An Integrated Approach for Traffic Scene Understanding from Monocular Cameras

Towards Resource-constrained Perception of Environment Representations with Multi-task Convolutional Neural Networks

DISSERTATION

submitted in partial fulfillment
of the requirements for the degree

Doktor-Ingenieur
(Doctor of Engineering)

in the

Faculty of Electrical Engineering and Information Technology
at TU Dortmund University

by

Malte Oeljeklaus, M.Sc.

Essen, Germany

Date of submission: November 11, 2020

First examiner: Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram
Second examiner: Univ.-Prof. Dr.-Ing. Klaus Dietmayer

Date of approval: May 7, 2021

To my family.

Acknowledgement

This thesis was written during my work as a research assistant at the Institute of Control Theory and Systems Engineering of the Faculty of Electrical Engineering and Information Technology of the TU Dortmund University. My special thanks go to Professor Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram for the excellent and ongoing support of my doctoral studies since the early stages until completion and for entrusting me with the scientific freedom that allowed me to write this thesis. Not only through his professional guidance, the constructive discussions and the funding of my doctoral project, but above all through the exceptionally open and trusting atmosphere he created within the team, he undoubtedly played a substantial role in the successful completion of my doctorate. I would also like to thank Professor Dr.-Ing. Klaus Dietmayer for the interest he took in my work, for reviewing this thesis as a second examiner and for the thorough and fair final exam. I would also like to thank Professor Dr.-Ing. Stefan Tappertzhofen for his kind involvement as a third examiner and Professor Dr.-Ing. Martin Pfost for chairing the examination committee.

I thank all employees and former employees of the institute for their friendly and collegial attitude and for their good advice on all scientific and technical problems. In particular, I am thankful to apl. Professor Dr. rer. nat. Frank Hoffmann for his early encouragement and support, which paved the way for me to take up a doctoral project in the first place. I would also like to thank Dr.-Ing. Daniel Schauten for sharing his years of teaching experience and also his experience in working with the university administration.

Furthermore, I want to thank my office colleagues Dr.-Ing. Martin Keller, Dr.-Ing. Javier Antonio Oliva Alonso, Moritz Lütkemöller, Katharina Bartsch, Dr.-Ing. Benedikt Meier and Alexander Hugenroth for enriching my daily work through their constructive and often humorous interaction. Special thanks also go to Dr.-Ing. Christian Wissing, Christian Lienke and Andreas Homann, for the valuable discussions, for sharing their insights on automotive industry practice and also for the great time on our joint conference trip, which I remember very fondly.

I would also like to express my gratitude to Artemi Makarow for his collegial support and especially for his help with the first print version. Likewise, I thank Dr.-Ing. Christoph Rösmann for sharing his extensive experience in software development with me and for establishing modern DevOps tools at the institute, which were extremely helpful for my project. My additional thