an input image. Our semantic INN  $e$  then translates this representation into semantic factors  $(\mathbf{e}_i)_i = \mathbf{e}(\hat{z})$ , where individual semantic concepts can be modified independently via the corresponding factor  $\mathbf{e}_i$ . In particular, we can replace each factor with that from another image, effectively transferring semantics from one representation onto another. Due to the invertibility of  $e$ , the modified representation can be trans-

<sup>1</sup> We used weights available at <https://github.com/rgeirhos/texture-vs-shape>.

Samples $\hat{x} = D(t^{-1}(v z))$ conditioned on ResNet pre-logs $z = \Phi(x)$											
Inputs	$\Phi_{\text{vanilla}}$ : ResNet-50 trained on standard ImageNet						$\Phi_{\text{stylized}}$ : ResNet-50 trained on stylized ImageNet				

**Fig. 9** Revealing texture bias in ImageNet classifiers. We compare visualizations of  $z$  from the penultimate layer of ResNet-50 trained on standard ImageNet (left) and a stylized version of ImageNet (right). On natural images (rows 1–3) both models recognize the input, removing textures through stylization (rows 4–6) makes images unrecognizable to the standard model, however it recognizes objects from textured patches (rows 7–9). Rows 10–12 show that a model without texture bias can be used for sketch-to-image synthesis

lated back into the space of the autoencoder and is readily decoded to a modified image  $\hat{x}$ .

To observe which semantic concepts FaceNet is sensitive to, we compute the average distance  $\|f(\mathbf{x}) - f(\mathbf{x}^*)\|$  between its embeddings of  $\mathbf{x}$  and semantically modified  $\mathbf{x}^*$  over the test set (last row in Fig. 10). Evidently, FaceNet is particularly sensitive to differences in gender and glasses. The latter suggests a failure of FaceNet to identify persons correctly after they put on glasses.

Input $x$	hair $e_1$	glasses $e_2$	gender $e_3$	beard $e_4$	age $e_5$	smiling $e_6$
Mean embedding	0.872	1.000	1.061	0.803	0.874	0.833
Distance ( $\pm$ std)	( $\pm$ 0.048)	( $\pm$ 0.046)	( $\pm$ 0.030)	( $\pm$ 0.041)	( $\pm$ 0.053)	( $\pm$ 0.034)

**Fig. 10** Semantic modifications on CelebA. For each column, after inferring the semantic factors  $(\mathbf{e}_i)_i = \mathbf{e}(\mathbf{E}(\mathbf{x}))$  of the input  $\mathbf{x}$ , we replace one factor  $\mathbf{e}_i$  by that from another randomly chosen image that differs in this concept. The inverse of  $\mathbf{e}$  translates this semantic change back into a modified  $\hat{\mathbf{z}}$ , which is decoded to a semantically modified image. Distances between FaceNet embeddings before and after modification demonstrate its sensitivity to differences in gender and glasses (see also Fig. 6)

## 4.5 Evaluation Details

Here, we provide additional details on the investigated neural networks in Sect. 4 and present a way to quantify the amount of invariances in those networks. This section is only for completeness. Similar to Sect. 3.3, this section can be skipped for readers who are interested in the higher level concepts of our approach rather than its technical details. An overview of INN hyperparameters for all experiments is provided in Table 5.

**Table 5** Hyperparameters of INNs for each experiment. Parameter  $n_{flow}$  denotes the number of invertible blocks within in the model, see Fig. 2. Parameters  $h_w$  and  $h_d$  refer to the width and depth of the fully connected subnetworks  $s_i$  and  $\tau_i$ , respectively

Experiment	INN	Input dim.	$n_{flow}$	$h_w$	$h_d$
Comparison Sect. 4.1	$t$	128	20	1024	2
Understanding models: FaceNet, Sect. 4.2	$t$	128	20	512	2
Understanding models: FaceNet, Sect. 4.2	$e$	128	12	512	2
Data effects: Adversarial attack, Sect. 4.3	$t$	128	20	1024	2
Data effects: Texture bias, Sect. 4.3	$t$	268	20	1024	2
Modifications: FaceNet & CelebA Sect. 4.4	$e$	128	12	512	2

**Table 6** High-level architectures of FaceNet and ResNet, depicted as pytorch-modules. Layers investigated in our experiments are marked in bold. Spatial sizes are provided as a visual aid and vary from model to model in our experiments. If not stated otherwise, we always extract from the *last* layer in a series of blocks (e.g., on the right:  $23 \times \text{BottleNeck down} \rightarrow \mathbb{R}^{8 \times 8 \times 1024}$  refers to the last module in the series of 23 blocks)

FaceNet: Implementations of layers Mixed, Block35, Block17, Block8 can be found at <https://github.com/timesler/facenet-pytorch>. In Section 4.2, the representation from the **2nd** convolutional layers are extracted. Furthermore, BN and AdaAvgPool refer to batch normalization and adaptive average pooling, respectively. The term “down” denotes downsampling by factor 2.

RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
$3 \times \text{Conv, BN, ReLU} \rightarrow \mathbb{R}^{61 \times 61 \times 64}$
MaxPool $\rightarrow \mathbb{R}^{30 \times 30 \times 64}$
$3 \times \text{Conv, BN, ReLU} \rightarrow \mathbb{R}^{13 \times 13 \times 256}$
$5 \times \text{Block35} \rightarrow \mathbb{R}^{13 \times 13 \times 256}$
<b>Mixed down</b> $\rightarrow \mathbb{R}^{6 \times 6 \times 896}$
$10 \times \text{Block17} \rightarrow \mathbb{R}^{6 \times 6 \times 896}$
Mixed down $\rightarrow \mathbb{R}^{2 \times 2 \times 1792}$
$5 \times \text{Block8} \rightarrow \mathbb{R}^{2 \times 2 \times 1792}$
<b>AdaAvgPool</b> $\rightarrow \mathbb{R}^{1 \times 1 \times 1792}$
Dropout, Linear, BN $\rightarrow \mathbb{R}^{512}$
<b>identity embedding</b> $\rightarrow \mathbb{R}^{512}$

ResNet-101: See <https://pytorch.org/docs/stable/torchvision/models.html> for details on other variants of ResNet. Bottleneck, FC, and Norm refer to bottleneck residual unit, fully connected layer, and batch normalization, respectively. The term “down” denotes downsampling by factor 2.

RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
Conv down $\rightarrow \mathbb{R}^{64 \times 64 \times 64}$
Norm, ReLU, <b>MaxPool</b> $\rightarrow \mathbb{R}^{32 \times 32 \times 64}$
$3 \times \text{BottleNeck} \rightarrow \mathbb{R}^{32 \times 32 \times 256}$
$4 \times \text{BottleNeck down} \rightarrow \mathbb{R}^{16 \times 16 \times 512}$
$23 \times \text{BottleNeck down} \rightarrow \mathbb{R}^{8 \times 8 \times 1024}$
$3 \times \text{BottleNeck down} \rightarrow \mathbb{R}^{4 \times 4 \times 2048}$
<b>AvgPool, FC</b>
<b>output</b> $\rightarrow \mathbb{R}^{1000}$

Throughout our experiments, we interpret four different models: SqueezeNet, AlexNet, ResNet, and FaceNet. Summaries of each of model’s architecture are provided in Table 6 and Table 7. Implementations and pretrained weights of these models are taken from:

- SqueezeNet (1.1) [https://pytorch.org/docs/stable/\\_modules/torchvision/models/squeezeNet](https://pytorch.org/docs/stable/_modules/torchvision/models/squeezeNet);
- ResNet: [https://pytorch.org/docs/stable/\\_modules/torchvision/models/resnet.html](https://pytorch.org/docs/stable/_modules/torchvision/models/resnet.html);
- AlexNet: [https://pytorch.org/docs/stable/\\_modules/torchvision/models/alexnet.html](https://pytorch.org/docs/stable/_modules/torchvision/models/alexnet.html);
- FaceNet: <https://github.com/timesler/facenet-pytorch>.

**Explained variance:** To quantify the amount of invariances and semantic concepts, we use the fraction of the total variance explained by invariances (Fig. 8) and the fraction of the variance of a semantic concept explained by the model representation (Fig. 7).

**Table 7** High-level architectures of SqueezeNet and AlexNet, depicted as pytorch-modules. See Table 6 for further details

SqueezeNet: We extract the penultimate Fire block for interpretation in Section 4.2. AdaAvgPool and Fire refer to adaptive average pooling and the Fire module introduced in [IHM<sup>+</sup>16].

AlexNet: The first convolution uses kernel size 11. AdaAvgPool, Flatten, and Linear refer to adaptive average pooling, reshaping the three-dimensional representation of the input to a vector, and fully-connected layer, respectively.

RGB image $\mathbf{x} \in \mathbb{R}^{128 \times 128 \times 3}$
Conv, ReLU, MaxPool $\rightarrow \mathbb{R}^{31 \times 31 \times 64}$
$2 \times$ Fire $\rightarrow \mathbb{R}^{31 \times 31 \times 128}$
MaxPool $\rightarrow \mathbb{R}^{15 \times 15 \times 128}$
$2 \times$ Fire $\rightarrow \mathbb{R}^{15 \times 15 \times 256}$
MaxPool $\rightarrow \mathbb{R}^{7 \times 7 \times 256}$
$4 \times$ Fire $\rightarrow \mathbb{R}^{7 \times 7 \times 512}$
Dropout, Conv, ReLU $\rightarrow \mathbb{R}^{7 \times 7 \times 1000}$
AdaAvgPool $\rightarrow \mathbb{R}^{7 \times 7 \times 1000}$
output $\rightarrow \mathbb{R}^{1000}$

RGB image $\mathbf{x} \in \mathbb{R}^{128 \times 128 \times 3}$
Conv, ReLU, MaxPool $\rightarrow \mathbb{R}^{15 \times 15 \times 64}$
Conv, ReLU, MaxPool $\rightarrow \mathbb{R}^{7 \times 7 \times 192}$
Conv, ReLU $\rightarrow \mathbb{R}^{7 \times 7 \times 384}$
$2 \times$ Conv, ReLU $\rightarrow \mathbb{R}^{7 \times 7 \times 256}$
MaxPool $\rightarrow \mathbb{R}^{3 \times 3 \times 256}$
AdaAvgPool, Flatten $\rightarrow \mathbb{R}^{9216}$
Dropout, Linear, ReLU $\rightarrow \mathbb{R}^{4096}$
Dropout, Linear, ReLU $\rightarrow \mathbb{R}^{4096}$
Linear $\rightarrow \mathbb{R}^{1000}$

Using the INN  $t$ , we can consider  $\hat{\mathbf{z}} = t^{-1}(\mathbf{v}|\mathbf{z})$  as a function of  $\mathbf{v}$  and  $\mathbf{z}$ . The total variance of  $\hat{\mathbf{z}}$  is then obtained by sampling  $\mathbf{v}$ , via its prior which is a standard normal distribution, and  $\mathbf{z}$ , via  $\mathbf{z} = \Phi(\mathbf{x})$  with  $\mathbf{x} \sim p_{\text{valid}}(\mathbf{x})$  sampled from a validation set. We compare this total variance to the average variance obtained when sampling  $\mathbf{v}$  for a given  $\mathbf{z}$  to obtain the fraction of the total variance explained by invariances:

$$\mathbb{E}_{\mathbf{x}' \sim p_{\text{valid}}(\mathbf{x}')} \left[ \frac{\text{Var}_{\mathbf{v} \sim \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I})} t^{-1}(\mathbf{v}|\Phi(\mathbf{x}'))}{\text{Var}_{\substack{\mathbf{x} \sim p_{\text{valid}}(\mathbf{x}) \\ \mathbf{v} \sim \mathcal{N}(\mathbf{v}|\mathbf{0}, \mathbf{I})}} t^{-1}(\mathbf{v}|\Phi(\mathbf{x}))} \right]. \quad (20)$$

In combination with the INN  $e$ , which transform  $\hat{\mathbf{z}}$  to semantically meaningful factors, we can analyze the semantic content of a model representation  $\mathbf{z}$ . To analyze how much of a semantic concept represented by factor  $e_i$  is captured by  $\mathbf{z}$ , we use  $e$  to transform  $\hat{\mathbf{z}}$  into  $e_i$  and measure its variance. To measure how much the semantic concept is explained by  $\mathbf{z}$ , we simply swap the roles of  $\mathbf{z}$  and  $\mathbf{v}$  in (20), to obtain

$$\mathbb{E}_{\mathbf{v}' \sim \mathcal{N}(\mathbf{v}' | \mathbf{0}, \mathbf{I})} \left[ \frac{\text{Var}_{\mathbf{x} \sim p_{\text{valid}}(\mathbf{x})} e(\mathbf{t}^{-1}(\mathbf{v}' | \Phi(\mathbf{x})))_i}{\text{Var}_{\substack{\mathbf{v} \sim \mathcal{N}(\mathbf{v} | \mathbf{0}, \mathbf{I}) \\ \mathbf{x} \sim p_{\text{valid}}(\mathbf{x})}} e(\mathbf{t}^{-1}(\mathbf{v} | \Phi(\mathbf{x})))_i} \right]. \quad (21)$$

Figure 8 reports (20) and its standard error when evaluated via 10k samples, and Fig. 7 reports (21) and its standard error when evaluated via 10k samples.

## Conclusions

Understanding a representation in terms of both its semantics and learned invariances is crucial for interpretation of deep networks. We presented an approach (*i*) to recover the invariances a model has learned and (*ii*) to translate the representation and its invariances onto an equally expressive yet semantically accessible encoding. Our diagnostic method is applicable in a plug-and-play fashion on top of existing deep models with no need to alter or retrain them. Since our translation onto semantic factors is bijective, it loses no information and also allows for semantic modifications. Moreover, recovering invariances probabilistically guarantees that we can correctly visualize representations and sample them without leaving the underlying distribution, which is a common cause for artifacts. Altogether, our approach constitutes a powerful, widely applicable diagnostic pipeline for explaining deep representations.

## References

- [AKW+19] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E.W. Pellegrini, R.S. Klessen, L. Maier-Hein, C. Rother, U. Köthe, Analyzing inverse problems with invertible neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)* (New Orleans, LA, USA, 2019), pp. 1–20
- [AS18] Alessandro Achille, Stefano Soatto, Emergence of invariance and disentanglement in deep representations. *J. Mach. Learn. Res. (JMLR)* **19**(50), 1–34 (2018)
- [BBM+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, Wojciech Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE* **10**(7), 1–46 (2015)
- [BDS19] A. Brock, J. Donahue, K. Simonyan, Large scale GAN training for high fidelity natural image synthesis, in *Proceedings of the International Conference on Learning Representations (ICLR)* (New Orleans, LA, USA, 2019), pp. 1–35
- [BMDO21a] A. Blattmann, T. Milbich, M. Dorkenwald, B. Ommer, Behavior-driven synthesis of human dynamics, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference (2021), pp. 12236–12246
- [BMDO21b] A. Blattmann, T. Milbich, M. Dorkenwald, B. Ommer, iPOKE: poking a still image for controlled stochastic video synthesis, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, virtual conference (2021), pp. 14707–14717

- [BZK+17] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: quantifying interpretability of deep visual representations, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI, USA, 2017), pp. 6541–6549
- [BZS+19] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J.B. Tenenbaum, W.T. Freeman, A. Torralba, GAN dissection: visualizing and understanding generative adversarial networks, in *Proceedings of the International Conference on Learning Representations (ICLR)* (New Orleans, LA, USA, 2019), pp. 1–18
- [CSX+18] Q. Cao, L. Shen, W. Xie, O.M. Parkhi, A. Zisserman, VGGFace2: a dataset for recognising faces across pose and age, in *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition (FG)* (Xi'an, China, 2018), pp. 67–74
- [CWG+18] S.A. Cadena, M.A. Weis, L.A. Gatys, M. Bethge, A.S. Ecker, Diverse feature visualizations reveal invariances in early layers of deep neural networks, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich, Germany, 2018), pp. 225–240
- [DB16] A. Dosovitskiy, T. Brox, Generating images with perceptual similarity metrics based on deep networks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Barcelona, Spain, 2016), pp. 658–666
- [DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, ImageNet: a large-scale hierarchical image database, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Miami, FL, USA, 2009), pp. 248–255
- [DMB+21] M. Dorkenwald, T. Milbich, A. Blattmann, R. Rombach, K.G. Derpanis, B. Ommer, Stochastic image-to-video synthesis using cINNs, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference (2021), pp. 3742–3753
- [DSB17] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real NVP, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Toulon, France, 2017), pp. 1–32
- [DW19] B. Dai, D. Wipf, Diagnosing and enhancing VAE models, in *Proceedings of the International Conference on Learning Representations (ICLR)* (New Orleans, LA, USA, 2019), pp. 1–12
- [EHO19] P. Esser, J. Haux, B. Ommer, Unsupervised robust disentangling of latent characteristics for image synthesis, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Seoul, Korea, 2019), pp. 2699–2709
- [ERO20] P. Esser, R. Rombach, B. Ommer, A disentangling invertible interpretation network for explaining latent representations, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference (2020), pp. 9223–9232
- [Eur20] European Commission, White Paper on Artificial Intelligence – A European Approach to Excellence and Trust. Technical report, European Union (2020). <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2020:65:FIN>
- [FV18] R. Fong, A. Vedaldi, Net2Vec: quantifying and explaining how concepts are encoded by filters in deep neural networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Salt Lake City, UT, USA, 2018), pp. 8730–8738
- [GAOI19] L. Goetschalckx, A. Andonian, A. Oliva, P. Isola, GANalyze: toward visual definitions of cognitive image properties, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Seoul, Korea, 2019), pp. 5744–5753
- [GF17] Bryce Goodman, Seth Flaxman, European union regulations on algorithmic decision-making and a “right to explanation”. *AI Mag.* **38**(3), 50–57 (2017)
- [GRM+19] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F.A. Wichmann, W. Brendel, ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, in *Proceedings of the International Conference on Learning Representations (ICLR)* (New Orleans, LA, USA, 2019), pp. 1–22

- [GSS15] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in *Proceedings of the International Conference on Learning Representations (ICLR)* (San Diego, CA, USA, 2015), pp. 1–11
- [HRU+17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local nash equilibrium, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Long Beach, CA, USA, 2017), pp. 6626–6637
- [HZRS16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, 2016), pp. 770–778
- [IHM+16] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-Level Accuracy With 50x Fewer Parameters and < 0.5 MB Model Size (2016) pp. 1–13. [arXiv:1602.07360](https://arxiv.org/abs/1602.07360)
- [JBZB19] J-H. Jacobsen, J. Behrmann, R. Zemel, M. Bethge, Excessive invariance causes adversarial vulnerability, in *Proceedings of the International Conference on Learning Representations (ICLR)* (New Orleans, LA, USA, 2019), pp. 1–17
- [KD18] D.P. Kingma, P. Dhariwal, Glow: generative flow with invertible 1x1 convolutions, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Montréal, QC, Canada, 2018), pp. 10236–10245
- [KSH12] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Lake Tahoe, NV, USA, 2012), pp. 1106–1114
- [KSLO19] D. Kotovenko, A. Sanakoyeu, S. Lang, B. Ommer, Content and style disentanglement for artistic style transfer, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Seoul, Korea, 2019), pp. 4422–4431
- [KW14] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Banff, AB, Canada, 2014), pp. 1–14
- [WKWT15] T.D. Kulkarni, W. Whitney, P. Kohli, J.B. Tenenbaum, Deep convolutional inverse graphics network, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Montréal, QC, Canada, 2015), pp. 2539–2547
- [LBL+19] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, O. Bachem, Challenging common assumptions in the unsupervised learning of disentangled representations, in *Proceedings of the International Conference on Machine Learning (ICML)* (Long Beach, CA, USA, 2019), pp. 4114–4124
- [LBMO19] D. Lorenz, L. Bereska, T. Milbich, B. Ommer, Unsupervised part-based disentangling of object shape and appearance, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA, USA, 2019), pp. 10955–10964
- [LCB98] Y. LeCun, C. Cortes, C.J.C. Burges, The MNIST Database of Handwritten Digits (1998) Retrieved from 2021-11-18
- [LeC12] Y. LeCun, Learning invariant feature hierarchies, in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops* (Firenze, Italy, 2012), pp. 496–505
- [LHM+19] M-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, J. Kautz, Few-shot unsupervised image-to-image translation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Seoul, Korea, 2019), pp. 10551–10560
- [Lip18] Z.C. Lipton, The mythos of model interpretability: in machine learning, the concept of interpretability is both important and slippery. *ACM Queue* **16**(3), 31–57 (2018)
- [LLWT15] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Santiago, Chile, 2015), pp. 3730–3738
- [LSOL20] Y. Li, K. Kumar Singh, U. Ojha, Y.J. Lee, MixNMatch: multifactor disentanglement and encoding for conditional image generation, in *Proceedings of the IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference (2020), pp. 8039–8048
- [Mil19] Tim Miller, Explanation in artificial intelligence: insights from the social sciences. *Artif. Intell.* **267**, 1–38 (2019)
- [MLB+17] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, Klaus-Robert Müller, Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognit.* **65**(5), 211–222 (2017)
- [MOT15] A. Mordvintsev, C. Olah, M. Tyka, Inceptionism: going deeper into neural networks (2015). Accessed 2021-11-18
- [MSM18] Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Methods for interpreting and understanding deep neural networks. *Digital Signal Process.* **73**, 1–15 (2018)
- [MV16] Aravindh Mahendran, Andrea Vedaldi, Visualizing deep convolutional neural networks using natural pre-images. *Int. J. Comput. Vis. (IJCV)* **120**, 233–255 (2016)
- [NDY+16] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Barcelona, Spain, 2016), pp. 3387–3395
- [NWKW19] C. Nash, N. Kushner, C.K.I. Williams, Inverting supervised representations with autoregressive neural density models, in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)* (Naha, Japan, 2019), pp. 1620–1629
- [PASC+20] G. Plumb, M. Al-Shedivat, Á.A. Cabrera, A. Perer, E. Xing, A. Talwalkar, Regularizing black-box models for improved interpretability, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, virtual conference (2020), pp. 10526–10536
- [Red93] A. Norman Redlich, Supervised factorial learning. *Neural Comput.* **5**(5), 750–766 (1993)
- [REO20] R. Rombach, P. Esser, B. Ommer, Network-to-network translation with conditional invertible neural networks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)*, virtual conference (2020), pp. 2784–2797
- [RMW14] D.J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Banff, AB, Canada, 2014), pp. 1–14
- [SCD+20] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis. (IJCV)* **128**, 336–359 (2020)
- [SGM+20] A. Shocher, Y. Gandalman, I. Mosseri, M. Yarom, M. Irani, W.T. Freeman, T. Dekel, Semantic pyramid for image generation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference (2020), pp. 7457–7466
- [SIT+19] S. Santurkar, A. Ilyas, D. Tsipras, L. Engstrom, B. Tran, A. Madry, Image synthesis with a single (robust) classifier, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Vancouver, BC, Canada, 2019), pp. 1260–1271
- [SKP15] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: a unified embedding for face recognition and clustering, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Boston, MA, USA, 2015), pp. 815–823
- [SR15] M. Simon, E. Rodner, Neural activation constellations: unsupervised part model discovery with convolutional networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Santiago, Chile, 2015), pp. 1143–1151
- [SRD14] M. Simon, E. Rodner, J. Denzler, Part detector discovery in deep convolutional neural networks, in *Proceedings of the Asian Conference on Computer Vision (ACCV)* (Singapore, Singapore, 2014), pp. 162–177

- [SVZ14] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: visualising image classification models and saliency maps, in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops* (Banff, AB, Canada, 2014), pp. 1–8
- [SWM17] W. Samek, T. Wiegand, K-R. Müller, *Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models* (Springer, 2017)
- [SZS+14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Banff, AB, Canada, 2014), pp. 1–10
- [UGP+17] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, K. Weinberger, Deep feature interpolation for image content changes, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI, USA, 2017), pp. 7064–7073
- [XYA19] Z. Xiao, Q. Yan, Y. Amit, Generative Latent Flow (2019), pp. 1–18. [arXiv:1905.10485](https://arxiv.org/abs/1905.10485)
- [YCN+15] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, in *Proceedings of the International Conference on Machine Learning (ICML) Workshops* (2015), pp. 1–12
- [ZF14] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Zurich, Switzerland, 2014), pp. 818–833
- [ZIE+18] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Salt Lake City, UT, USA, 2018), pp. 586–595
- [ZKL+15] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Object detectors emerge in deep scene CNNs, in *Proceedings of the International Conference on Learning Representations (ICLR)* (San Diego, CA, USA, 2015), pp. 1–12
- [ZKL+16] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, 2016), pp. 2921–2929
- [ZNWZ18] Q. Zhang, Y. Nian Wu, S-C. Zhu, Interpretable convolutional neural networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Salt Lake City, UT, USA, 2018), pp. 8827–8836

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



# Confidence Calibration for Object Detection and Segmentation



Fabian Küppers, Anselm Haselhoff, Jan Kronenberger, and Jonas Schneider

**Abstract** Calibrated confidence estimates obtained from neural networks are crucial, particularly for safety-critical applications such as autonomous driving or medical image diagnosis. However, although the task of confidence calibration has been investigated on classification problems, thorough investigations on object detection and segmentation problems are still missing. Therefore, we focus on the investigation of confidence calibration for object detection and segmentation models in this chapter. We introduce the concept of multivariate confidence calibration that is an extension of well-known calibration methods to the task of object detection and segmentation. This allows for an extended confidence calibration that is also aware of additional features such as bounding box/pixel position and shape information. Furthermore, we extend the expected calibration error (ECE) to measure miscalibration of object detection and segmentation models. We examine several network architectures on MS COCO as well as on Cityscapes and show that especially object detection as well as instance segmentation models are intrinsically miscalibrated given the introduced definition of calibration. Using our proposed calibration methods, we have been able to improve calibration so that it also has a positive impact on the quality of segmentation masks as well.

---

F. Küppers (✉) · A. Haselhoff · J. Kronenberger  
Hochschule Ruhr West, Duisburger Str. 100, 45479 Mülheim a.d. Ruhr, Germany  
e-mail: [fabian.kueppers@hs-ruhrwest.de](mailto:fabian.kueppers@hs-ruhrwest.de)

A. Haselhoff  
e-mail: [anselm.haselhoff@hs-ruhrwest.de](mailto:anselm.haselhoff@hs-ruhrwest.de)

J. Kronenberger  
e-mail: [jan.kronenberger@hs-ruhrwest.de](mailto:jan.kronenberger@hs-ruhrwest.de)

J. Schneider  
Elektronische Fahrwerksysteme GmbH, Dr.-Ludwig-Kraus-Str. 6,  
85080 Gaimersheim, Germany  
e-mail: [jonas.schneider@efs-auto.com](mailto:jonas.schneider@efs-auto.com)

## 1 Introduction

Common neural networks for object detection must not only determine the position and class of an object but also denote their confidence about the correctness of each detection. This also holds for instance or semantic segmentation models that output a score for each pixel indicating the confidence of object mask membership. A reliable confidence estimate for detected objects is crucial, particularly for safety-critical applications such as autonomous driving, to reliably process detected objects. Another example is medical diagnosis, where, for example, the shape of a brain tumor within an MRI image is of special interest [MWT+20].

Each confidence estimate might be seen as a probability of correctness, reflecting the model’s uncertainty about a detection or the pixel mask. During inference, we expect the estimated confidence to match the observed precision for a prediction. For example, given 100 predictions with 80% confidence each, we expect 80 predictions to be correctly predicted [GPSW17, KKSH20]. However, recent work has shown that the confidence estimates of either classification or detection models based on neural networks are *miscalibrated*, i.e., the confidence does not match the observed accuracy in classification [GPSW17] or the observed precision in object detection [KKSH20]. While confidence calibration within the scope of classification has been extensively investigated [NCH15, NC16, GPSW17, KSFF17, KPNK+19], we recently defined the term of calibration for object detection and proposed methods to measure and resolve miscalibration [KKSH20, SKR+21]. In this context, we measured miscalibration w.r.t. the position and scale of detected objects by also including the regression branch of an object detector into a calibration mapping. We have been able to show that modern object detection models also tend to be miscalibrated. On the one hand, this can be mitigated using standard calibration methods. On the other hand, we show that our proposed methods for position- and shape-dependent calibration [KKSH20] are able to further reduce miscalibration.

In this chapter we review our methods for position- and shape-dependent confidence calibration and provide a definition for confidence calibration for the task of *instance/semantic segmentation*. To this end, we extend the definition of the *expected calibration error* (ECE) to measure miscalibration within the scope of instance/semantic segmentation. Furthermore, we adapt the extended calibration methods originally designed for object detection [KKSH20] to enable position-dependent confidence calibration for segmentation tasks as well.

This chapter is structured as follows. In Sect. 2 we review the most important related works regarding confidence calibration. In Sect. 3 we provide the definitions of calibration for the tasks of object detection, instance segmentation, and semantic segmentation. Furthermore, in Sect. 4 we present the extended confidence calibration methods for object detection and segmentation. Extensive experimental evaluations on a variety of architectures and computer vision problems, including object detection and instance/semantic segmentation, are discussed in Sect. 5. Finally, we provide conclusions and discuss our most important findings.

## 2 Related Works

In the past, most research focused on measuring and resolving miscalibration for classification tasks. In this scope, the *expected calibration error* (ECE) [NCH15] is commonly used in conjunction with the Brier score and negative log likelihood (NLL) loss to measure miscalibration. For calculating the ECE, all samples are grouped into equally sized bins by their confidence. Afterward, for each bin the accuracy is calculated and used as an approximation of the accuracy of a single sample in its respective bin. Recently, several extensions such as classwise ECE [KPNK+19], marginal calibration error [KLM19], or adaptive calibration error [NDZ+19] for the evaluation of multi-class problems have also been proposed, where the ECE is evaluated for each class separately. In contrast to previous work, we extend the common ECE definition to the task of object detection [KKSH20] and instance/semantic segmentation. This extension allows for a position-dependent miscalibration evaluation so that it is possible to quantify miscalibration separately for certain image regions. The definition is given in Sect. 3.

Besides measuring miscalibration, it is also possible to resolve a potential miscalibration by using calibration methods that are applied after inference. These post-hoc calibration methods can be divided into binning and scaling methods. Binning methods such as histogram binning [ZE01], isotonic regression [ZE02], Bayesian binning into quantiles (BBQ) [NCH15], or ensemble of near-isotonic regression (ENIR) [NC16] group all samples into several bins by their confidence (similar to the ECE calculation) and perform a mapping from uncalibrated confidence estimates to calibrated ones. In contrast, scaling methods such as logistic calibration (Platt scaling) [Pla99], temperature scaling [GPSW17], beta calibration [KSFF17], or Dirichlet calibration [KPNK+19] scale the network logits before sigmoid/softmax activation to obtain calibrated confidences. The scaling parameters are commonly obtained by logistic regression. Other approaches comprise binwise temperature scaling [JJY+19] or scaling-binning calibrator [KLM19], combining both approaches to further improve calibration performance for classification tasks.

Recently, we proposed an extension of common calibration methods to object detection by also including the bounding box regression branch of an object detector into a calibration mapping [KKSH20]. On the one hand, we extended the histogram binning [ZE01] to perform calibration using a multidimensional binning scheme. On the other hand, we also extended scaling methods to include position information into a calibration mapping. Both approaches are presented in Sect. 4 in more detail.

Unlike classification, the task of instance or semantic segmentation calibration has not yet been addressed by many authors. In the work of [WLK+20], the authors perform online confidence calibration of the classification head of an instance segmentation model and show that this has a significant impact on the mask quality. The authors in [KG20] use a multi-task learning approach for semantic segmentation models to improve model calibration and out-of-distribution detection within the scope of medical image diagnosis. A related approach is proposed by [MWT+20], where the authors train multiple fully-connected networks as an ensemble to obtain

well-calibrated semantic segmentation models. However, none of these methods provide an explicit definition of calibration for segmentation models (instance or semantic). This problem is addressed by [DLXS20], where the authors explicitly define semantic segmentation calibration and propose a local temperature scaling for semantic image masks. This approach utilizes the well-known temperature scaling [GPSW17] and assigns a temperature for each mask pixel separately. Furthermore, they use a dedicated convolutional neural network (CNN) to infer the temperature parameters for each image. Our definition of segmentation calibration in Sect. 3 is conform with their definition. Our approach differs from their proposed image-based temperature scaling as we use a single position-dependent calibration mapping to model the probability distributions for all images.

### 3 Calibration Definition and Evaluation

In this section, we review the term of confidence calibration for *classification* tasks [NCH15, GPSW17] and extend this definition to *object detection*, *instance segmentation*, and *semantic segmentation*. Furthermore, we derive the *detection expected calibration error* (D-ECE) to measure miscalibration.

#### 3.1 Definitions of Calibration

**Classification:** In a first step, we define the datasets  $\mathcal{D}$  of size  $|\mathcal{D}| = N$  with indices  $i \in \mathcal{I} = \{1, \dots, N\}$ , consisting of images  $\mathbf{x} \in \mathbb{I}^{H \times W \times C}$ ,  $\mathbb{I} = [0, 1]$ , with height  $H$ , width  $W$ , and number of channels  $C$ . Each image has ground-truth information that consists of the class information  $\bar{Y} \in \mathcal{Y} = \{1, \dots, Y\}$ . As an introduction into the task of confidence calibration, we start with the definition of perfect calibration for classification. A classification model  $\mathbf{F}_{\text{cls}}$  outputs a label  $\hat{Y}$  and a corresponding confidence score  $\hat{P}$  indicating its belief about the prediction’s correctness. In this case, perfect calibration is defined by [GPSW17]

$$\mathbb{P}(\hat{Y} = \bar{Y} | \hat{P} = p) = p \quad \forall p \in [0, 1]. \quad (1)$$

In other words, the accuracy  $\mathbb{P}(\hat{Y} = \bar{Y} | \hat{P} = p)$  for a certain confidence level  $p$  should match the estimated confidence. If we observe a deviation, a model is called *miscalibrated*. In binary classification, we rather consider the *relative frequency* of  $\bar{Y} = 1$  instead of the accuracy as the calibration measure. We illustrate this difference using the following example. Consider  $N = 100$  images of a dataset  $\mathcal{D}$  with binary ground-truth labels  $\bar{Y} \in \{0, 1\}$ , where 50 images are labeled as  $\bar{Y} = 0$  and 50 images with  $\bar{Y} = 1$ . Furthermore, consider a classification model  $\mathbf{F}_{\text{cls}}$  with a sigmoidal output in  $[0, 1]$ , indicating its confidence for  $\bar{Y} = 1$ . In our example, this model is able to always predict the correct ground-truth label with confidences  $\hat{P} = 0$  and  $\hat{P} = 1$  for

$\hat{Y} = 0$  and  $\hat{Y} = 1$ , respectively. Thus, the network is able to achieve an accuracy of 100% but with an average confidence of 50%. Therefore, we consider the relative frequency for  $\bar{Y} = 1$  in a binary classification task as the calibration goal that is also 50% in this scenario.

**Object detection:** In the next step, we extend our dataset and model to the task of object detection. In contrast to a classification dataset, an object detection dataset consists of ground-truth annotations  $\bar{Y} \in \mathcal{Y} = \{1, \dots, Y\}$  for each object within an image as well as the ground-truth position and shape information  $\bar{\mathbf{R}} \in \mathcal{R} = [0, 1]^A$  ( $A$  denotes the size of the normalized box encoding, comprising the center  $x$  and  $y$  positions  $c_x, c_y$ , as well as width  $w$  and height  $h$ ). An object detection model  $\mathbf{F}_{\text{det}}$  further outputs a confidence score  $\hat{P} \in [0, 1]$ , a label  $\hat{Y} \in \mathcal{Y}$ , and the corresponding position  $\hat{\mathbf{R}} \in \mathcal{R}$  for each detection in an image  $\mathbf{x}$ . Extending the original formulation for confidence calibration within classification tasks [GPSW17], perfect calibration for *object detection* is defined by [KKSH20]

$$\begin{aligned} P(M = 1 | \hat{P} = p, \hat{Y} = y, \hat{\mathbf{R}} = \mathbf{r}) &= p \\ \forall p \in [0, 1], y \in \mathcal{Y}, \mathbf{r} \in \mathcal{R}, \end{aligned} \quad (2)$$

where  $M = 1$  denotes a correctly classified prediction that matches a ground-truth object with a certain intersection-over-union (IoU) score. Commonly, an object detection model is calibrated by means of its precision, as the computation of the accuracy is not possible without knowing all anchors of a model [KKSH20, SKR+21]. Thus,  $P(M = 1)$  is a shorthand notation for  $p(\hat{Y} = \bar{Y}, \hat{\mathbf{R}} = \bar{\mathbf{R}})$  that expresses the precision for a dedicated IoU threshold.

**Instance segmentation:** At this point, we adapt this idea to define confidence calibration for *instance segmentation*. Consider a dataset with  $K$  annotated objects  $\mathcal{K}$  over all images. For notation simplicity, we further use  $j \in \mathcal{J}_k = \{1, \dots, H_k \cdot W_k\}$  as the index for pixel  $j$  within a bounding box  $\bar{\mathbf{R}}_k$  of object  $k \in \mathcal{K}$  in the instance segmentation dataset, where  $H_k$  and  $W_k$  denote the width and height of object  $k$ , respectively. In addition to object detection, a ground-truth dataset for instance segmentation also consists of pixel-wise mask labels denoted by  $\bar{Y}_j \in \mathcal{Y}^* = \{0, 1\}$  for each pixel  $j$  in the bounding box  $\bar{\mathbf{R}}_k$  of object  $k$ . Note that we introduce the star superscript (\*) here to distinguish between the bounding box label/prediction encoding and the instance segmentation label/prediction encoding. An instance segmentation model  $\mathbf{F}_{\text{ins}}$  predicts the membership  $\hat{Y}_j \in \mathcal{Y}^*$  for each pixel  $j$  in the predicted bounding box  $\hat{\mathbf{R}}_k$  to the object mask with a certain confidence  $\hat{P}_j \in [0, 1]$ . We further denote  $\mathbf{R}_j \in \mathcal{R}^* = [0, 1]^{A^*}$  as the position of pixel  $j$  *within the bounding box*  $\hat{\mathbf{R}}_k$ , where  $A^*$  denotes the size of the used position encoding of a pixel within its bounding box. In contrast to object detection, it is possible to treat the confidence scores of each pixel within an instance segmentation mask as a binary classification problem. In this case, the confidence  $\hat{P}_j$  can be interpreted as the probability of a pixel belonging to the object mask. Therefore, the aim of confidence calibration for *instance segmentation* is that the pixel confidence should match the relative frequency that a pixel is part

of the object mask. According to the task of object detection, we further include a position dependency into the definition, for instance, segmentation and obtain

$$\begin{aligned} \text{P}(\hat{Y}_j = \bar{Y}_j | \hat{Y} = y, \hat{P}_j = p, \mathbf{R}_j = \mathbf{r}) &= p, \\ \forall p \in [0, 1], y \in \mathcal{Y}, \mathbf{r} \in \mathcal{R}^*, j \in \mathcal{J}_k, k \in \mathcal{K}. \end{aligned} \quad (3)$$

The former term can be interpreted as the probability that the prediction  $\hat{Y}_j$  for a pixel with index  $j$  within an object  $k$  matches the ground-truth annotation  $\bar{Y}_j$  given a certain confidence  $p$ , a certain pixel position  $\mathbf{r}$  within the bounding box, as well as a certain object category  $y$  that is predicted by the bounding box head of the instance segmentation model.

**Semantic Segmentation:** Compared to object detection or instance segmentation, a *semantic segmentation* dataset does not hold ground-truth information for individual objects but rather consists of pixel-wise class annotations  $\bar{Y}_j \in \mathcal{Y} = \{1, \dots, Y\}$ ,  $j \in \mathcal{J} = \{1, \dots, H \cdot W\}$ . A semantic segmentation model  $\mathbf{F}_{\text{sem}}$  outputs pixel-wise labels  $\hat{Y}_j$  and probabilities  $\hat{P}_j$  with relative position  $\mathbf{R}_j$  *within an image*. Therefore, we can define perfect calibration for *semantic segmentation* as

$$\begin{aligned} \text{P}(\hat{Y}_j = \bar{Y}_j | \hat{P}_j = p, \mathbf{R}_j = \mathbf{r}) &= p, \\ \forall p \in [0, 1], \mathbf{r} \in \mathcal{R}^*, j \in \mathcal{J}, \end{aligned} \quad (4)$$

that is related to the calibration definition of classification [GPSW17]. In addition, the confidence score of each pixel must not only reflect its accuracy given a certain confidence level but also at a certain pixel position.

### 3.2 Measuring Miscalibration

We can measure the miscalibration of an *object detection* model as the expected deviation between confidence and observed precision which is also known as the *detection expected calibration error* (D-ECE) [KKSH20]. Let further  $\mathbf{s} = (p, y, \mathbf{r})$  denote a single detection with confidence  $p$ , class  $y$ , and bounding box  $\mathbf{r}$  so that  $\mathbf{s} \in \mathcal{S}$ , where  $\mathcal{S}$  is the aggregated set of the confidence space  $[0, 1]$ , the set of ground-truth labels  $\mathcal{Y}$ , and the set of possible bounding box positions  $\mathcal{R}$ . Since  $M$  is a continuous random variable, we need to approximate the D-ECE using the Riemann-Stieltjes integral [GPSW17]

$$\mathbb{E}_{\hat{P}, \hat{Y}, \hat{\mathbf{R}}}[\lvert \text{P}(M = 1 | \hat{P} = p, \hat{Y} = y, \hat{\mathbf{R}} = \mathbf{r}) - p \rvert] \quad (5)$$

$$= \int_{\mathcal{S}} \lvert \text{P}(M = 1 | \hat{P} = p, \hat{Y} = y, \hat{\mathbf{R}} = \mathbf{r}) - p \rvert dF_{\hat{P}, \hat{Y}, \hat{\mathbf{R}}}(\mathbf{s}) \quad (6)$$

with  $F_{\hat{P}, \hat{Y}, \hat{\mathbf{R}}}(\mathbf{s})$  as the joint cumulative distribution of  $\hat{P}$ ,  $\hat{Y}$ , and  $\hat{\mathbf{R}}$ . Let  $\mathbf{a} = (a_p, a_y, a_{r_1}, \dots, a_{r_A})^\top \in \mathcal{S}$  and  $\mathbf{b} = (b_p, b_y, b_{r_1}, \dots, b_{r_A})^\top \in \mathcal{S}$  denote interval

boundaries for the confidence  $a_p, b_p \in [0, 1]$ , the estimated labels  $a_y, b_y \in \mathcal{Y}$  and each quantity of the bounding box encoding  $a_{r_1}, b_{r_1}, \dots, a_{r_A}, b_{r_A} \in [0, 1]$  defined on the joint cumulative  $F_{\hat{P}, \hat{Y}, \hat{\mathbf{R}}}(\mathbf{s})$  so that<sup>1</sup>  $F_{\hat{P}, \hat{Y}, \hat{\mathbf{R}}}(\mathbf{b}) - F_{\hat{P}, \hat{Y}, \hat{\mathbf{R}}}(\mathbf{a}) = P(\hat{P} \in [a_p, b_p], \hat{Y} \in [a_y, b_y], \hat{R}_1 \in [a_{r_1}, b_{r_1}], \dots, \hat{R}_A \in [a_{r_A}, b_{r_A}])$ . The integral in (6) is now approximated by

$$\sum_{m \in \mathcal{B}} P(\mathbf{s}_m) \cdot |P(M = 1 | \hat{P} = p_m, \hat{Y} = y_m, \hat{\mathbf{R}} = \mathbf{r}_m) - p_m|, \quad (7)$$

with  $B$  as the number of equidistant bins used for integral approximation so that  $\mathcal{B} = \{1, \dots, B\}$ , and  $p_m \in [0, 1]$ ,  $y_m \in \mathcal{Y}$  and  $\mathbf{r}_m \in \mathcal{R}$  being the respective bin entities for average confidence, the current label, and the average (unpacked) bounding box scores within bin  $m \in \mathcal{B}$ , respectively. Let  $N_m$  denote the amount of samples within a single bin  $m$ . For large datasets  $|\mathcal{D}| = N$ , the probability  $P(M = 1 | \hat{P} = p_m, \hat{Y} = y_m, \hat{\mathbf{R}} = \mathbf{r}_m)$  is approximated by the average precision within a single bin  $m$ , whereas  $p_m$  is approximated by the average confidence, so that the D-ECE can finally be computed by

$$\sum_{m \in \mathcal{B}} \frac{1}{N_m} |\text{prec}(m) - \text{conf}(m)|, \quad (8)$$

where  $\text{prec}(m)$  and  $\text{conf}(m)$  denote the precision and average confidence within bin  $m$ , respectively.

Similarly, we can extend the D-ECE to *instance segmentation* by

$$\mathbb{E}_{\hat{Y}, \hat{P}_j, \mathbf{R}_j} [|P(\hat{Y}_j = \bar{Y}_j | \hat{Y} = y, \hat{P}_j = p, \mathbf{R}_j = \mathbf{r}) - p|] \quad (9)$$

$$\approx \sum_{m \in \mathcal{B}} \frac{1}{N_m} |\text{freq}(m) - \text{conf}(m)|, \quad (10)$$

with a binning scheme over all pixels and  $\text{freq}(m)$  as the average frequency within each bin. In this case, each pixel is treated as a separate prediction and binned by its confidence, label class, and relative position. In the same way, the D-ECE for *semantic segmentation* is approximated by

$$\mathbb{E}_{\hat{P}_j, \mathbf{R}_j} [|P(\hat{Y}_j = \bar{Y}_j | \hat{P}_j = p, \mathbf{R}_j = \mathbf{r}^*) - p|] \quad (11)$$

$$\approx \sum_{m \in \mathcal{B}} \frac{1}{N_m} |\text{acc}(m) - \text{conf}(m)|, \quad (12)$$

with accuracy  $\text{acc}(m)$  within each bin  $m$ .

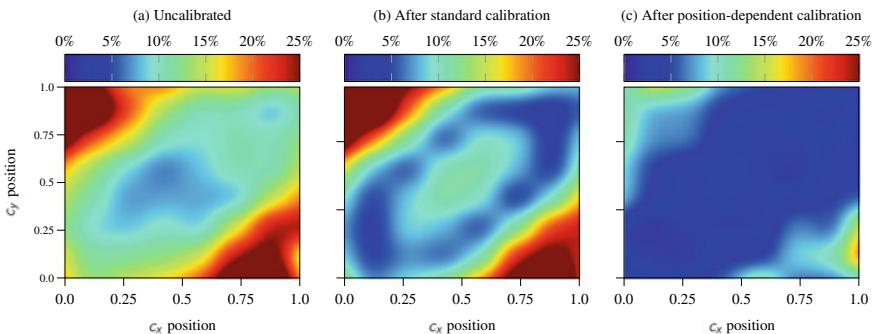
---

<sup>1</sup> For discrete random variables (e.g.,  $Y$ ), we use a continuous density function of the form  $p_Y(y) = \sum_{y^* \in \mathcal{Y}} P(y^*) \delta(y - y^*)$  with  $\delta(x)$  as the Dirac delta function.

For the calibration evaluation of object detection models, we can use the relative position  $c_x, c_y$ , and the shape  $h, w$ , of the bounding boxes [KKSH20]. For segmentation, we consider the relative position  $x, y$ , of each pixel within a bounding box (instance segmentation) or within the image (semantic segmentation), as well as its distance  $d$  to the next segment boundary, as we expect a higher uncertainty in the peripheral areas of a segmentation mask. We use these definitions to evaluate different models in Sect. 5.

## 4 Position-Dependent Confidence Calibration

For post-hoc calibration, we distinguish between binning and scaling methods. According to the approximation in (7), binning methods such as histogram binning [ZE01] divide all samples into several bins by their confidence and measure the average accuracy/precision within each bin. In contrast, scaling methods rescale the logits of a neural network before sigmoid/softmax activation to calibrate a network's output. In this section, we extend standard calibration methods so that they are capable of dealing with additional information such as position and/or shape. These extended methods can be used for confidence calibration of *object detection*, *instance segmentation*, and *semantic segmentation* tasks. We illustrate the difference between standard calibration and position-dependent calibration using an artificially created dataset in Fig. 1. This dataset consists of points with a confidence score and a binary ground-truth information in  $\{0, 1\}$ , so that we are able to compute the frequency of the points across the whole image. We observe that standard calibration only shifts the average confidence to fit the average precision/accuracy. This leads to an



**Fig. 1** **a** Consider an artificial dataset with low calibration error in the center and increasing miscalibration toward the image boundaries. **b** A common logistic calibration model only shifts the average confidence to match the accuracy. This leads to a better global ECE score but also degrades the miscalibration in the center of the image. **c** In contrast, position-dependent calibration is capable of possible dependencies and leads to an overall improvement of calibration (example taken from previous work [KKSH20])

increased calibration error in the center of the image. In contrast, position-dependent calibration as defined in this section is able to capture correlations between position information and calibration error and reduces the D-ECE across the whole image.

## 4.1 Histogram Binning

Given a binning scheme with  $B$  different bins so that  $\mathcal{B} = \{1, \dots, B\}$  with the according bin boundaries  $0 = a_1 \leq a_2 \leq \dots \leq a_{B+1} = 1$  and the corresponding calibrated estimate  $\theta_m$  within each bin, the objective for histogram binning to estimate  $\Theta = \{\theta_m | m \in \mathcal{B}\}$  is the minimization

$$\min_{\Theta} \sum_{i \in \mathcal{I}} \sum_{m \in \mathcal{B}} \mathbb{1}(a_m \leq \hat{p}_i < a_{m+1}) \cdot (\theta_m - \bar{y}_i)^2, \quad (13)$$

with  $\mathbb{1}(\cdot)$  as the indicator function, yielding a 1 if its argument is true and a 0 if it is false [ZE01, GPSW17]. This objective converges to the fraction of positive samples within each bin under consideration. However, within the scope of object detection or segmentation, we also want to measure the accuracy/precision w.r.t. position and shape information. As an extension to the standard histogram binning [ZE01], we therefore propose a multidimensional binning scheme that divides all samples into several bins by their confidence **and** by all additional information such as position and shape [KKSH20]. We further denote  $\hat{\mathbf{s}} = (\hat{P}, \hat{\mathbf{R}})$  of size  $Q = A + 1$  as the input vector to a calibration function consisting of the confidence and the bounding box encoding, so that  $Q = \{1, \dots, Q\}$ . In a multidimensional histogram binning, we indicate the number of bins as  $\mathbf{B} = (B_1, \dots, B_Q)$  so that  $\mathcal{B}^* = \{\mathcal{B}_q = \{1, \dots, B_q\} | q \in Q\}$  with bin boundaries  $\{0 = a_{1,q} \leq a_{2,q} \leq \dots \leq a_{B_q+1,q} = 1 | q \in Q\}$ . For each bin combination  $m_1 \in \mathcal{B}_1, \dots, m_Q \in \mathcal{B}_Q$ , we have a dedicated calibration parameter  $\theta_{m_1, \dots, m_Q} \in \mathbb{R}$  so that the calibration parameters  $\Theta^*$ , with  $|\Theta^*| = \prod_{q \in Q} B_q$ , are given as  $\Theta^* = \{\theta_{m_1, \dots, m_Q} | m_1 \in \mathcal{B}_1, \dots, m_Q \in \mathcal{B}_Q\}$ . This results in an objective function given by

$$\min_{\Theta^*} \sum_{i \in \mathcal{I}} J(\hat{\mathbf{s}}_i), \quad (14)$$

where

$$J(\hat{\mathbf{s}}_i) = \begin{cases} (\theta_{m_1, \dots, m_Q} - \bar{y}_i)^2, & \text{if } (a_{m,q} \leq \hat{s}_{i,q} < a_{m+1,q}) \quad \forall q \in Q, \forall m \in \mathcal{B}_q \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

which again converges to the fraction of positive samples within each bin. The term  $J(\hat{\mathbf{s}}_i)$  simply denotes that the loss is only applied if a sample  $\hat{\mathbf{s}}_i$  falls in a certain bin combination. A drawback of using this method is that the number of bins is given by  $\prod_{q \in Q} B_q$ , which grows exponentially as the number of dimensions  $Q$  grows.

## 4.2 Scaling Methods

As opposed to binning methods like histogram binning, scaling methods perform a rescaling of the logits before sigmoid/softmax activation to obtain calibrated confidences. We can distinguish between logistic calibration (Platt scaling) [Pla99] and beta calibration [KSFF17].

**Logistic calibration:** According to the well-known logistic calibration (Platt scaling) [Pla99], the calibration parameters are commonly obtained using logistic regression. For a binary logistic regression model, we assume normally distributed scores for the positive and negative class, so that  $p(p|+) \sim \mathcal{N}(p; \mu_+, \sigma^2)$  and  $p(p|-) \sim \mathcal{N}(p; \mu_-, \sigma^2)$ . The mean values for the two classes are given by  $\mu_+$ ,  $\mu_-$  and the variance  $\sigma^2$  is equal for all classes. First, we follow the derivation of logistic calibration introduced by [Pla99, KSFF17] using the likelihood ratio

$$LR(p) = \frac{p(p|+)}{p(p|-)} = \exp\left(\frac{1}{2\sigma^2}[-(p - \mu_+)^2 + (p - \mu_-)^2]\right) \quad (16)$$

$$= \exp\left(\frac{1}{2\sigma^2}[2p(\mu_+ - \mu_-) - (\mu_+^2 - \mu_-^2)]\right) \quad (17)$$

$$= \exp\left(\frac{\mu_+ - \mu_-}{2\sigma^2}[p - (\mu_+ + \mu_-)]\right) \quad (18)$$

$$= \exp(\gamma(p - \eta)), \quad (19)$$

where  $\gamma = \frac{1}{2\sigma^2}(\mu_+ - \mu_-)$  and  $\eta = \mu_+ + \mu_-$ . Assuming a uniform prior over the positive and negative classes, the likelihood ratio equals the posterior odds. Hence, a calibrated probability is derived by

$$P(+|\hat{p}) = \frac{1}{1 + LR(\hat{p})^{-1}} = \frac{1}{1 + \exp(-\gamma(\hat{p} - \eta))}, \quad (20)$$

which recovers the logistic function.

Recently, [KKSH20] used this formulation to derive a position-dependent confidence calibration by using multivariate Gaussians for the positive and negative classes. Introducing the concept of multivariate confidence calibration [KKSH20], we can derive a likelihood ratio for position-dependent logistic calibration by

$$LR_{LC}(\hat{s}) = \exp\left(\frac{1}{2}\left[(\hat{s}_-^\top \Sigma_-^{-1} \hat{s}_-) - (\hat{s}_+^\top \Sigma_+^{-1} \hat{s}_+)\right] + c\right), \quad c = \log \frac{|\Sigma_-|}{|\Sigma_+|}, \quad (21)$$

where  $\hat{s}_+ = \hat{s} - \mu_+$  and  $\hat{s}_- = \hat{s} - \mu_-$  using  $\mu_+, \mu_- \in \mathbb{R}^Q$  as the mean vectors and  $\Sigma_+, \Sigma_- \in \mathbb{R}^{Q \times Q}$  as the covariance matrices for the positive and negative classes, respectively.

**Beta calibration:** Similarly, we can also extend the beta calibration method [KSFF17] to a multivariate calibration scheme. However, we need a special form of a multivariate beta distribution as the Dirichlet distribution is only defined over a  $Q$ -simplex and thus is not suitable for this kind of calibration. Therefore, we use a multivariate beta distribution proposed by [LN82] which is defined as

$$p(\hat{s}|\alpha) = \frac{1}{B(\alpha)} \prod_{q \in Q} \left[ \lambda_q^{\alpha_q} (\hat{s}_q^*)^{\alpha_q-1} \left( \frac{\hat{s}_q^*}{\hat{s}_q} \right)^2 \right] \left[ 1 + \sum_{q \in Q} \lambda_q \hat{s}_q^* \right]^{-\sum_{q \in Q_0} \alpha_q}, \quad (22)$$

with  $Q_0 = \{0, \dots, Q\}$  and the shape parameters  $\alpha = (\alpha_0, \dots, \alpha_q)^T$ ,  $\beta = (\beta_0, \dots, \beta_q)^T$  that are restricted to  $\alpha_q, \beta_q > 0$ . Furthermore, we denote  $\lambda_q = \frac{\beta_q}{\beta_0}$  and  $\hat{s}_q^* = \frac{\hat{s}_q}{1-\hat{s}_q}$ . In this context,  $B(\alpha)$  denotes the multivariate beta function. However, this kind of beta distribution is only able to capture positive correlations [LN82]. Nevertheless, it is possible to derive a likelihood ratio given by

$$\begin{aligned} LR_{BC}(\hat{s}) = \exp \left( \sum_{q \in Q} \left[ \alpha_q^+ \log(\lambda_q^+) - \alpha_q^- \log(\lambda_q^-) + (\alpha_q^+ - \alpha_q^-) \log(\hat{s}_q^*) \right] + \right. \\ \left. \sum_{q \in Q_0} \left[ \alpha_q^- \log \left( \sum_{j \in Q} \lambda_j^- \hat{s}_j^* \right) - \alpha_q^+ \log \left( \sum_{j \in Q} \lambda_j^+ \hat{s}_j^* \right) \right] + c \right), \end{aligned} \quad (23)$$

with  $\alpha^+$ ,  $\alpha^-$  and  $\lambda^+$ ,  $\lambda^-$  as the shape parameters for the multivariate beta distribution in (22) and for the positive and negative class, respectively, so that  $c = \log \left( \frac{B(\alpha^-)}{B(\alpha^+)} \right)$ . We investigate the effect of position-dependent calibration in the next section using these methods.

## 5 Experimental Evaluation and Discussion

In this section, we evaluate our proposed calibration methods for the tasks of object detection, instance segmentation, and semantic segmentation using pretrained neural networks. For calibration evaluation, we use the MS COCO validation dataset [LMB+14] consisting of 5,000 images with 80 different object classes for object detection and instance segmentation. For semantic segmentation, we use the panoptic segmentation annotations consisting of 171 different object and stuff categories in total. Our investigations are limited to the validation dataset since the training set has already been used for network training and no ground-truth annotations are available for the respective test dataset. Thus, we use a 50%/50% random split and use the first set for calibration training, while the second set is used for the evaluation of the calibration methods. Furthermore, we also utilize the Cityscapes validation dataset

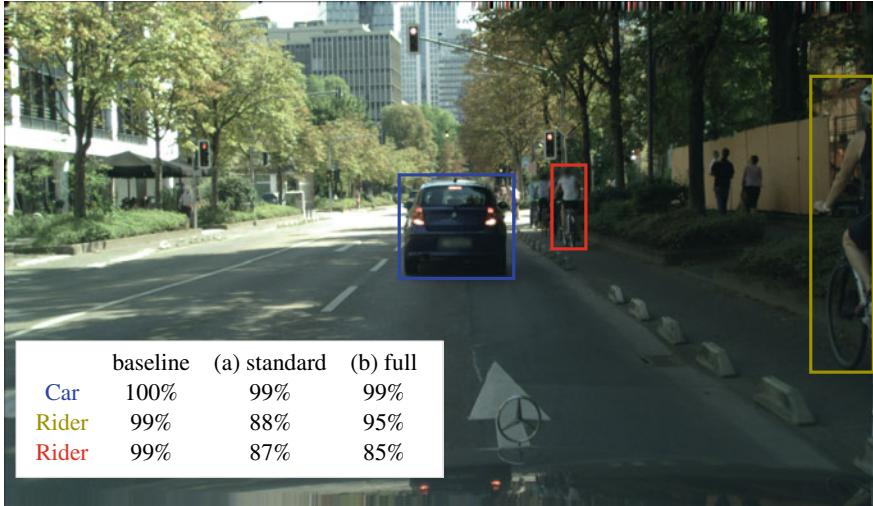
[COR+16] consisting of 500 images with 19 different classes that are used for model training. The Munster and Lindau images are used for calibration training, whereas the Frankfurt images are held back for calibration evaluation.

## 5.1 Object Detection

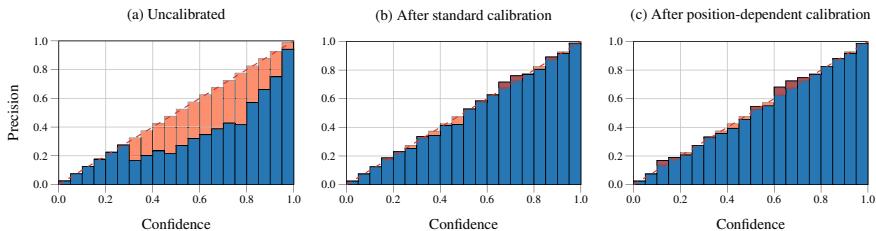
We evaluate our proposed calibration methods *histogram binning* (HB) (14), *logistic calibration* (LC) (21) and *beta calibration* (BC) (23) for the task of object detection using a pretrained Faster R-CNN X101-FPN [RHGS15, WKM+19] as well as a pretrained RetinaNet [LGG+17, WKM+19] on the MS COCO validation set. For Cityscapes, we use a pretrained Mask R-CNN R50-FPN [HGDG17, WKM+19], where only the predicted bounding box information is used for calibration. For calibration evaluation, we use the proposed D-ECE with the same subsets of data that have been used for calibration training. Thus, we use a binning scheme of  $B = 20$  using the confidence  $\hat{p}$  only. In contrast, we use  $B_q = 5$  for  $Q = 5$  when all auxiliary information is used. Each bin with less than 8 samples is neglected to increase D-ECE robustness. We further measure the *Brier score* (BS) and the *negative log likelihood* (NLL) of each model to evaluate its calibration properties. It is of special interest to assess if calibration has an influence to the precision/recall. Thus, we also denote the *area under precision/recall curve* (AUPRC) for each model. As opposed to previous examinations [KKSH20], we evaluate calibration using *all* available classes within each dataset. Each of these scores is measured for each class separately. In our experiments, we denote weighted average scores that are weighted by the amount of samples for each class. We perform calibration using IoU scores of 0.50 and 0.75, respectively. Furthermore, only bounding boxes with a score over 0.3 are used for calibration to reduce the amount of non-informative predictions. We give a qualitative example Fig. 2 that illustrates the effect of confidence calibration in the scope of object detection.

In our experiments, we first apply the standard calibration methods (Table 1) and compare the results with the baseline miscalibration of the detection model. We can already observe a default miscalibration of each examined network. This miscalibration is reduced by standard calibration where the scaling methods offer the best performance compared to the histogram binning. In a next step, we apply our box-sensitive calibration that includes the confidence  $\hat{p}$ , position  $c_x, c_y$  and shape  $w$  and  $h$  into a calibration mapping (Table 2). Similar to the confidence-only case in Table 1, the scaling methods consistently outperform baseline and histogram binning calibration.

In each calibration case, we observe a miscalibration of the base network that is alleviated using our proposed calibration methods. By examining the reliability diagram (Fig. 3), we observe that the networks are consistently miscalibrated for all confidence levels. This can be alleviated either by standard calibration or by position-dependent calibration. By examining the position-dependence of the



**Fig. 2** Qualitative example of position-dependent confidence calibration on a Cityscapes image with detections obtained by a Mask-RCNN. First, standard calibration is performed using similar confidence estimates  $\hat{p}$  without any position information. This results in rescaled but still similar calibrated confidence estimates (note that we have distinct calibration models for each class). Second, we can observe the effect of position-dependent calibration using the full position information (position and shape) as the two riders with similar confidence are rescaled differently. When using all available information, we can observe a significant difference in calibration since both, position and shape, have an effect to the calibration



**Fig. 3** Reliability diagram (object detection) for a Mask-RCNN class *pedestrian* inferred on Cityscapes Frankfurt images measured using the confidence only. The blue bar denotes the precision of samples that fall into this bin, with the gap to perfect calibration (diagonal) highlighted in red. **a** The Mask-RCNN model is consistently overconfident for each confidence level. **b** This can be alleviated by standard logistic calibration **c** as well as by position-dependent logistic calibration

**Table 1** Calibration results for object detection using the confidence  $\hat{p}$  only. The best scores are highlighted in bold. All calibration methods are able to improve miscalibration. The best performance is achieved using the scaling methods. Unlike histogram binning, the scaling methods are monotonically increasing and thus do not affect the AUPRC score

Network	Dataset	IoU	Cal. method	D-ECE (%)	Brier	NLL	AUPRC
Faster R-CNN	COCO	0.50	Baseline	17.839	0.206	0.622	0.833
			HB	5.622	0.166	0.598	0.781
			LC	4.449	<b>0.158</b>	<b>0.478</b>	0.833
			BC	<b>4.441</b>	<b>0.158</b>	0.482	0.833
		0.75	Baseline	29.721	0.272	0.861	0.767
			HB	5.173	0.158	0.594	0.676
			LC	<b>4.387</b>	<b>0.147</b>	<b>0.461</b>	0.766
			BC	4.540	<b>0.147</b>	0.463	0.766
RetinaNet	COCO	0.50	Baseline	10.061	0.171	0.514	0.831
			HB	5.332	0.166	0.579	0.795
			LC	4.464	<b>0.161</b>	<b>0.486</b>	0.831
			BC	<b>4.443</b>	<b>0.161</b>	0.487	0.831
		0.75	Baseline	17.543	0.181	0.544	0.765
			HB	5.773	0.147	0.551	0.721
			LC	4.395	<b>0.142</b>	<b>0.447</b>	0.765
			BC	<b>4.361</b>	<b>0.142</b>	<b>0.447</b>	0.765
Mask R-CNN	Cityscapes	0.50	Baseline	10.822	0.146	0.500	0.951
			HB	3.516	0.133	0.491	0.902
			LC	<b>3.305</b>	<b>0.125</b>	<b>0.380</b>	0.951
			BC	3.306	<b>0.125</b>	0.381	0.951
		0.75	Baseline	29.530	0.271	1.063	0.893
			HB	<b>3.718</b>	0.161	0.547	0.755
			LC	4.399	<b>0.136</b>	<b>0.425</b>	0.893
			BC	4.299	<b>0.136</b>	0.426	0.893

miscalibration (Figs. 4 and 5), a considerable increase of the calibration error toward the image boundaries can be observed. This calibration error is already well mitigated by standard calibration methods and can be further improved by position-dependent calibration. Also note that the AUPRC is not affected by standard scaling methods (logistic/beta calibration) as these methods perform a monotonically increasing mapping of the confidence estimates and thus do not affect the order of the samples. However, this is not the case with histogram binning which may lead to a significant drop of the AUPRC. Furthermore, even the position-dependent scaling methods cannot guarantee a monotonically increasing mapping. However, compared to the improvement of the calibration, the impact on the AUPRC is marginal. Therefore,

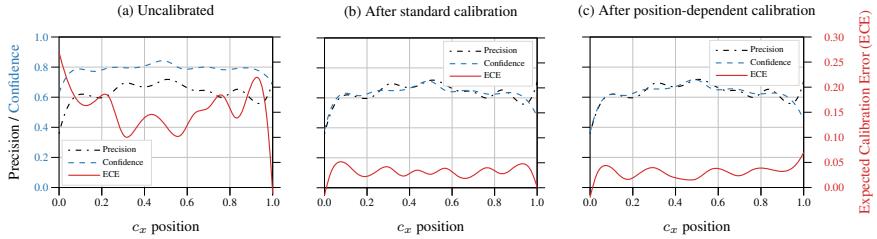
**Table 2** Calibration results for object detection using all available information  $\hat{p}, c_x, c_y, w$  and  $h$ . Similar to the confidence-only case in Table 1, the scaling methods consistently outperform baseline and histogram binning calibration. However, in the position-dependent case, the calibration mapping is not monotonically increasing. This has an effect to the AUPRC scores, but is marginal compared to the improvement in calibration

Network	Dataset	IoU	Cal. method	D-ECE [%]	Brier	NLL	AUPRC
Faster R-CNN	COCO	0.50	Baseline	7.519	0.206	0.622	<b>0.833</b>
			HB	3.891	0.206	1.076	0.701
			LC	<b>3.100</b>	0.175	0.627	0.807
			BC	3.240	<b>0.168</b>	<b>0.508</b>	0.807
		0.75	Baseline	13.277	0.272	0.861	<b>0.767</b>
			HB	3.848	0.200	1.037	0.587
			LC	<b>2.965</b>	0.162	0.587	0.730
			BC	3.217	<b>0.159</b>	<b>0.492</b>	0.719
RetinaNet	COCO	0.50	Baseline	4.303	<b>0.171</b>	<b>0.514</b>	<b>0.831</b>
			HB	3.369	0.199	1.100	0.727
			LC	<b>3.124</b>	0.177	0.615	0.804
			BC	3.491	0.175	0.521	0.800
		0.75	Baseline	6.724	0.181	0.544	<b>0.765</b>
			HB	3.241	0.176	1.026	0.640
			LC	<b>2.872</b>	0.157	0.595	0.732
			BC	3.205	<b>0.153</b>	<b>0.479</b>	0.725
Mask R-CNN	Cityscapes	0.50	Baseline	10.168	0.146	0.500	<b>0.951</b>
			HB	5.241	0.151	0.535	0.854
			LC	<b>4.551</b>	<b>0.134</b>	0.440	0.946
			BC	6.117	<b>0.134</b>	<b>0.413</b>	0.925
		0.75	Baseline	27.066	0.271	1.063	0.893
			HB	8.062	0.195	0.607	0.681
			LC	<b>6.267</b>	<b>0.140</b>	<b>0.464</b>	<b>0.894</b>
			BC	9.427	0.160	0.491	0.833

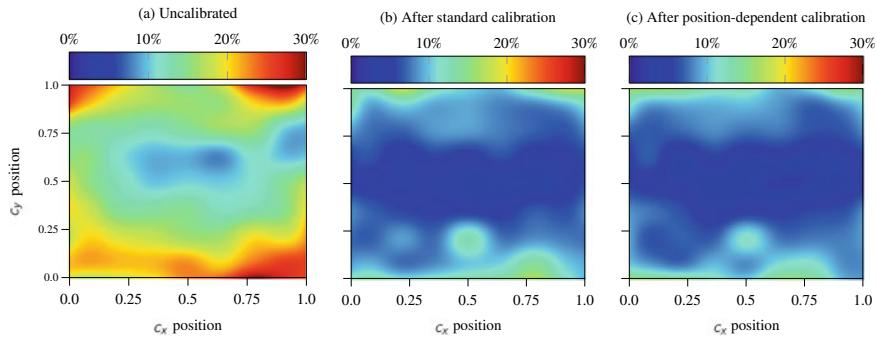
we conclude that our calibration methods are a valuable contribution especially for safety-critical systems, since they lead to statistically better calibrated confidences.

## 5.2 Instance Segmentation

After object detection, we investigate the calibration properties of instance segmentation models. According to the definition of calibration for segmentation models in (9), we can use each pixel within a segmentation mask as a separate prediction.



**Fig. 4** Reliability diagram (object detection) for a Mask-RCNN class *pedestrian* inferred on Cityscapes Frankfurt images measured by the confidence and the relative  $c_x$  position of each bounding box. **a** We observe an increasing calibration error toward the boundaries of the images. **b** This can be alleviated by standard logistic calibration **c** as well as by position-dependent logistic calibration



**Fig. 5** Reliability diagram (object detection) for a Mask-RCNN class *pedestrian* measured using all available information. This diagram shows the D-ECE [%] for all image regions. **a** We can observe a default miscalibration that increases toward the image boundaries. **b** Standard logistic calibration already achieves good calibration results, **c** that can be further improved by position-dependent calibration

This alleviates the problem of limited data availability and allows for a more robust calibration evaluation. Recently, Kumar et al. [KLM19] started a discussion about the sample-efficiency of binning and scaling methods. The authors show that binning methods yield a more robust calibration mapping for large datasets but also tend to overfit for a small database. We can confirm this observation as histogram binning provides poor calibration performance in our examinations on object detection calibration, particularly for classes with fewer samples. In contrast, scaling methods are more sample-efficient but also more inaccurate [KLM19]. Furthermore, scaling methods are computationally more expensive as they require an iterative update of the calibration parameters over the complete dataset, whereas a binning of samples comes at low computational costs, especially for large datasets. Therefore, our examinations are focused on the calibration performance of a (multivariate) histogram binning using 15 bins for each dimension. For inference, we use pretrained Mask R-CNN [HGDG17, WKM+19] as well as pretrained PointRend [KWHG20] models to obtain predictions of instance segmentation masks for both datasets. As within

object detection evaluation, all objects with a bounding box score below 0.3 are neglected. We perform standard calibration using the confidence only, as well as a position-dependent calibration including the  $x$  and  $y$  position of each pixel. The pixel position is scaled by the mask's bounding box size to get position information in the  $[0, 1]$  interval. Furthermore, we also include the distance of each pixel to the nearest mask segment boundary as a feature in a calibration mapping since we expect a higher uncertainty especially at the segment boundaries. The distance is normalized by the bounding box's diagonal to obtain distance values in  $[0, 1]$ .

Since many data samples are available, we measure calibration using a D-ECE with 15 bins neglecting each bin with less than 8 samples. We further assess the Brier score as well as the NLL loss as complementary metrics. The task of instance segmentation is related to object detection. In a first step, the network also needs to infer a bounding box for each detected object. Similar to the calibration evaluation for object detection, we further use IoU scores of 0.50 and 0.75 to specify whether a prediction has matched a ground-truth object or not. In contrast to object detection, the IoU score within instance segmentation is computed by the overlap of the inferred segmentation mask and the according ground-truth object. Using this definition, we can compute the AUPRC to evaluate the average quality of the object detection branch as well as the according segmentation masks. All results for instance segmentation calibration for IoU of 0.50 and 0.75 are shown in Tables 3 and 4, respectively. These tables compare standard calibration (subset: confidence only) with position-dependent calibration (subset: full). We observe a significant miscalibration of the networks by default. Using standard calibration or our calibration methods, it is possible to improve the calibration score D-ECE, Brier, and NLL for each case. The miscalibration is successfully corrected by histogram binning using either the standard binning or the position-dependent binning. This is also underlined by inspecting the reliability diagrams for the confidence-only case (Fig. 6). Furthermore, we can also observe miscalibration that is dependent on the relative  $x$  and  $y$  pixel position within a mask (Figs. 7 and 8). We can observe that standard histogram binning calibration already achieves good calibration results but also offers a weak dependency on the pixel position as well. Although the position-dependent calibration only achieves a minor improvement in calibration compared to the standard calibration case, it shows an equal calibration performance across the whole image.

In contrast to object detection, we observe a slightly increased miscalibration toward the center of a mask. As it can be seen in Fig. 9, most pixels belonging to an object mask are also located in the center. This underlines the need for a position-dependent calibration. Interestingly, although position-dependent calibration does not seem to offer better Brier or NLL scores compared to the confidence-only case, it significantly improves the mask quality as we can observe higher AUPRC scores for position-dependent calibration. We further provide a qualitative example that illustrates the effect of confidence calibration in Fig. 9. In this example, we can see the difference between standard calibration and position-dependent calibration. In the former case, the mask scores are only rescaled by their confidence which might lead to a better calibration but sometimes also to unwanted losses of mask segments.

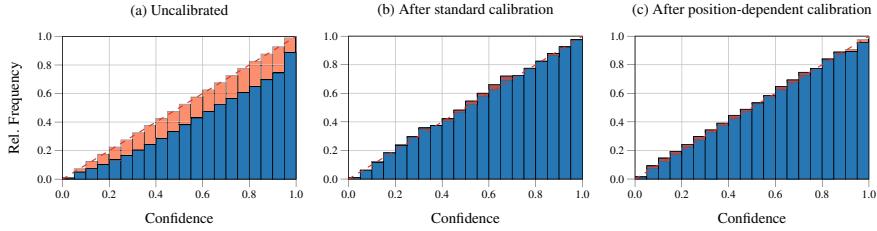
**Table 3** Calibration results for instance segmentation @ IoU=0.50. The best D-ECE scores are underlined for each subset separately, since it is not convenient to compare D-ECE scores with different subsets to each other [KKSH20]. Furthermore, the best Brier, NLL, and AUPRC scores are highlighted in bold. These scores are only calculated using the confidence information and thus can be compared to each other. The histogram binning calibration consistently improves miscalibration. Furthermore, we find that although position dependence does not improve the Brier and NLL scores as in the standard case, it leads to a significant improvement in the mIoU score

Network	Dataset	Cal. method	Subset: confidence only				Subset: full			
			D-ECE [%]	Brier	NLL	AUPRC	D-ECE [%]	Brier	NLL	AUPRC
Mask R-CNN	CS	Baseline	7.088	0.110	0.432	0.724	11.205	0.110	0.432	0.724
		HB	<u>5.661</u>	<b>0.099</b>	<b>0.320</b>	0.723	<u>10.119</u>	0.117	0.530	<b>0.787</b>
	COCO	Baseline	21.983	0.222	0.940	0.663	23.409	0.222	0.940	0.663
		HB	<u>6.420</u>	<b>0.150</b>	<b>0.442</b>	0.662	<u>13.640</u>	0.171	0.776	<b>0.760</b>
PointRend	CS	Baseline	12.893	0.160	0.785	0.709	20.858	0.160	0.785	0.709
		HB	<u>2.706</u>	<b>0.105</b>	<b>0.326</b>	0.698	<u>19.021</u>	0.190	1.299	<b>0.758</b>
	COCO	Baseline	22.284	0.222	0.946	0.672	23.973	0.222	0.946	0.672
		HB	<u>6.348</u>	<b>0.144</b>	<b>0.428</b>	0.664	<u>16.060</u>	0.180	1.005	<b>0.751</b>

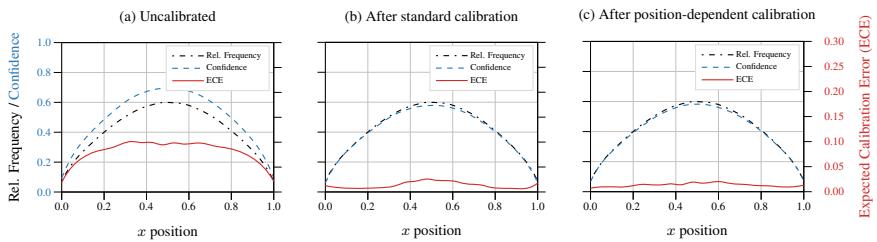
**Table 4** Calibration results for instance segmentation @ IoU=0.75. The best scores are highlighted. We observe the same behavior in calibration for all models as for the IoU=0.50 case shown in Table 3

Network	Dataset	Cal. method	Subset: confidence only				Subset: full			
			D-ECE [%]	Brier	NLL	AUPRC	D-ECE [%]	Brier	NLL	AUPRC
Mask R-CNN	CS	Baseline	12.862	0.147	0.622	0.375	14.470	0.147	0.622	0.375
		HB	<u>5.895</u>	<b>0.108</b>	<b>0.340</b>	0.375	<u>10.273</u>	0.120	0.523	<b>0.479</b>
	COCO	Baseline	26.559	0.250	1.070	0.237	27.219	0.250	1.070	0.237
		HB	<u>6.006</u>	<b>0.144</b>	<b>0.423</b>	0.235	<u>12.888</u>	0.165	0.720	<b>0.425</b>
PointRend	CS	Baseline	18.668	0.192	0.929	0.347	25.386	0.192	0.929	0.347
		HB	<u>3.899</u>	<b>0.115</b>	<b>0.349</b>	0.344	<u>18.415</u>	0.191	1.247	<b>0.486</b>
	COCO	Baseline	26.643	0.248	1.060	0.258	27.377	0.248	1.060	0.258
		HB	<u>6.708</u>	<b>0.138</b>	<b>0.411</b>	0.238	<u>15.344</u>	0.173	0.936	<b>0.388</b>

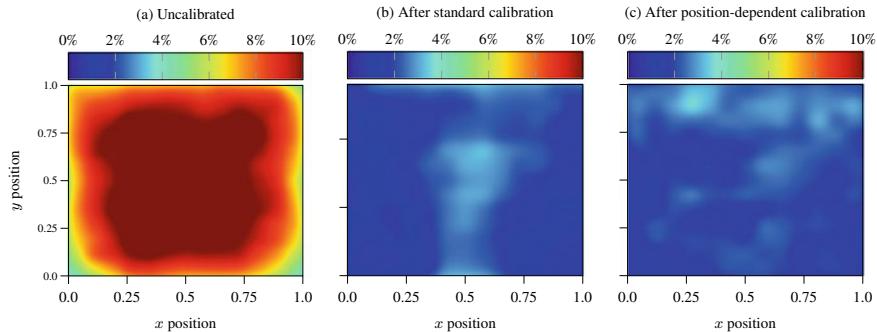
(especially small objects in the background). In contrast, position-dependent calibration is capable of possible correlations between confidence and pixel position or the size of an object. This leads to improved estimates of the mask confidences even for smaller objects. Therefore, we conclude that multidimensional confidence calibration has a positive influence on the calibration properties as well as on the quality of the object masks.



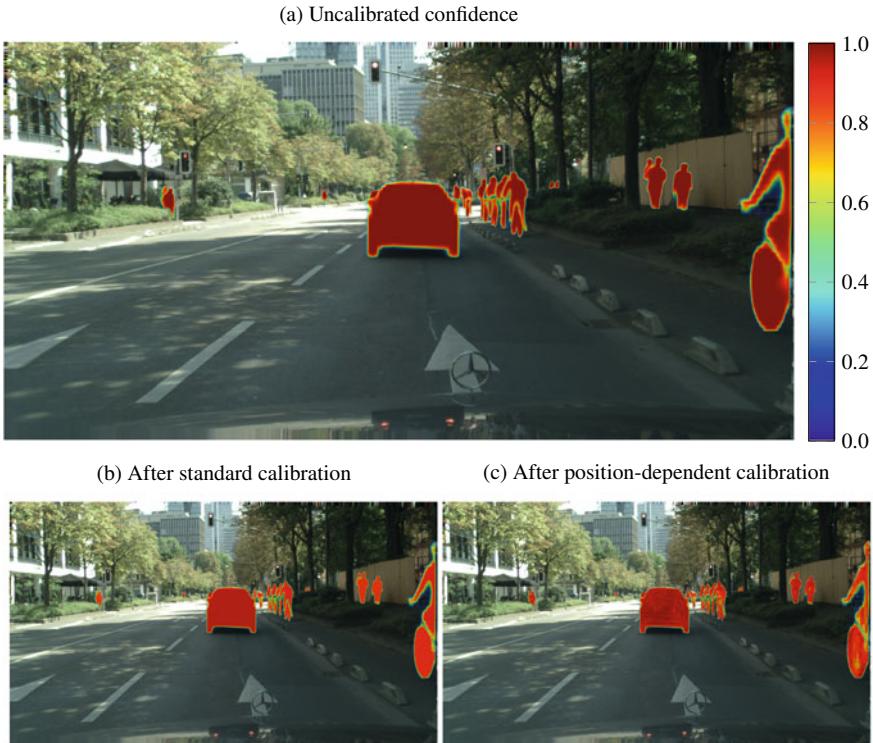
**Fig. 6** Reliability diagram (instance segmentation) for a Mask R-CNN class *pedestrian* inferred on MS COCO validation images measured using the confidence only. The blue bar denotes the frequency of samples that fall into this bin, with the gap to perfect calibration (diagonal) highlighted in red. **a** The Mask R-CNN model is consistently overconfident for each confidence level. **b** This can be alleviated by standard histogram binning calibration **c** as well as by position-dependent logistic calibration



**Fig. 7** Reliability diagram (instance segmentation) for a Mask R-CNN class *pedestrian* inferred on MS COCO validation images measured by the relative  $x$  position of each pixel within its bounding box. **a** Similar to the examinations for object detection, we observe an increasing calibration error toward the boundaries of the images. **b** Standard histogram binning significantly reduces the calibration error, **c** whereas position-dependent histogram binning is capable of a slightly further refinement of the position-dependent D-ECE



**Fig. 8** Reliability diagram (instance segmentation) for a Mask R-CNN class *pedestrian* inferred on MS COCO validation images. The diagram shows the D-ECE [%] that is measured using the relative  $x$  and  $y$  position of each pixel within its bounding box. Similar to the observations in Fig. 7, standard calibration already reduces the calibration error. However, we can observe a slightly increased calibration error toward the image's center. In contrast, position-dependent calibration error shows equal calibration performance across the whole image



**Fig. 9** Qualitative example of a Cityscapes image with instance segmentation masks obtained by a Mask R-CNN. **a** Uncalibrated confidence of the predicted masks ( $mIoU = 0.289$ ). **b** Instance masks are calibrated using standard histogram binning ( $mIoU = 0.343$ ). **c** Mask confidences using a position-dependent confidence calibration ( $mIoU = 0.370$ ). Unlike a simple scaling of confidences in **b**, the calibration in **c** offers a more fine-grained mapping that achieves a better fit to the true instance masks

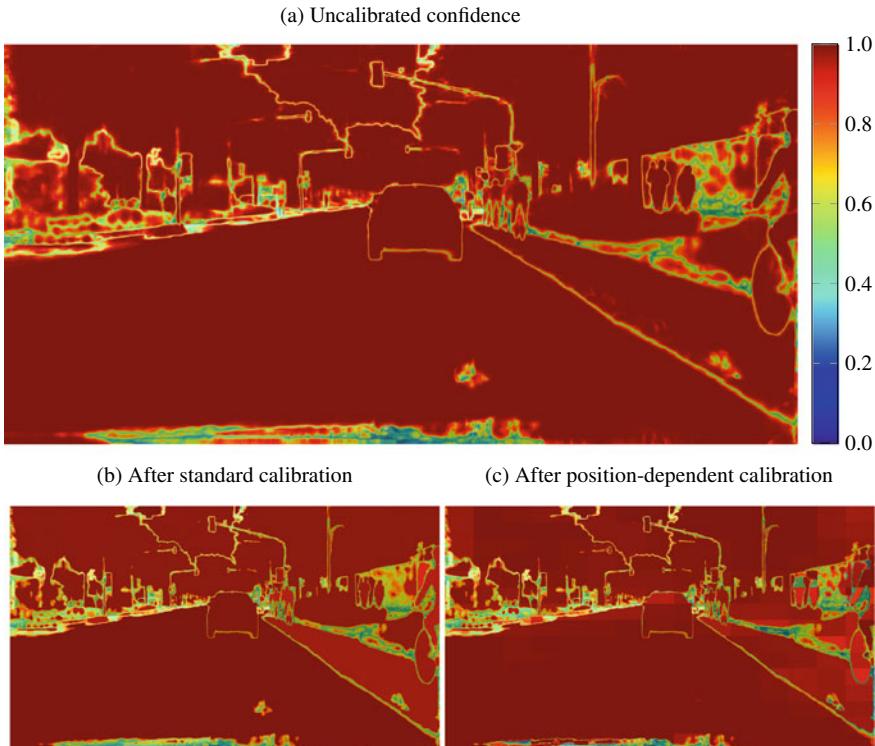
### 5.3 Semantic Segmentation

For the evaluation of semantic segmentation, we use the same datasets, the same binning scheme, and the same features that have already been used for the instance segmentation in Sect. 5.2. For COCO mask inference, we use a pretrained Deeplabv2 [CPK+18] as well as a pretrained HRNet model [SXLW19, WSC+20, YCW20]. For Cityscapes, we also use a HRNet as well as a pretrained Deeplabv3+ model [CZP+18]. Similar to the instance segmentation, we also use the D-ECE, Brier score, NLL loss, and  $mIoU$  to compare the baseline calibration with a histogram binning model. As opposed to our previous experiments, we only use 15% of the provided samples to reduce computational complexity. The results for default miscalibration, standard calibration (subset: confidence only) and position-dependent calibration (subset: full) are shown in Table 5. Unlike instance segmentation calibration, the

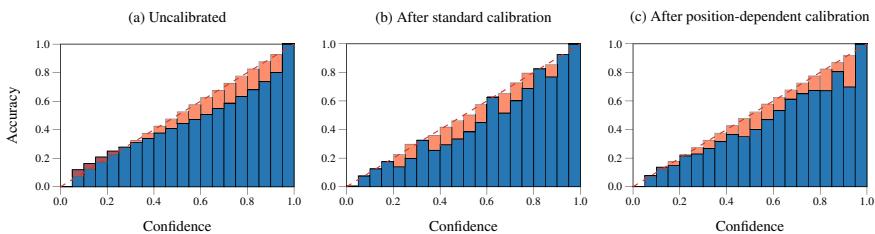
**Table 5** Calibration results for semantic segmentation. The best D-ECE scores are underlined for each subset separately, since it is not convenient to compare D-ECE scores with different subsets to each other [KKSH20]. Furthermore, the best Brier, NLL, and mIoU scores are highlighted in bold. These scores are only calculated using the confidence information and thus can be compared to each other. In contrast to object detection and instance segmentation evaluation, we observe a low baseline calibration error that is only slightly affected by confidence calibration

Network	Dataset	Cal. method	Subset: confidence only				Subset: full			
			D-ECE [%]	Brier	NLL	mIoU	D-ECE [%]	Brier	NLL	mIoU
Deeplabv3+	CS	Baseline	0.162	<b>0.060</b>	<b>0.139</b>	<b>0.623</b>	<u>0.186</u>	<b>0.060</b>	<b>0.139</b>	<b>0.623</b>
		HB	<u>0.081</u>	<b>0.060</b>	0.170	0.619	0.199	0.062	0.189	0.589
Deeplabv2	COCO	Baseline	0.094	0.458	<b>1.173</b>	<b>0.933</b>	<u>0.149</u>	0.458	<b>1.173</b>	<b>0.933</b>
		HB	<u>0.060</u>	<b>0.456</b>	1.515	<b>0.933</b>	0.150	0.485	1.790	0.913
HRNet	CS	Baseline	<u>0.067</u>	<b>0.057</b>	<b>0.115</b>	<b>0.629</b>	<u>0.149</u>	<b>0.057</b>	<b>0.115</b>	<b>0.629</b>
		HB	0.083	<b>0.057</b>	0.148	0.628	0.191	0.060	0.171	0.582
	COCO	Baseline	0.455	0.779	5.812	<b>0.939</b>	0.455	0.779	5.812	<b>0.939</b>
		HB	<u>0.062</u>	<b>0.563</b>	<b>2.261</b>	<b>0.939</b>	<u>0.138</u>	0.571	2.372	0.931

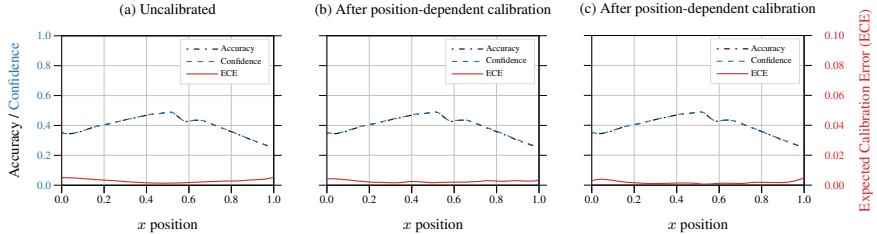
baseline model is already intrinsically well-calibrated, offering a very low calibration error. A qualitative example is shown in Fig. 10 to illustrate the effect of calibration for semantic segmentation. In this example, we can observe only minor differences between the uncalibrated mask and the masks either after standard calibration or after position-dependent calibration. This aspect is supported by the confidence reliability diagram shown in Fig. 11. Although we can observe an overconfidence, most samples are located in the low confident space with a low calibration error that also results in an overall low miscalibration score. Furthermore, our calibration methods are able to achieve even better calibration results, but mostly in the confidence-only calibration case. In addition, we also observe only a low correlation between position and calibration error (Figs. 12 and 13). One reason for the major difference between instance and semantic segmentation calibration may be the difference in model training. An instance segmentation model needs to infer an appropriate bounding box first to achieve qualitatively good results in mask inference. In contrast, a semantic segmentation model does not need to infer the position of objects within an image but is able to directly learn and improve mask quality. We also suspect an influence of the amount of available data points, since a semantic segmentation model is able to use each pixel within an image as a separate sample, whereas an instance segmentation model is only restricted to the pixels that are available within an estimated bounding box. Therefore, we conclude that semantic segmentation models do not require a post-hoc confidence calibration as they already offer a good calibration performance.



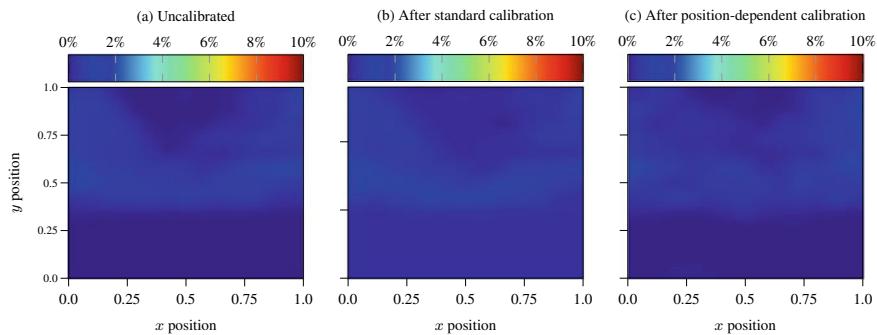
**Fig. 10** Qualitative example of a Cityscapes image with semantic segmentation masks obtained by a DeepLabv3+ model. **a** Uncalibrated confidence of the predicted masks ( $mIoU = 0.576$ ), **b** here, the pixel masks are calibrated using standard histogram binning ( $mIoU = 0.522$ ). Furthermore, **c** mask confidences using a position-dependent confidence calibration ( $mIoU = 0.570$ ). Although we observe some minor differences, we conclude that confidence calibration does not have a major influence to the calibration properties for semantic segmentation models



**Fig. 11** Reliability diagram (semantic segmentation) for a DeepLabv3+ class *pedestrian* inferred on Cityscapes Frankfurt images measured using the confidence only. The blue bar denotes the accuracy of samples that fall into this bin, with the gap to perfect calibration (diagonal) highlighted in red. By default **(a)**, the DeepLabv3+ model already offers a good calibration performance that is only slightly affected either by standard calibration **(b)** or by position-dependent calibration **(c)**



**Fig. 12** Reliability diagram (semantic segmentation) for a Deeplabv3+ class *pedestrian* inferred on Cityscapes Frankfurt images measured by the relative  $x$  position of each pixel. Since the model has a low miscalibration by default **(a)**, the effect of calibration in **b** or **c** is marginal



**Fig. 13** Reliability diagram (semantic segmentation) for a Deeplabv3+ class *pedestrian* inferred on Cityscapes Frankfurt images. This diagram shows the D-ECE [%] that is measured using the relative  $x$  and  $y$  position of each pixel. Since the model has a low miscalibration by default **(a)**, the effect of calibration in **b** or **c** is marginal

## Conclusions

Within the scope of confidence calibration, recent work has mainly focused on the task of classification calibration. In this chapter, we presented an analysis of confidence calibration for object detection calibration as well as, for instance, and semantic segmentation calibration. Firstly, we introduced definitions for confidence calibration within the scope of object detection, instance segmentation, and semantic segmentation. Secondly, we presented methods to measure and alleviate miscalibration of detection and segmentation networks. These methods are extensions of well-known calibration methods such as histogram binning [ZE01], Platt scaling [Pla99], and beta calibration [KSFF17]. We extend these methods so that they encompass additional calibration information such as position and shape. Finally, the experiments revealed that common object detection models as well as instance segmentation networks tend to miscalibration. In addition, we showed that auxiliary information such as estimated position or shape of a predicted object also have an influence to confidence calibration. However, we also found that semantic segmentation models are

already intrinsically calibrated. Thus, the examined models do not require additional post-hoc calibration and already offer well-calibrated mask confidence scores. We argue that this difference between instance and semantic segmentation is a result of data quality and availability during training. This leads to the assumption that limited data availability is a direct cause for miscalibration during training and thus an effect of overfitting. Nevertheless, our proposed calibration framework is capable to calibrate object detection and instance segmentation models. In safety-critical applications, the confidence in the applied algorithms is paramount. The proposed calibration algorithms allow to detect situations of low confidence and thus perform the appropriate system reaction. Therefore, calibrated confidence values can be used as additional information especially in safety-critical applications.

**Acknowledgements** The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project “Methoden und Maßnahmen zur Absicherung von KI-basierten Wahrnehmungsfunktionen für das automatisierte Fahren (KI Absicherung)”. The authors would like to thank the consortium for the successful cooperation.

## References

- [COR+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV, USA, June 2016), pp. 3213–3223
- [CPK+18] Liang-Chieh. Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **40**(4), 834–848 (2018)
- [CZP+18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich, Germany, September 2018), pp. 833–851
- [DLXS20] Y. Ding, J. Liu, J. Xiong, Y. Shi, Revisiting the evaluation of uncertainty estimation and its application to explore model complexity-uncertainty trade-off, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* virtual conference (June 2020), pp. 22–31
- [GPSW17] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in *Proceedings of the International Conference on Machine Learning (ICML)* (Sydney, NSW, Australia, August 2017), pp. 1321–1330
- [HGDG17] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Venice, Italy, October 2017), pp. 2980–2988
- [JJY+19] B. Ji, H. Jung, J. Yoon, K. Kim, Y. Shin, Bin-wise temperature scaling (BTS): improvement in confidence calibration performance through simple scaling techniques, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops* (Seoul, Korea, October 2019), pp. 4190–4196
- [KG20] D. Karimi, A. Gholipour, Improving Calibration and Out-of-Distribution Detection in Medical Image Segmentation With Convolutional Neural Networks (May 2020), pp. 1–12. [arXiv:2004.06569](https://arxiv.org/abs/2004.06569)

- [KKSH20] F. Küppers, J. Kronenberger, A. Shantia, A. Haselhoff, Multivariate confidence calibration for object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, virtual conference (June 2020), pp. 1322–1330
- [KLM19] A. Kumar, P.S. Liang, T. Ma, Verified uncertainty calibration, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Vancouver, BC, Canada, December 2019), pp. 3787–3798
- [KPNK+19] M. Kull, M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, P. Flach, Beyond temperature scaling: obtaining well-calibrated multi-class probabilities with Dirichlet calibration, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Vancouver, BC, Canada, December 2019), pp. 12316–12326
- [KSFF17] M. Kull, T. Silva Filho, P. Flach, Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers, in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)* (Fort Lauderdale, FL, USA, May 2017), pp. 623–631
- [KWHG20] A. Kirillov, Y. Wu, K. He, R. Girshick, PointRend: image segmentation as rendering, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference (June 2020), pp. 9799–9808
- [LGG+17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Venice, Italy, October 2017), pp. 2980–2988
- [LMB+14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. Lawrence Zitnick, Microsoft COCO: common objects in context, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Zurich, Switzerland, September 2014), pp. 740–755
- [LN82] David L. Libby, Melvin R. Novick, Multivariate generalized beta distributions with applications to utility assessment. *J. Educ. Stat.* **7**(4), 271–294 (1982)
- [MWT+20] Alireza Mehrtash, William M. Wells, Clare M. Tempany, Purang Abolmaesumi, Tina Kapur, Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. *IEEE Trans. Med. Imaging* **39**(12), 3868–3878 (2020)
- [NC16] M. Naeini, G. Cooper, Binary classifier calibration using an ensemble of near isotonic regression models, in *Proceedings of the IEEE International Conference on Data Mining (ICDM)* (Barcelona, Spain, December 2016), pp. 360–369
- [NCH15] M. Pakdaman Naeini, G. Cooper, M. Hauskrecht, Obtaining well calibrated probabilities using Bayesian binning, in *Proceedings of the AAAI Conference on Artificial Intelligence* (Austin, TX, USA, January 2015), pp. 2901–2907
- [NDZ+19] J. Nixon, M.W. Dusenberry, L. Zhang, G. Jerfel, D. Tran, Measuring calibration in deep learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (Long Beach, CA, USA, June 2019), pp. 38–41
- [Pla99] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in *Advances in Large Margin Classifiers*. ed. by A.J. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (MIT Press, 1999), pp. 61–74
- [RHGS15] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Montréal, QC, Canada, December 2015), pp. 91–99
- [SKR+21] F. Schwaiger, M. Henne Fabian Küppers, F. Schmoeller Roza, K. Roscher, A. Haselhoff, From black-box to white-box: examining confidence calibration under different conditions, in *Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI)*, virtual conference (February 2021), pp. 1–8
- [SXLW19] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA, USA, June 2019), pp. 5693–5703

- [WKM+19] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick, Detectron2 (2019). [Online; assessed 2021-11-18]
- [WLK+20] T. Wang, Y. Li, B. Kang, J. Li, J. Liew, S. Tang, S. Hoi, J. Feng, The devil is in classification: a simple framework for long-tail instance segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, virtual conference (August 2020), pp. 728–744
- [WSC+20] Jingdong Wang, Ke. Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Mu. Yadong, Mingkui Tan, Xinggang Wang et al., Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **43**(10), 3349–3364 (2020)
- [YCW20] Y. Yuan, X. Chen, J. Wang, Object-contextual representations for semantic segmentation, in *Proceedings of the European Conference on Computer Vision (ECCV)*, virtual conference (August 2020), pp. 173–190
- [ZE01] B. Zadrozny, C. Elkan, Obtaining calibrated probability estimates from decision trees and Naive Bayesian classifiers, in *Proceedings of the International Conference on Machine Learning (ICML)* (Williamstown, MA, USA, June 2001), pp. 609–616
- [ZE02] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (Edmonton, AB, Canada, July 2002), pp. 694–699

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.



# Uncertainty Quantification for Object Detection: Output- and Gradient-Based Approaches



Tobias Riedlinger, Marius Schubert, Karsten Kahl, and Matthias Rottmann

**Abstract** Safety-critical applications of deep neural networks require reliable confidence estimation methods with high predictive power. However, evaluating and comparing different methods for uncertainty quantification is oftentimes highly context-dependent. In this chapter, we introduce flexible evaluation protocols which are applicable to a wide range of tasks with an emphasis on object detection. In this light, we investigate uncertainty metrics based on the network output, as well as metrics based on a learning gradient, both of which significantly outperform the confidence score of the network. While output-based uncertainty is produced by post-processing steps and is computationally efficient, gradient-based uncertainty, in principle, allows for localization of uncertainty within the network architecture. We show that both sources of uncertainty are mutually non-redundant and can be combined beneficially. Furthermore, we show direct applications of uncertainty quantification by improving detection accuracy.

## 1 Introduction

Deep artificial neural networks (DNNs) employed for tasks such as object detection or semantic segmentation yield by construction probabilistic predictions on feature data, such as camera images. Modern deep object detectors [RHGS15, LAE+16, LGG+17, RF18] predict bounding boxes for depicted objects of a given set of learned classes. The so-called “score” (sometimes also called objectness or confidence score)

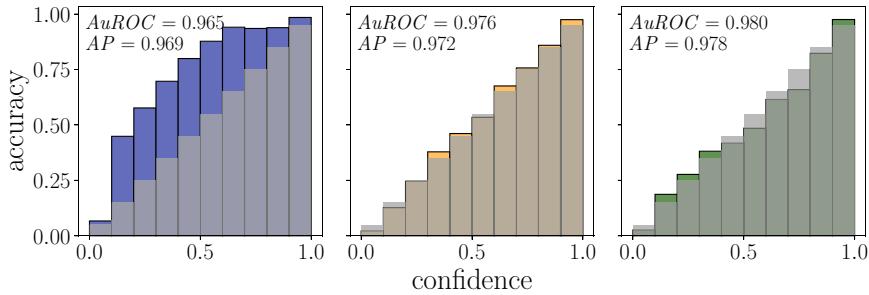
---

T. Riedlinger (✉) · M. Schubert · K. Kahl · M. Rottmann  
University of Wuppertal, Gaußstr. 20, 42119 Wuppertal, Germany  
e-mail: [riedlinger@uni-wuppertal.de](mailto:riedlinger@uni-wuppertal.de)

M. Schubert  
e-mail: [schubert@uni-wuppertal.de](mailto:schubert@uni-wuppertal.de)

K. Kahl  
e-mail: [kkahl@uni-wuppertal.de](mailto:kkahl@uni-wuppertal.de)

M. Rottmann  
e-mail: [rottmann@uni-wuppertal.de](mailto:rottmann@uni-wuppertal.de)



**Fig. 1** Confidence calibration plots for score (left), Monte-Carlo (MC) dropout uncertainty (center), and our proposed gradient uncertainty metrics G (right) on  $\mathcal{D}_{\text{eval}}^{\text{KITTI}}$ . For the sake of reference, we provide the optimally calibrated diagonal in gray. See Sect. 4 for more details

indicates the probability that a bounding box provided by the DNN contains an object. Throughout this chapter, we sometimes use the term “confidence” instead of “score” to refer to quantities that represent the probability of a detection being correct. The term confidence in a broader sense is meant to reflect an estimated degree of competency of the model when evaluating an input.

Reliable and accurate probabilistic predictions are desirable for many applications, such as medical diagnosis or automated driving. It is well-known that predictions of DNNs often tend to be statistically *miscalibrated* [SZS+14, GSS15, GPSW17], i.e., the score computed by the DNN is not representative for the relative frequency of correct predictions. For an illustration of this, see Fig. 1, where we compare different sources of uncertainty in terms of their accuracy conditioned to confidence bins for a YOLOv3 model.

For individual DNN predictions, confidences that are not well-calibrated are misleading and constitute a reliability issue. Over-confident predictions might cause inoperability of a perception system due to producing non-existent instances (false positives / FP). Perhaps even more adverse, under-confidence might lead to false negative (FN) predictions, for instance overlooking pedestrians or other road users which could lead to dangerous situations or even fatalities. Given that the set of available data features is fixed, the sources of uncertainty in machine learning (and deep learning) can be divided into two types [HW21], referring to their primary source [Gal17, KG17]. Whereas *aleatoric* uncertainty refers to the inherent and irreducible stochasticity of the distribution the data stems from, *epistemic* uncertainty refers to the reducible part of the uncertainty. The latter originates from the finite size of a random data sample used for training, as well as from the choice of model and learning algorithm.

Up to now, there exist a moderate number of established methods in the field of deep learning that have demonstrated to properly estimate the uncertainty of DNNs. In [KG17], DNNs for semantic segmentation are equipped with additional regression output that model heteroscedastic (in machine learning also called predictive) uncertainty, with the aim to capture aleatoric uncertainty. For training, the uncer-

tainty output is integrated into the usual empirical cross-entropy loss. A number of adaptations for different modern DNN architectures originate from that work. From a theoretical point of view, Bayesian DNNs [DL90, Mac92], in which the model weights are treated as random variables, yield an attractive framework for capturing the epistemic uncertainty. In practice, the probability density functions corresponding to the model weights are estimated via Markov chain Monte-Carlo, which currently is infeasible already for a moderate number of model weights, hence being inapplicable in the setting of object detection. As a remedy, variational inference approaches have been considered, where the model weights are sampled from predefined distributions, which are often modeled in simplified ways, e.g., assuming mutual independence. A standard method for the estimation of epistemic uncertainty based on variational inference is Monte-Carlo (MC) dropout [SHK+14, GG16]. Algorithmically, this method performs several forward passes under dropout at inference time. Deep ensemble sampling [LPB17] follows a similar idea. Here, separately trained models with the same architecture are deployed to produce a probabilistic inference.

In semantic segmentation, Rottmann et al. [RCH+20] introduced the tasks of meta classification and regression that are both applied as post-processing to the output of DNNs. Meta classification refers to the task of classifying a prediction as true positive (TP) or FP. Meta regression is a more object detection-specific task with the aim to predict the intersection-over-union (*IoU*) of a prediction with its ground truth directly. These tasks are typically learned with the help of ground truth, but afterwards performed in the absence of ground truth, only on the basis of uncertainty measures stemming from the output of the DNN. The framework developed in [RCH+20], called MetaSeg, was extended into multiple directions: Time-dynamic uncertainty quantification for semantic segmentation [MRG20] and instance segmentation [MRV+21]; a study on the influence of image resolution [RS19]; controlled reduction of FN and meta fusion, the latter utilizes the uncertainty metrics to increase the performance of the DNN [CRH+19]; out-of-distribution detection [BCR+20, ORF20, CRG21]; and active learning [CRGR21]. Last but not least, a similar approach solely based on the output of the DNN has been developed for object detection in [SKR20] (called MetaDetect), however, equipped with different uncertainty metrics specifically designed for the task of object detection. This approach was also used in an uncertainty-aware sensor fusion approach for object detection based on camera images and RaDAR point clouds; see [KRBG21].

In [ORG18], it was proposed to utilize a learning gradient, replacing the usually needed ground truth labels by the prediction of the network itself, which is supposed to contain epistemic uncertainty information. Since the investigation of gradient metrics for image classification tasks, that method was considered in different applications as well. Gradient metrics were compared to other uncertainty quantification methods in [SFKM20]. It was found that gradient uncertainty is strongly correlated with the softmax entropy of classification networks. In natural language understanding, gradient metrics turned out to be beneficial; see [VSG19]. Therein, gradient metrics and deep ensemble uncertainty were aggregated and yield well-calibrated confidences on out-of-distribution data. Recently, gradient metrics have

been developed for object detection in [RRSG21]. In order to demonstrate the predictive power of uncertainty metrics, the framework of meta classification and regression provided by MetaDetect was used.

In this chapter, we review the works published in [SKR20, RRSG21] and put them into a common context. More precisely, our contributions are as follows:

- We review the concepts of meta classification and regression, meta fusion, and confidence calibration. We explain how they serve as a general benchmark for evaluating the predictive power of any uncertainty metric developed for object detection.
- We review output-based uncertainty metrics developed in [SKR20], as well as gradient-based ones from inside the network [RRSG21].
- We compare baseline uncertainty measures such as the DNN’s score, well-established ones like Monte-Carlo dropout [Gal17], the output-based uncertainty metrics [SKR20], and finally the gradient-based uncertainty metrics [RRSG21] from inside the DNN. We do so in terms of comparing their standalone performances but also in terms of analyzing their mutual information and how much performance they add to the prediction of the network itself.

## 2 Related Work

In this section, we gather and discuss previous research related to the present work.

**Epistemic and aleatoric uncertainty in object detection:** In recent years, there has been increasing interest in methods to properly estimate or quantify uncertainty in DNNs. *Aleatoric uncertainty*, the type of uncertainty resulting from the data generating process, has been proposed to be captured by providing additional regression output to the network architecture for learning uncertainty directly from the training dataset [KG17]. Since the initial proposition, this approach has been adapted to, among others, the object detection setting [LDBK18, CCKL19, HZW+19, LHK+21, HSW20, CCLK20]. The latter approaches add further regression variables for aleatoric uncertainty which are assigned to their bounding box localization variables, thereby learning localization as Gaussian distributions. An alternative approach to quantifying localization uncertainty is to learn for each predicted box the corresponding *IoU* with additional regression variables [JLM+18].

Several methods have been proposed that aim to capture *epistemic uncertainty*, i.e., “reducible” kinds of uncertainty. Monte-Carlo (MC) dropout [SHK+14, GG16] employs forward passes under dropout and has become one of the standard tools for estimating epistemic uncertainty. As such, MC dropout has been investigated also for deep object detection [MNDS18, KD19, HSW20]. Similarly to MC dropout, deep ensemble methods [LPB17] employ separately trained networks (“experts”) to obtain variational inference, an approach which also has found adaptations in deep object detection [LGRB20].

**Uncertainty calibration:** A large variety of methods have been proposed to rectify the intrinsic confidence miscalibration of modern deep neural networks by introducing post-processing methods (see, e.g., [GPSW17] for a comparison). Prominent examples of such methods include histogram binning [ZE02], isotonic regression [ZE01], Bayesian binning [NCH15], Platt scaling [Pla99, NMC05], or temperature scaling [HVD14]. Alternatively, calibrated confidences have been obtained by implementing calibration as an optimization objective [KP20] via a suitable loss term. In the realm of object detection, the authors of [NZV18] found that temperature scaling improves calibration. Also, natural extensions to make calibration localization-dependent have been proposed [KKSH20].

**Meta classification and meta regression:** Meta classification denotes the task of discriminating FP and FN predictions based on uncertainty or confidence information, an idea that has been explored initially in [HG17]. In case a real-valued metric (e.g., different versions of the  $IoU$  which is in  $[0, 1]$ ) can be assigned to the quality of a prediction, meta regression denotes the task of predicting this quantity in the same spirit as meta classification. In contrast to the IoUNet framework introduced in [JLM+18], meta classification and meta regression do not require architectural changes or dedicated training of the respective DNN. In [RRSG21], it has been found that meta classifiers tend to yield well-calibrated confidences by default as opposed to the confidence score of DNNs. Applications of meta classification and meta regression have since been introduced for several different disciplines [CRH+19, RS19, MRG20, RMC+20, RCH+20, MRV+21, KRBG21], including semantic image segmentation, video instance segmentation, and object detection. In all of these applications, the central idea is to use a simple classification model to learn the map between uncertainty and the binary labels TP/FP. A related idea trying to exploit meta classification was introduced as “meta fusion” in [CRH+19]. The ability to discriminate TPs against FPs allows for an increase in the neural network’s sensitivity, thereby producing an increase in FPs which are subsequently detected by a meta classifier decreasing the total number of errors made.

### 3 Methods

#### 3.1 Uncertainty Quantification Protocols

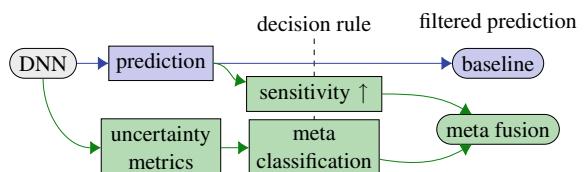
Various methods for uncertainty or confidence estimation exist. While confidence scores are directly produced by a DNN, oftentimes other uncertainty-related quantities, such as softmax margin or entropy, can be generated from the network output. In variational inference (e.g., MC dropout, deep ensembles, or batch norm uncertainty), for a single network input, several predictions are made. Their variance or standard deviation is then interpreted as a measure of prediction uncertainty. Oftentimes, a common strategy for different kinds of uncertainty is missing and comparison between methods is highly context-dependent. Moreover, different methods to esti-

mate uncertainty may produce mutually redundant information which, for lack of comparability, cannot be established experimentally. We propose uncertainty aggregation methods as unifying strategy for different tasks.

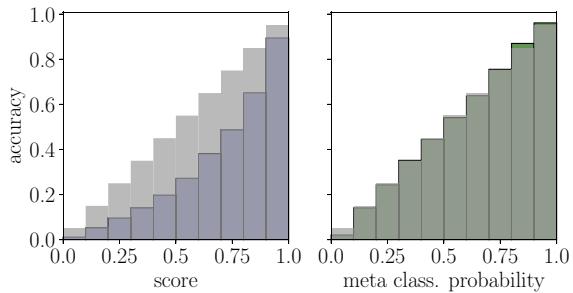
**Meta classification:** In classification tasks, an input is categorized as one class out of a finite number of classes. Evaluation usually takes a binary value of either true or false. Meta classification can then be understood as the task of predicting the label true or false from uncertainty metrics. To this end, a lightweight classification model (e.g., logistic regression, a shallow neural network, or a gradient boosting classifier) is fitted to the training dataset, where uncertainty metrics can be computed and the actual, true label for each sample can be established. This can be regarded as *learning a binary decision rule* based on uncertainty information. On an evaluation dataset, such a meta classification model can be evaluated in terms of classification metrics like the area under the receiver operating characteristic *AUROC* or the area under precision-recall curve *AuPR* [DG06]. Different sources of uncertainty information, thus, serve as co-variables for meta classification models. Their performance can be regarded as a unified evaluation of confidence information. Network-intrinsic confidence scores naturally fit into this framework and can be either directly evaluated as any meta classification model or also serve as a single co-variable for such a model. In particular, different sources can be combined by fitting a meta classification model on the combined set of uncertainty metrics. This allows for the investigation of mutual redundancy between uncertainty information. Meta classification can be applied to any setting where a prediction can be marked as true or false and has served as a baseline for image classification [HG17], but also the tasks of semantic segmentation (“MetaSeg”) [RCH+20] and object detection (“MetaDetect”) [SKR20] can be subject to meta classification. In semantic segmentation, segments can be classified as true or false based on their intersection-over-union (*IoU*) with the ground truth being above a certain threshold. Similarly, bounding boxes are ordinarily declared as true or false predictions based on their maximum *IoU* with any ground truth box (see Sect. 3.2).

**Meta fusion:** Obtaining well-performing meta classifiers allows for their incorporation in the form of decision rules (replacing the standard score threshold) into the original prediction pipeline. Doing so is especially useful in settings where the DNN prediction is based on positive (foreground/detection) and negative (background) predictions, such as semantic segmentation and object detection. Implementing meta classifiers into such a framework is called meta fusion, whereby the improvement of the detection of false predictions over the DNN confidence is supplemented by increased prediction sensitivity of the DNN (see Fig. 2). Greater sensitivity tends to

**Fig. 2** Schematic sketch of the prediction baseline and the alternative meta fusion pipeline



**Fig. 3** Reliability plots (accuracy over confidence) for the confidence score and a gradient boosting meta classifier based on the confidence score (Faster R-CNN on  $\mathcal{D}_{\text{val}}^{\text{COCO}}$ )



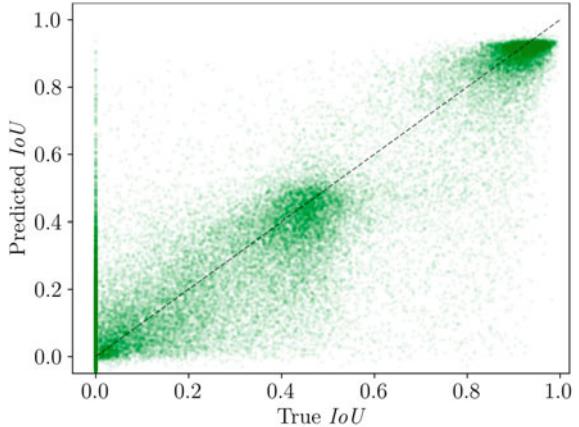
reduce false negative (FN) predictions and increase true (TP) and false positive (FP) predictions. The meta classifier’s task is then to identify the additional FPs as such and filter them from the prediction which leads to a net increase in prediction performance of the combination DNN with the meta classifier over the standalone DNN baseline. Such an increase can be evaluated utilizing metrics usually employed to assess the performance of a DNN in the respective setting, e.g., mean *IoU* in semantic segmentation [CRH+19] or mean average precision (*mAP*) in object detection [RRSG21].

**Calibration:** Confidence estimates with high predictive power (as measured in meta classification) have useful applications, e.g., in meta fusion where detection accuracy can be improved by investing uncertainty information. In addition to this advantage, there is another aspect to the probabilistic estimation of whether the corresponding prediction is correct or not. Confidence calibration tries to gauge the accuracy of the frequentistic interpretation, for example, from 100 examples each with a confidence of about  $\frac{1}{4}$ , statistically we expect about 25 of those examples to be true predictions.

This is often investigated by dividing the confidence range into bins of equal width and computing the accuracy of predictions conditioned to the respective bin (see Fig. 3). For calibrated confidences, the resulting distribution should then be close to the ideally calibrated diagonal. Meta classification is closely related to post-processing calibration methods in that a map is learned on a validation dataset which yields confidences with improved calibration. In particular, meta classification with the confidence score as the only co-variable is (up to the explicit optimization objective) isotonic regression as introduced in [ZE02].

**Meta regression:** In a regression task, a continuous variable (e.g., in  $\mathbb{R}$ ) is estimated given an input. By the same logic as in meta classification, whenever the quality of the prediction of a DNN can be expressed as a real number, we may fit a simple regression model (e.g., linear regression, shallow neural network, or a gradient boosting regression model) on a training dataset where both uncertainty metrics and the actual prediction quality can be computed. This is called a meta regression model which maps uncertainty information to an (e.g.,  $\mathbb{R}$ -valued) *estimation of the prediction quality of the DNN* (see Fig. 4). The relationship between actual and predicted quality can, again, be evaluated on a dedicated dataset in terms of the coefficient

**Fig. 4** Scatter plot of predicted bounding box  $IoU$  over true  $IoU$  of a gradient boosting meta regression model (RetinaNet on  $\mathcal{D}_{\text{eval}}^{\text{KITTI}}$ )



of determination  $R^2$ , a well-established quality metric for regression models. Meta regression shares many features as an evaluation protocol as meta classification, perhaps with the exception that network confidence scores fit less directly into the meta regression logic. However, a meta regression model based on such a score still follows the same design. We note that fitting non-linear regression models (e.g., a neural network or a gradient boosting as opposed to a linear regression) on one variable tends to not significantly improve meta regression performance. The  $IoU$  or modifications thereof in semantic segmentation [RCH+20] or object detection [SKR20] are suitable quality measures to be used in meta regression.

### 3.2 Deep Object Detection Frameworks

We develop output- and gradient-based uncertainty metrics for the task of 2D bounding box (“object”) detection on camera images  $\mathbf{x} \in \mathbb{I}^{H \times W \times C}$ , with  $\mathbb{I} = [0, 1]$ . A DNN tailored to this task usually produces a fixed number  $N_{\text{out}} \in \mathbb{N}$  of output boxes

$$\mathfrak{O}(\mathbf{x}; \boldsymbol{\theta}) = (\mathfrak{O}^1(\mathbf{x}; \boldsymbol{\theta}), \dots, \mathfrak{O}^{N_{\text{out}}}(\mathbf{x}; \boldsymbol{\theta})) \in (\mathbb{R}^4 \times \mathcal{S} \times \mathbb{I}^{N_C})^{N_{\text{out}}}, \quad (1)$$

where  $\boldsymbol{\theta} \in \mathbb{R}^{N_{\text{DNN}}}$  with  $N_{\text{DNN}} \in \mathbb{N}$  denotes the number of weights in the respective DNN architecture and  $N_C \in \mathbb{N}$  is the fixed number of classes (the set of classes is denoted by  $\mathcal{N}_C = \{1, \dots, N_C\}$ ) to be learned. Each output box  $\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta}) = (\hat{\xi}^j, \hat{s}^j, \hat{p}^j) \in \mathbb{R}^4 \times \mathcal{S} \times \mathbb{I}^{N_C}$  for  $j \in \mathcal{N}_{\text{out}} = \{1, \dots, N_{\text{out}}\}$  is encoded by

- Four localization variables, e.g.,  $\hat{\xi} = (\hat{c}_{\min}^j, \hat{r}_{\min}^j, \hat{c}_{\max}^j, \hat{r}_{\max}^j) \in \mathbb{R}^4$  (top-left and bottom-right corner coordinates),
- Confidence score  $\hat{s} \in \mathcal{S} = (0, 1)$  indicating the probability of an object existing at  $\hat{\xi}^j$ , and

- Class probability distribution  $\hat{\mathbf{p}}^j = (\hat{p}_1^j, \dots, \hat{p}_{N_C}^j) \in \mathbb{I}^{N_C}$ .

The predicted class of  $\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta})$  is determined to be  $\hat{\kappa}^j = \arg \max_{k \in \mathcal{N}_C} \hat{p}_k^j$ . The  $N_{\text{out}}$  boxes subsequently undergo filtering mechanisms resulting in  $|\mathcal{N}_{\mathbf{x}}| = \hat{N}_{\mathbf{x}}$  detected instances  $\mathcal{N}_{\mathbf{x}} \subseteq \mathcal{N}_{\text{out}}$

$$\hat{\mathbf{y}} = \left( \mathfrak{B}^1(\mathbf{x}; \boldsymbol{\theta}), \dots, \mathfrak{B}^{\hat{N}_{\mathbf{x}}}(\mathbf{x}; \boldsymbol{\theta}) \right) \in \mathbb{R}^{\hat{N}_{\mathbf{x}} \times (4+1+N_C)}. \quad (2)$$

Commonly, the much smaller number  $\hat{N}_{\mathbf{x}} \ll N_{\text{out}}$  of predicted boxes is the result of two mechanisms. First, *score thresholding*, i.e., discarding any  $\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta})$  with  $s^j < \varepsilon_s$  for some fixed threshold  $\varepsilon_s > 0$ , is performed as an initial distinction between “foreground” and “background” boxes. Afterward, the *non-maximum suppression* (NMS) algorithm is used to reduce boxes with the same class and significant mutual overlap to only one box. This way, for boxes that are likely indicating the same object in  $\mathbf{x}$ , there is only one representative in  $\hat{\mathbf{y}}$ . By “overlap”, we usually mean the intersection-over-union (*IoU*) between two bounding boxes  $\hat{\xi}^j$  and  $\hat{\xi}^k$  which is defined as

$$IoU\left(\hat{\xi}^j, \hat{\xi}^k\right) = \frac{\left|\hat{\xi}^j \cap \hat{\xi}^k\right|}{\left|\hat{\xi}^j \cup \hat{\xi}^k\right|}, \quad (3)$$

i.e., the ratio of their area of intersection and their joint area. The *IoU* between two boxes is always in  $[0, 1]$ , where 0 means no overlap and 1 means the boxes have identical localization. One also rates the localization quality of a prediction in terms of the maximum *IoU* it has with any ground truth box on  $\mathbf{x}$  which has the same class.

The NMS algorithm is based on what we call “candidate boxes”. An output instance  $\mathfrak{O}^k(\mathbf{x}; \boldsymbol{\theta})$  is a *candidate box* for another output instance  $\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta})$  if it

1. has a sufficiently high score ( $\hat{s}^k \geq \varepsilon_s$  above a chosen, fixed threshold  $\varepsilon_s \geq 0$ ),
2. has the same class as  $\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta})$ , i.e.,  $\hat{\kappa}^k = \hat{\kappa}^j$ ,
3. has sufficient *IoU* with  $\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta})$ , i.e.,  $IoU(\hat{\xi}^k, \hat{\xi}^j) \geq \varepsilon_{IoU}$  for some fixed threshold  $\varepsilon_{IoU} \geq 0$  (oftentimes  $\varepsilon_{IoU} = \frac{1}{2}$ ).

We then denote by

$$\text{cand}[\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta})] := \left\{ \mathfrak{O}^k(\mathbf{x}; \boldsymbol{\theta}) : \mathfrak{O}^k(\mathbf{x}; \boldsymbol{\theta}) \text{ is a candidate box for } \mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta}) \right\} \quad (4)$$

the set of candidate boxes for  $\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta})$ . In the NMS algorithm, all output boxes  $\mathfrak{O}(\mathbf{x}; \boldsymbol{\theta})$  are sorted according to their score in descending order. Iteratively, the box  $\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta})$  with the highest score is selected as a prediction and  $\text{cand}[\mathfrak{O}^j(\mathbf{x}; \boldsymbol{\theta})]$  is removed from the ranking of output boxes. This procedure is repeated until there are no boxes left. Note that NMS usually follows score thresholding so this stage may be reached quickly, depending on  $\varepsilon_s$ .

DNN object detection frameworks are usually trained in a supervised manner on images with corresponding labels or ground truth. The latter usually consist of a

number  $N_x$  of ground truth instances  $\bar{\mathbf{y}} = (\bar{\mathbf{y}}^1, \dots, \bar{\mathbf{y}}^{N_x}) \in (\mathbb{R}^4 \times \mathcal{N}_C)^{N_x}$  consisting of similar data as output boxes which we denote by the bar ( $\cdot$ ). In particular, each ground truth instance  $\bar{\mathbf{y}}^j$  has a specified localization  $\bar{\xi}^j = (\bar{c}_{\min}^j, \bar{r}_{\min}^j, \bar{c}_{\max}^j, \bar{r}_{\max}^j) \in \mathbb{R}^4$  and category  $\kappa^j \in \mathcal{N}_C$ . From the network output  $\mathfrak{O}(\mathbf{x}; \boldsymbol{\theta})$  on an image  $\mathbf{x}$  and the ground truth  $\bar{\mathbf{y}}$ , a loss function  $J(\mathfrak{O}(\mathbf{x}; \boldsymbol{\theta}), \bar{\mathbf{y}})$  is computed and iteratively minimized over  $\boldsymbol{\theta}$  by stochastic gradient descent or some of its variants. In most object detection frameworks,  $J$  splits up additively into

$$J = J_\xi + J_s + J_p, \quad (5)$$

with  $J_\xi$  punishing localization mistakes,  $J_s$  punishing incorrect score assignments to boxes (incorrect boxes with high score and correct boxes with low score), and  $J_p$  punishing an incorrect class probability distribution, respectively. In standard gradient descent optimization, the weights  $\boldsymbol{\theta}$  are then updated by the following rule:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} J(\mathfrak{O}(\mathbf{x}; \boldsymbol{\theta}), \bar{\mathbf{y}}), \quad (6)$$

where  $\eta > 0$  is the either fixed or variable learning rate parameter. The learning gradient  $\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \bar{\mathbf{y}}) := \nabla_{\boldsymbol{\theta}} J(\mathfrak{O}(\mathbf{x}; \boldsymbol{\theta}), \bar{\mathbf{y}})$  will play a central role for the gradient-based uncertainty metrics which will be introduced in Sect. 3.4.

### 3.3 Output-Based Uncertainty: MetaDetect

In this section, we construct uncertainty metrics for every  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta}) \in \hat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\theta})$ . We do so in two stages, first by introducing the general metrics that can be obtained from the object detection pipeline. Second, we extend this by additional metrics that can be computed when using MC dropout. We consider a predicted bounding box  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta}) \in \hat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\theta})$  and its corresponding filtered candidate boxes  $\mathfrak{O}^k(\mathbf{x}; \boldsymbol{\theta}) \in \text{cand}[\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})]$  that were discarded by the NMS.

The number of corresponding candidate boxes  $\mathfrak{O}^k(\mathbf{x}; \boldsymbol{\theta}) \in \text{cand}[\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})]$  filtered by the NMS intuitively gives rise to the likelihood of observing a true positive. The more candidate boxes  $\mathfrak{O}^k(\mathbf{x}; \boldsymbol{\theta})$  belong to  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$ , the more likely it seems that  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  is a true positive. We denote by  $N^{(j)}$  the number of candidate boxes  $\mathfrak{O}^k(\mathbf{x}; \boldsymbol{\theta})$  belonging to  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$ , but suppressed by NMS. We increment this number by 1 and also count in  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$ .

For a given image  $\mathbf{x}$ , we have the set of predicted bounding boxes  $\hat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\theta})$  and the ground truth  $\bar{\mathbf{y}}$ . As we want to calculate values that represent the quality of the prediction of the neural network, we first have to define uncertainty metrics for the predicted bounding boxes in  $\hat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\theta})$ . For each  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta}) \in \hat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\theta})$ , we define the following quantities:

- the number of candidate boxes  $N^{(j)} \geq 1$  that belong to  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  (i.e.,  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  belongs to itself; one metric),

- the predicted box  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  itself, i.e., the values of the tuple

$$\left( \hat{c}_{\min}^j, \hat{r}_{\min}^j, \hat{c}_{\max}^j, \hat{r}_{\max}^j, \hat{s}^j, \hat{p}_1^j, \dots, \hat{p}_{N_C}^j \right) \in \mathbb{R}^4 \times \mathcal{S} \times \mathbb{I}^{N_C}, \quad (7)$$

as well as  $\sum_{i \in \mathcal{N}_C} \hat{p}_i^j \in \mathbb{R}$  whenever class probabilities are not normalized ( $6 + N_C$  metrics),

- size  $d = (\hat{r}_{\max}^j - \hat{r}_{\min}^j) \cdot (\hat{c}_{\max}^j - \hat{c}_{\min}^j)$  and circumference  $g = 2 \cdot (\hat{r}_{\max}^j - \hat{r}_{\min}^j) + 2 \cdot (\hat{c}_{\max}^j - \hat{c}_{\min}^j)$  (two metrics),
- $IoU_{pb}^j$ : the  $IoU$  of  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  and the box with the second highest score that was suppressed by  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$ . This value is zero if there are no boxes corresponding to  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  suppressed by the NMS (i.e.,  $N^{(j)} = 1$ ; one metric),
- the minimum, maximum, arithmetic mean, and standard deviation for  $(\hat{r}_{\min}^j, \hat{r}_{\max}^j, \hat{c}_{\min}^j, \hat{c}_{\max}^j, \hat{s}^j)$ , size  $d$  and circumference  $g$  from  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  and all the filtered candidate boxes that were discarded from  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  in the NMS ( $4 \times 7$  metrics),
- the minimum, maximum, arithmetic mean, and standard deviation for the  $IoU$  of  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  and all the candidate boxes corresponding to  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  that were suppressed in the NMS (four metrics),
- relative sizes  $rd = d/g$ ,  $rd_{\max} = d/g_{\min}$ ,  $rd_{\min} = d/g_{\max}$ ,  $rd_{\text{mean}} = d/g_{\text{mean}}$ , and  $rd_{\text{std}} = d/g_{\text{std}}$  (five metrics),
- the maximal  $IoU$  of  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  and all ground truth boxes in  $\bar{\mathbf{y}}$ ; this is not an input to a meta model but serves as the ground truth provided to the respective loss function.

Altogether, this results in  $47 + N_C$  uncertainty metrics which can be aggregated further in a meta classifier or a meta regression model.

We now elaborate on how to calculate uncertainty metrics for every  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta}) \in \hat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\theta})$  when using MC dropout. To this end, we consider the bounding box  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta}) \in \hat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\theta})$  that was predicted without dropout and then we observe under dropout  $K$  times the output of the same anchor box that produced  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})$  and denote them by  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})_1, \dots, \mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})_K$ . For these  $K + 1$  boxes  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta}), \mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})_1, \dots, \mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta})_K$ , we calculate the standard deviation for the localization variables, the objectness score, and class probabilities. This is done for every  $\mathfrak{B}^j(\mathbf{x}; \boldsymbol{\theta}) \in \hat{\mathbf{y}}(\mathbf{x}, \boldsymbol{\theta})$  and results in  $4 + 1 + N_C$  additional dropout uncertainty metrics which we denote by  $\text{std}_{\text{MC}}(\cdot)$  for each of the respective  $4 + 1 + N_C$  box features. Executing this procedure for all available test images, we end up with a structured dataset. Each row represents exactly one predicted bounding box and the columns are given by the registered metrics. After defining a training/test splitting of this dataset, we learn meta classification ( $IoU > 0.5$  vs.  $IoU \leq 0.5$ ) and meta regression (quantitative  $IoU$  prediction) from the training part of this data. All the presented metrics, except for the true  $IoU$ , can be computed without the knowledge of the ground truth. We now want to explore to which extent they are suited for the tasks of meta classification and meta regression.

### 3.4 Gradient-Based Uncertainty for Object Detection

In the setting provided in Sect. 3.2, a DNN learns from a data point  $(\mathbf{x}, \bar{\mathbf{y}}) \in \mathbb{I}^{H \times W \times C} \times (\mathbb{R}^4 \times \mathcal{N}_C)^{N_x}$  by computing the gradient  $\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \bar{\mathbf{y}})$ . The latter is subsequently used to iteratively adjust the current weights  $\boldsymbol{\theta}$ , for example, by the formula given in (6). As such, the quantity  $\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \bar{\mathbf{y}})$  can be interpreted as *learning stress* on  $\boldsymbol{\theta}$  induced by the training data  $(\mathbf{x}, \bar{\mathbf{y}})$ . We explain next how a related quantity gives rise to confidence information.

**Some intuition about gradient uncertainty:** Generally, the loss function  $J$  measures the “closeness” of output boxes to the ground truth  $\bar{\mathbf{y}}$  and  $\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \bar{\mathbf{y}})$  expresses the induced change in  $\boldsymbol{\theta}$  for any deviation from the ground truth. If we assume that  $\hat{\mathbf{y}}(\mathbf{x}; \boldsymbol{\theta})$  is close to  $\bar{\mathbf{y}}$ , only little change in  $\boldsymbol{\theta}$  will be required in an update step. We, therefore, expect  $\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \bar{\mathbf{y}})$  to be of comparably small magnitude. A DNN which is already “well-trained” in the ordinary sense is expected to express high confidence when its prediction is correct for all practical purposes. Conversely, if the network output deviates from the ground truth significantly, learning on  $(\mathbf{x}, \bar{\mathbf{y}})$  induces a larger adjustment in  $\boldsymbol{\theta}$ , leading to a gradient of larger magnitude. In this situation, the confidence in this prediction should be small.

The gradient  $\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \bar{\mathbf{y}})$  is not realizable as a measure of uncertainty or confidence, since it uses the ground truth  $\bar{\mathbf{y}}$ , which is not accessible during inference time. An approach to circumvent this shortcoming was presented in [ORG18], in that the ground truth  $\bar{\mathbf{y}}$  was replaced by the prediction of the DNN  $\hat{\mathbf{y}}$  on  $\mathbf{x}$ . We format the prediction so that it has the same format as the ground truth. Particularly, in the initial application to image classification, the softmax output of the DNN was collapsed by an arg max to the predicted class before insertion into  $J$ . With respect to bounding box localization, there is no adjustment required. Additionally, when computing

$$\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \hat{\mathbf{y}}) = \nabla_{\boldsymbol{\theta}} J(\mathfrak{O}(\mathbf{x}; \boldsymbol{\theta}), \hat{\mathbf{y}}), \quad (8)$$

we disregard the dependency of  $\hat{\mathbf{y}}$  on  $\boldsymbol{\theta}$ , thereby sticking to the design motivation from above. The quantity in (8) represents the weight adjustment induced in the DNN when learning that its prediction on  $\mathbf{x}$  is correct. *Uncertainty or confidence information* is distilled from  $\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \hat{\mathbf{y}})$  by taking it to a scalar. To this end, it is natural to employ norms, e.g.,  $\|\cdot\|_1$  or  $\|\cdot\|_2$ , but we also use other maps such as minimal and maximal entry as well as the arithmetic mean and standard deviation over the gradient’s entries (see (9)). Gradient uncertainty metrics allow for some additional flexibility. Regarding (8), we see that if  $J$  splits into additive terms (such as in the object detection setting, see (5)), gradient metrics allow the extraction of gradient uncertainty from each individual contribution. Additionally, we can restrict the variables for which we compute partial derivatives. One might, for example, be interested in computing the gradient  $\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}_\ell, \hat{\mathbf{y}})$  for the weights  $\boldsymbol{\theta}_\ell$  from only one network layer  $\ell$ . In principle, this also allows for computing uncertainty metrics for individual convolutional filters or even individual weights, thus, also offering a way of localizing uncertainty within the DNN architecture.

$$\begin{aligned} m_1^\theta(J) &= \|\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \hat{\mathbf{y}})\|_1, & m_2^\theta(J) &= \|\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \hat{\mathbf{y}})\|_2, & m_{\max}^\theta(J) &= \max(\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \hat{\mathbf{y}})), \\ m_{\min}^\theta(J) &= \min(\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \hat{\mathbf{y}})), & m_{\text{std}}^\theta(J) &= \text{std}(\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \hat{\mathbf{y}})), & m_{\text{mean}}^\theta(J) &= \text{mean}(\mathbf{g}(\mathbf{x}, \boldsymbol{\theta}, \hat{\mathbf{y}})). \end{aligned} \quad (9)$$

**Application to instance-based settings:** The approach outlined above requires some adaptation in order to produce meaningful results for settings in which the network output  $\mathcal{O}(\mathbf{x}; \boldsymbol{\theta})$  and the ground truth  $\bar{\mathbf{y}}$  consist of several distinct instances, e.g., as in object detection. In such a situation, each component  $\mathcal{O}^j(\mathbf{x}; \boldsymbol{\theta})$  needs to be assigned individual confidence metrics. If we want to estimate the confidence for  $\hat{\mathbf{y}}^j := \mathcal{B}^j(\mathbf{x}; \boldsymbol{\theta})$ , we may regard  $\hat{\mathbf{y}}^j$  as the ground truth replacement entering the second slot of  $J$ . Then, the first argument of  $J$  needs to be adjusted according to  $\hat{\mathbf{y}}^j$ . This is necessary because the loss  $J(\mathcal{O}(\mathbf{x}; \boldsymbol{\theta}), \hat{\mathbf{y}})$  expresses the mistakes made by all the output boxes, i.e., the prediction of instances which actually appear in the ground truth  $\bar{\mathbf{y}}$  but which have nothing to do with  $\hat{\mathbf{y}}^j$  would be punished. The corresponding gradient would, therefore, misleadingly adjust  $\boldsymbol{\theta}$  toward forgetting to see such ground truth instances. Thus, it is a natural idea to identify  $\text{cand}[\hat{\mathbf{y}}^j]$  in the network output and only enter those boxes into  $J$  which likely indicate one and the same instance on  $\mathbf{x}$ . We define the corresponding gradient

$$\mathbf{g}^{\text{cand}}(\mathbf{x}, \boldsymbol{\theta}, \hat{\mathbf{y}}^j) := \nabla_{\boldsymbol{\theta}} J(\text{cand}[\hat{\mathbf{y}}^j](\mathbf{x}, \boldsymbol{\theta}), \hat{\mathbf{y}}^j), \quad (10)$$

which serves as the basis of gradient uncertainty in instance-based settings. The flexibility mentioned in the previous paragraph carries over to this definition and will be exploited in our experiments.

## 4 Experimental Setup

In this section, we explain our choice of datasets, models, and metrics as well as experimental setup and implementation details.

### 4.1 Databases, Models, and Metrics

We investigate three different object detection datasets in the Pascal VOC [EVGW+15], the MS COCO [LMB+14], and the KITTI vision benchmark [GLSU15]. Pascal VOC  $\mathcal{D}^{\text{VOC}12}$  and MS COCO  $\mathcal{D}^{\text{COCO}17}$  are vision datasets containing images of a wide range of mostly everyday scenarios with 20 and 80 annotated classes, respectively. They both possess training and dedicated evaluation splits for testing and constitute established benchmarks in object detection. In order to test our methods in driving scenarios, we also investigate the KITTI dataset  $\mathcal{D}^{\text{KITTI}}$  which shows different German urban environments and comes with annotated training

**Table 1** Dataset split sizes for training and testing

Dataset	$\mathcal{D}_{\text{train}}^{\text{VOC12}}$	$\mathcal{D}_{\text{test}}^{\text{VOC12}}$	$\mathcal{D}_{\text{train}}^{\text{COCO17}}$	$\mathcal{D}_{\text{val}}^{\text{COCO17}}$	$\mathcal{D}_{\text{train}}^{\text{KITTI}}$	$\mathcal{D}_{\text{eval}}^{\text{KITTI}}$
Size	14,805	4,952	118,287	5,002	5,481	2,000

images. From the annotated training set, we take 2,000 randomly chosen images for evaluation. An overview of the sizes of the respective training and evaluation datasets can be found in Table 1.

The experiments we present here have been carried out based on a YOLOv3 [RF18] re-implementation in PyTorch. We have trained our model from scratch on each of the three training splits under dropout with a probability 0.5 between the last and the second-to-last convolution layer in each of the regression heads. As meta classification and meta regression models, we employ the gradient boosting models in [CG16] with standard settings.

For the evaluation of our meta classification models, we use cross-validated area-under-curve metrics (*AuROC* and *AuPR* [DG06]) instead of accuracy, since the former measure the classification quality independently of any threshold, whereas accuracy only reflects the decisions made for the classification probability 0.5. For cross-validation, we use 10 individual image-wise splits of uncertainty metrics for the training of meta classifiers and the respective complementary split for evaluation. This particular procedure is employed whenever cross-validation is used in the following experiments. Especially in the VOC and KITTI dataset, *AuROC* values are mostly over 0.9, so we regard *AuPR* as a second classification metric. Object detection quality in meta fusion experiments will on the one hand be evaluated in terms of cross-validated mean average precision (*mAP*) [EVGW+15]. However, as the authors of [RF18] recognized *mAP* as computed in the VOC challenge is insensitive to FPs, which is why, on the other hand, we compute the cross-validated mean  $F_1$  score ( $mF_1$ ) as well. For the precision-recall curves of each class, we compute individual  $F_1$  scores which are sensitive to FPs and we average them over all classes as a complementary metric to *mAP*. Confidence calibration is usually evaluated in terms of expected or maximum calibration error (*MCE*) [NCH15], computed over bins of width 0.1. As the expected calibration error is sensitive to the box count per bin, we instead regard in addition to the *MCE* the average calibration error (*ACE*) [NZV18]. Meta regression quality is evaluated by the usual coefficient of determination  $R^2$ .

## 4.2 Implementation Details

Our object detection models receive a dropout layer between the second-to-last and the last layers in each of the three YOLOv3 detection heads. Dropout is active during training with a rate of 0.5 and also during MC inference, where we take standard deviations over 30 dropout samples for each of the  $4 + 1 + N_C$  instance features of

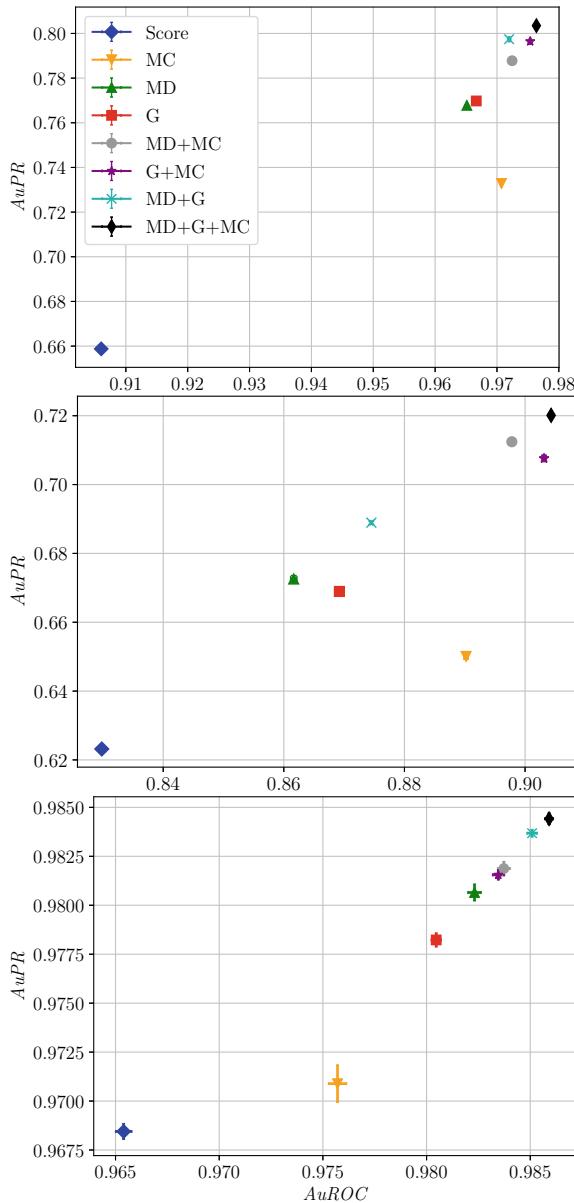
all output boxes. The MetaDetect metrics introduced in Sect. 3.3 are computed from a score threshold of  $\varepsilon_s = 0.0$ , as it has been found in [SKR20] that lower thresholds lead to better performance in meta classification and meta regression.

Gradient uncertainty metrics were computed for the weights in the second-to-last (“ $T - 1$ ”) and the last (“ $T$ ”) convolutional layers in the YOLOv3 architecture at the same candidate score threshold as for the MetaDetect metrics. As there are three detection heads, corresponding to a  $76 \times 76$  (“S”), a  $38 \times 38$  (“M”), and a  $19 \times 19$  (“L”) cell grid, we also compute gradient uncertainty metrics individually for each detection head. We use this distinction in our notation and, for example, indicate the set of parameters from the last layer ( $T$ ) of the detection head producing the  $76 \times 76$  cell grid (S) by  $\theta(T, S)$ . Moreover, as indicated in Sect. 3.4, we also exploit the split of the loss function in (5). Each of the computed  $2 \times 3 \times 3 = 18$  gradients per box results in the 6 uncertainty metrics presented in (9) giving a total of 108 gradient uncertainty metrics per bounding box. Due to the resulting computational expense, we only compute gradient metrics for output boxes with score values  $\hat{s} \geq 10^{-4}$  and regard only those boxes for all of the following experiments.

### 4.3 Experimental Setup and Results

Starting from pre-trained models, we compute the DNN output, as well as MC dropout, MetaDetect, and gradient uncertainty metrics on each evaluation dataset (see Table 1). For each output box, we also compute the maximum  $IoU$  with the respective ground truth such that the underlying relation can be leveraged. Before fitting any model, we perform NMS on the output boxes which constitutes the relevant case for our meta fusion investigation. In the latter, we are primarily interested in finding true boxes which are usually discarded due to their low assigned score. We compare the performance of different sets of uncertainty metrics in different disciplines. In the following, “Score” denotes the network’s confidence score, “MC” stands for MC dropout, and “MD” the MetaDetect uncertainty metrics as discussed in Sect. 3.3. Moreover, “G” denotes the full set of gradient uncertainty metrics from Sect. 3.4 and the labels “MD+MC”, “G+MC”, “MD+G”, and “MD+G+MC” stand for the respective unions of MC dropout, MetaDetect, and gradient uncertainty metrics.

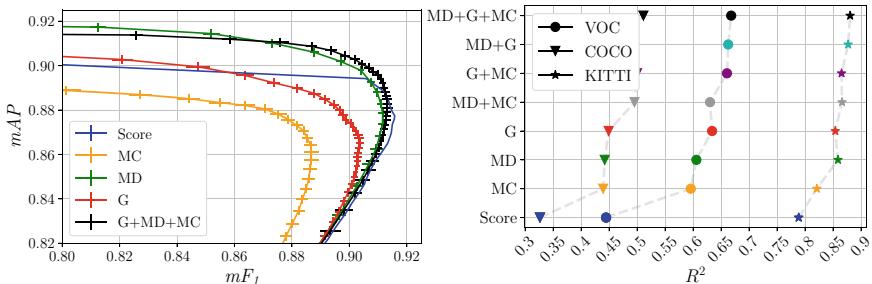
**Meta classification:** We create 10 splits of the dataset of uncertainty metrics and computed  $IoU$  by randomly dividing it 10 times in half. We assign the TP label to those examples which have  $IoU > 0.5$  and the FP label otherwise. On each split, we fit a gradient boosting classifier to the respective set of uncertainty metrics (see Fig. 5) on one half of the data. The resulting model is used to predict a classification probability on the uncertainty metrics of the second half of the data and vice-versa. This results in meta classification predictions on all data points. We evaluate these in terms of the previously introduced area-under-curve metrics under usage of the TP/FP labels generated from the ground truth, where we obtain averages and standard deviations from the 10 splits shown in Fig. 5. Performance measured in either of the two area-



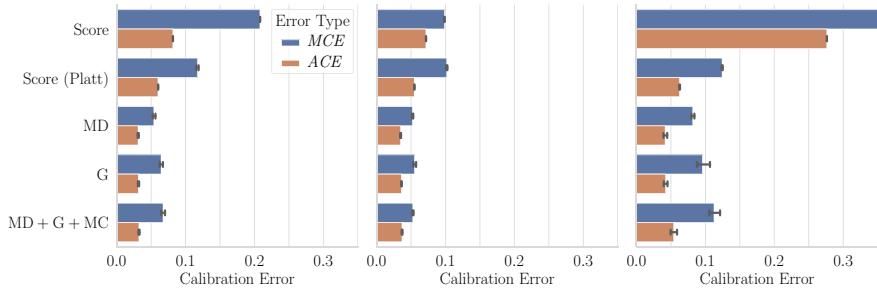
**Fig. 5** Score and meta classification  $AuPR$  over  $AuROC$  from 10-fold cross-validation with respective standard deviation error bars for  $\mathcal{D}_{\text{test}}^{\text{VOC12}}$ ,  $\mathcal{D}_{\text{val}}^{\text{COCO17}}$ , and  $\mathcal{D}_{\text{eval}}^{\text{KITTI}}$  (from top to bottom)

under-curve metrics suggests that the models situated in the top-right separate TP from FP best. In all datasets, the meta classifiers MD, G, and combinations thereof outperform the network confidence by a large margin (upwards of 6 *AuROC* percent points (pps) on  $\mathcal{D}_{\text{test}}^{\text{VOC12}}$ , 2.5 on  $\mathcal{D}_{\text{val}}^{\text{COCO17}}$ , and 1.5 on  $\mathcal{D}_{\text{eval}}^{\text{KITTI}}$ ), with MC dropout being among the weakest meta classifiers. Note that for both  $\mathcal{D}_{\text{test}}^{\text{VOC12}}$  (top) and  $\mathcal{D}_{\text{val}}^{\text{COCO17}}$  (center), the standard deviation bars are occluded by the markers due to the size of the range. On  $\mathcal{D}_{\text{eval}}^{\text{KITTI}}$ , we can see some of the error bars and the models forming a strict hierarchy across the two metrics with overlapping MD+MC and G+MC. Overall, the three sources of uncertainty MC, MD, and G show a significant degree of non-redundancy across the three datasets with mutual boosts of up to 7 *AuPR* ppts on  $\mathcal{D}_{\text{test}}^{\text{VOC12}}$ , 4.5 on  $\mathcal{D}_{\text{val}}^{\text{COCO17}}$ , and 1.1 on  $\mathcal{D}_{\text{eval}}^{\text{KITTI}}$ .

**Meta fusion:** We use the 10 datasets of uncertainty metrics and resulting confidences as in meta classification in combination with the bounding box information as new post-NMS output of the network. Thresholding on the respective confidence on decision thresholds  $\varepsilon_{\text{dec}} \in (0, 1)$  with a fixed step width of 0.01 reproduces the baseline object detection pipeline in the case of the score. The network prediction with assigned confidence can then be evaluated in terms of the aforementioned class-averaged performance measures for which we obtain averages and standard deviations from the 10 splits, except for the deterministic score baseline. In our comparison, we focus on the score baseline, MC dropout baseline, output- and gradient-based uncertainty metrics (MD and G), and the full model based on all available uncertainty metrics (G+MD+MC). We show the resulting curves of *mAP* over  $mF_1$  in Fig. 6 on the right, where we see that the maximum  $mF_1$  value is attained by the confidence score. In addition, we observe a sharp bend in the score curve resulting from a large count of samples with a score between 0 and 0.01, where the interpolation to the threshold  $\varepsilon_s = 0$  reaches to 0.912 *mAP* at a low 0.61  $mF_1$  (outside the plotting range). In terms of *mAP*, however, the score is surpassed by the meta fusion models G and in particular the meta classification models involving MD (maximum *mAP* close to 0.92 at around 0.81  $mF_1$ ). The raw MetaDetect confidences seem



**Fig. 6** *Left:* *mAP* over  $mF_1$  score for five meta fusion models computed from 10-fold cross-validation. We show standard deviation error bars obtained from cross-validation. *Right:* meta regression  $R^2$  for uncertainty models on  $\mathcal{D}_{\text{test}}^{\text{VOC12}}$ ,  $\mathcal{D}_{\text{val}}^{\text{COCO17}}$ , and  $\mathcal{D}_{\text{eval}}^{\text{KITTI}}$ . The plot shows the averages over 10-fold cross-validation with standard deviation error bars hidden by the symbols



**Fig. 7** Average  $ACE$  and mean calibration error  $MCE$  of the score and meta classification confidences for  $D_{\text{test}}^{\text{VOC12}}$ ,  $D_{\text{val}}^{\text{COCO17}}$ , and  $D_{\text{eval}}^{\text{KITTI}}$  (from left to right)

to slightly improve over G+MD+MC in terms of  $mAP$  by about 0.3 ppts., which may be due to overfitting of the gradient boosting model, but could also result from the altered confidence ranking which is initially performed when computing mean average precision. Meta fusion based on MC dropout uncertainty shows the worst performance of all four methods considered in this test.

**Calibration:** From the generated meta classification probabilities and the score, calibration errors can be computed indicating their calibration as binary classifiers and compared; see Fig. 7 with standard deviations indicated by error bars. In addition, we show the Platt-scaled confidence score as a reference for a post-processing calibration method. Naturally,  $ACE$  is smaller than  $MCE$ . The meta classification models show consistently far-smaller calibration errors (maximum  $MCE$  of 0.11 on  $D_{\text{eval}}^{\text{KITTI}}$ ) than the confidence score (minimum  $MCE$  of 0.1 on  $D_{\text{val}}^{\text{COCO17}}$ ) with comparatively small miscalibration on the COCO validation dataset (center). We also observe that meta classifiers are also better calibrated than the Platt-scaled confidence score with the smallest margin on  $D_{\text{eval}}^{\text{KITTI}}$ . Note from the  $MCE$  that meta classification models show generally good calibration with a maximum  $ACE$  around 0.05 as indicated in Sect. 3 and by the example of Fig. 3.

**Meta regression:** By an analogous approach to meta classification, we can fit gradient boosting regression models in a cross-validated fashion to sets of uncertainty metrics and the true maximum  $IoU$  with the ground truth. The resulting predicted  $IoU$  values can be evaluated in terms of  $R^2$  which is illustrated in Fig. 6 on the left. Note once again that error bars for the standard deviation are covered by the markers. We observe an overall similar trend of the metrics to the meta classification performance in that combined models tend to perform better (boosts of up to 6  $R^2$  ppts.) indicating mutual non-redundancy among the different sources of uncertainty. The models based on MD and G individually show significant improvements over the confidence score (over 5  $R^2$  ppts. on all datasets) and a slight improvement over MC (up to 3  $R^2$  ppts.) for all investigated datasets.

**Parameter importance:** We investigate which parameters are most important for meta classification in order to find the best performance with as few metrics as

**Table 2** Cumulative parameter ranking in terms of their importance for  $AuROC$  and  $AuPR$  obtained from a greedy search for the first 9 chosen parameters. Metrics indicated are averages obtained from cross-validation. We show the performance of the models with all metrics as co-variables for reference on each of the datasets  $\mathcal{D}_{\text{VOC}12}$ ,  $\mathcal{D}_{\text{COCO}17}$ , and  $\mathcal{D}_{\text{KITTI}}^{\text{eval}}$

VOC	COCO			KITTI		
	$AuROC$	$AuPR$	$AuROC$	$AuPR$	$AuROC$	$AuPR$
0.917	$\text{stdMC}(\hat{s})$	0.658	$\hat{s}$	0.830	$\hat{s}$	0.623
0.959	$m_{\theta(T,S)}(J_s)$	0.740	$m_{\max}^{\theta(T,S)}(J_p)$	0.875	$\text{stdMC}(\hat{s})$	0.664
0.966	$m_{\theta(T,S)}^{\max}(J_p)$	0.769	$\text{stdMC}(\hat{s})$	0.886	$m_{\max}^{\theta(T,L)}(J_p)$	0.685
0.971	$m_{\theta(T,M)}(J_p)$	0.784	$m_{\text{mean}}^{\theta(T,M)}(J_p)$	0.893	$m_{\max}^{\theta(T,S)}(J_p)$	0.704
0.972	$\hat{c}_{\min}$	0.790	$m_{\theta(T-1,S)}^{\max}(J_\xi)$	0.899	$m_{\max}^{\theta(T-1,M)}(J_p)$	0.710
0.974	$\hat{s}$	0.795	$\hat{c}_{\min}$	0.900	$m_{\theta(T,M)}^{\max}(J_p)$	0.713
0.975	$m_{\theta(T,S)}(J_\xi)$	0.799	$\text{stdMC}(\hat{c}_{\min})$	0.901	$m_{\max}^{\theta(T-1,L)}(J_s)$	0.715
0.975	$\hat{r}_{\min}$	0.802	$m_{\text{mean}}^{\theta(T,S)}(J_\xi)$	0.902	$\text{stdMC}(\hat{c}_{\min})$	0.716
0.976	$\hat{c}_{\max}$	0.803	$m_{\theta(T-1,M)}^{\max}(J_p)$	0.902	$m_{\text{mean}}^{\theta(T,S)}(J_s)$	0.717
0.976	MD+G+MC	0.803	MD+G+MC	0.904	MD+G+MC	0.720
					MD+G+MC	0.986
					MD+G+MC	0.984

possible. We pursue a greedy approach and evaluate single-metric models at first in terms of *AuROC* and *AuPR* in the cross-validated manner used before. We then fix the best-performing single metric and investigate two-metric combinations and gather the resulting metrics ranking with their area-under-curve score in Table 2. In addition to the nested models, we also show the model using all metrics which is also depicted in Fig. 5. Note that the numbers represent cross-validated averages, where we choose not to show standard deviations as we are mainly interested in seeing which metrics combine to yield well-separating meta classifiers. On all datasets, we find that a total of 9 metrics suffices to come close to or reach the performance of the best model in question. The network confidence appears as the most informative single metric 5 out of 6 times. Another metric that leads to early good improvements is the related MC dropout standard deviation of the confidence score. Other noteworthy metrics are the left border coordinate  $\hat{c}_{\min}$  and gradient metrics for the class probability contribution  $J_p$ . Overall, gradient metrics contribute a large amount of information to  $\hat{s}$  and  $\text{std}_{\text{MC}}(\hat{s})$  which is indicative to meta classification performance.

## Conclusions and Outlook

In this chapter, we have outlined two largely orthogonal approaches to quantifying epistemic uncertainty in deep object detection networks, the output-based MetaDetect approach, and gradient-based uncertainty metrics. Both compare favorably to the network confidence score and MC dropout in terms of meta classification and meta regression with the additional benefit that meta classifiers give rise to well-calibrated confidences irrespective of the set of metrics they are based on. Moreover, different sources of uncertainty information lead to additional boosts when combined, indicating mutual non-redundancy. We have seen that the meta fusion approach enables us to trade uncertainty information for network performance in terms of mean average precision with the best meta fusion models involving the MetaDetect metrics. In terms of raw meta classification capabilities, well-performing models can be achieved by only regarding a small fraction of all metrics, with gradient-based uncertainty metrics contributing highly informative components. The design of well-performing meta classifiers and meta regression models opens up possibilities for further applications. On the one hand, the implementation of meta classification models into an active learning cycle could potentially lead to a drastic decrease in data annotation costs. On the other hand, meta classification models may be applicable beneficially in determining the labeling quality of datasets and detecting labeling mistakes.

**Acknowledgements** The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project “Methoden und Maßnahmen zur Absicherung von KI-basierten Wahrnehmungsfunktionen für das automatisierte Fahren (KI-Absicherung)”. The authors would like to thank the consortium for the successful cooperation. Furthermore, we gratefully acknowledge financial support from the state Ministry of Economy, Innovation and Energy of Northrhine Westphalia (MWIDE) and the European Fund for Regional Development via the FIS.NRW project BIT-KI, grant no. EFRE-0400216. The authors gratefully acknowledge the Gauss

Centre for Supercomputing e.V. ([www.gauss-centre.eu](http://www.gauss-centre.eu)) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

## References

- [BCR+20] D. Brüggemann, R. Chan, M. Rottmann, H. Gottschalk, S. Bracke, Detecting out-of-distribution objects in semantic segmentation of street scenes, in *Proceedings of the European Safety and Reliability Conference (ESREL)*, virtual conference (November 2020), pp. 3023–3030
- [CCKL19] J. Choi, D. Chun, H. Kim, H.-J. Lee, Gaussian YOLOv3: an accurate and fast object detector using localization uncertainty for autonomous driving, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA, USA, June 2019), pp. 502–511
- [CCLK20] J. Choi, D. Chun, H.-J. Lee, H. Kim, Uncertainty-based object detector for autonomous driving embedded platforms, in *Proceedings of the IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, virtual conference (August 2020), pp. 16–20
- [CG16] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (August 2016), pp. 785–794
- [CRG21] R. Chan, M. Rottmann, H. Gottschalk, Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, virtual conference (October 2021), pp. 5128–5137
- [CRGR21] P. Colling, L. Roese-Koerner, H. Gottschalk, M. Rottmann, MetaBox+: a new region based active learning method for semantic segmentation using priority maps, in *Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, virtual conference (February 2021), pp. 51–62
- [CRH+19] R. Chan, M. Rottmann, F. Hüger, P. Schlicht, H. Gottschalk, Controlled false negative reduction of minority classes in semantic segmentation, in *2020 International Joint Conference on Neural Networks (IJCNN)* (IEEE, July 2020), pp. 1–8
- [DG06] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in *Proceedings of the International Conference on Machine Learning (ICML)* (Pittsburgh, PA, USA, June 2006), pp. 233–240
- [DL90] J.S. Denker, Y. LeCun, Transforming neural-net output levels to probability distributions, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Denver, CO, USA, November 1990), pp. 853–859
- [EVGW+15] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The Pascal visual object classes challenge: a retrospective. *Int. J. Comput. Vis. (IJCV)* **111**(1), 98–136 (2015)
- [Gal17] Y. Gal, *Uncertainty in Deep Learning*. Dissertation (University of Cambridge, 2017)
- [GG16] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: representing model uncertainty in deep learning, in *Proceedings of the International Conference on Machine Learning (ICML)* (New York, NY, USA, June 2016), pp. 1050–1059
- [GLSU15] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, The KITTI Vision Benchmark Suite (2015). [Online; accessed 2021-11-18]
- [GPSW17] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in *Proceedings of the International Conference on Machine Learning (ICML)* (Sydney, NSW, Australia, August 2017), pp. 1321–1330

- [GSS15] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in *Proceedings of the International Conference on Learning Representations (ICLR)* (San Diego, CA, USA, 2015), pp. 1–11
- [HG17] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Toulon, France, April 2017), pp. 1–12
- [HSW20] A. Harakeh, M. Smart, S.L. Waslander, BayesOD: a Bayesian approach for uncertainty estimation in deep object detectors, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, virtual conference (May 2020), pp. 87–93
- [HVD14] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS) Workshops* (Montréal, QC, Canada, December 2014), pp. 1–9
- [HW21] Eyke Hüllermeier, Willem Waegeman, Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Mach. Learn.* **110**, 457–506 (2021)
- [HZW+19] Y. He, C. Zhu, J. Wang, M. Savvides, X. Zhang, Bounding box regression with uncertainty for accurate object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA, USA, May 2019), pp. 2888–2897
- [JLM+18] B. Jiang, R. Luo, J. Mao, T. Xiao, Y. Jiang, Acquisition of localization confidence for accurate object detection, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich, Germany, September 2018), pp. 784–799
- [KD19] F. Kraus, K. Dietmayer, Uncertainty estimation in one-stage object detection, in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)* (Auckland, New Zealand, October 2019), pp. 53–60
- [KG17] A. Kendall, Y. Gal, What uncertainties do we need in Bayesian deep learning for computer vision? in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Long Beach, CA, USA, December 2017), pp. 5574–5584
- [KKSH20] F. Küppers, J. Kronenberger, A. Shantia, A. Haselhoff, Multivariate confidence calibration for object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, virtual conference (June 2020), pp. 1322–1330
- [KP20] A. Kumar, B. Poole, On implicit regularization in  $\beta$ -VAEs, in *Proceedings of the International Conference on Machine Learning (ICML)* (July 2020), pp. 5480–5490
- [KRBG21] K. Kowol, M. Rottmann, S. Bracke, H. Gottschalk, YOdar: uncertainty-based sensor fusion for vehicle detection with camera and radar sensors, in *Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART)*, virtual conference (February 2021), pp. 177–186
- [LAE+16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Amsterdam, The Netherlands, October 2016), pp. 21–37
- [LDBK18] M. Truong Le, F. Diehl, T. Brunner, A. Knol., Uncertainty estimation for deep neural object detectors in safety-critical applications, in *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)* (Maui, HI, USA, November 2018), pp. 3873–3878
- [LGG+17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (Venice, Italy, October 2017), pp. 2980–2988,
- [LGRB20] Z. Lyu, N. Gutierrez, A. Rajguru, W.J. Beksi, Probabilistic object detection via deep ensembles, in *Proceedings of the European Conference on Computer Vision (ECCV)*, virtual conference (August 2020), pp. 67–75

- [LHK+21] Y. Lee, J.-w. Hwang, H.-I. Kim, K. Yun, J. Park, S. Ju Hwang, Localization Uncertainty Estimation for Anchor-Free Object Detectors (September 2021), pp. 1–10. [arXiv:2006.15607](https://arxiv.org/abs/2006.15607)
- [LMB+14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. Lawrence Zitnick, Microsoft COCO: common objects in context, in *Proceedings of the European Conference on Computer Vision (ECCV)* (Zurich, Switzerland, September 2014), pp. 740–755
- [LPB17] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Long Beach, CA, USA, December 2017), pp. 6402–6413
- [Mac92] D.J.C. MacKay, A practical Bayesian framework for backpropagation networks. *Neural Comput.* **4**(3), 448–472 (1992)
- [MNDS18] D. Miller, L. Nicholson, F. Dayoub, N. Sünderhauf, Dropout sampling for robust object detection in open-set conditions, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD, Australia, May 2018), pp. 3243–3249
- [MRG20] K. Maag, M. Rottmann, H. Gottschalk, Time-dynamic estimates of the reliability of deep semantic segmentation networks, in *Proceedings of the IEEE International Conference on Tools With Artificial Intelligence (ICTAI)* virtual conference (November 2020), pp. 502–509
- [MRV+21] K. Maag, M. Rottmann, S. Varghese, F. Hüger, P. Schlicht, H. Gottschalk, Improving video instance segmentation by light-weight temporal uncertainty estimates, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* (July 2021), pp. 1–8
- [NCH15] M. Pakdaman Naeini, G. Cooper, M. Hauskrecht, Obtaining well calibrated probabilities using Bayesian binning, in *Proceedings of the AAAI Conference on Artificial Intelligence* (Austin, TX, USA, January 2015), pp. 2901–2907
- [NMC05] A. Niculescu-Mizil, R. Caruana, Predicting good probabilities with supervised learning, in *Proceedings of the International Conference on Machine Learning (ICML)* (Bonn, Germany, August 2005), pp. 625–632
- [NZV18] L. Neumann, A. Zisserman, A. Vedaldi, Relaxed softmax: efficient confidence auto-calibration for safe pedestrian detection, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS) Workshops* (Montréal, QC, Canada, December 2018), pp. 1–8
- [ORF20] P. Oberdiek, M. Rottmann, G.A. Fink, Detection and retrieval of out-of-distribution objects in semantic segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, virtual conference (June 2020), pp. 1331–1340
- [ORG18] P. Oberdiek, M. Rottmann, H. Gottschalk, Classification uncertainty of deep neural networks based on gradient information, in *Proceedings of the IAPR TC3 Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR)* (Siena, Italy, September 2018), pp. 113–125
- [Pla99] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in *Advances in Large Margin Classifiers*. ed. by A.J. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (MIT Press, 1999), pp. 61–74
- [RCH+20] M. Rottmann, P. Colling, T.-P. Hack, R. Chan, F. Hüger, P. Schlicht, H. Gottschalk, Prediction error meta classification in semantic segmentation: detection via aggregated dispersion measures of softmax probabilities, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, virtual conference (July 2020), pp. 1–9
- [RF18] J. Redmon, A. Farhadi, Yolov3: An Incremental Improvement (April 2018), pp. 1–6. [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)

- [RHGS15] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in *Proceedings of the Conference on Neural Information Processing Systems (NIPS/NeurIPS)* (Montréal, QC, Canada, December 2015), pp. 91–99
- [RMC+20] M. Rottmann, K. Maag, R. Chan, F. Hüger, P. Schlicht, H. Gottschalk, Detection of false positive and false negative samples in semantic segmentation, in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)* (Grenoble, France, March 2020), pp. 1351–1356
- [RRSG21] T. Riedlinger, M. Rottmann, M. Schubert, H. Gottschalk, Gradient-Based Quantification of Epistemic Uncertainty for Deep Object Detectors (July 2021), pp. 1–20. [arXiv:2107.04517](https://arxiv.org/abs/2107.04517)
- [RS19] M. Rottmann, M. Schubert, Uncertainty measures and prediction quality rating for the semantic segmentation of nested multi resolution street scene images, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (Long Beach, CA, USA, June 2019), pp. 1361–1369
- [SFKM20] N. Stähl, G. Falkman, A. Karlsson, G. Mathiason, Evaluation of uncertainty quantification in deep learning, in *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, virtual conference (June 2020), pp. 556–568
- [SHK+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(56), 1929–1958 (2014)
- [SKR20] M. Schubert, K. Kahl, M. Rottmann, Metadetect: uncertainty quantification and prediction quality estimates for object detection, in *2021 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2021), pp. 1–10
- [SZS+14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in *Proceedings of the International Conference on Learning Representations (ICLR)* (Banff, AB, Canada, December 2014), pp. 1–10
- [VSG19] V. Thanvantri Vasudevan, A. Sethy, A. Roshan Ghias, Towards better confidence estimation for neural models, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Brighton, UK, May 2019), pp. 7335–7339
- [ZE01] B. Zadrozny, C. Elkan, Obtaining calibrated probability estimates from decision trees and Naive Bayesian classifiers, in *Proceedings of the International Conference on Machine Learning (ICML)* (Williamstown, MA, USA, June 2001), pp. 609–616
- [ZE02] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (Edmonton, AB, Canada, July 2002), pp. 694–699

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

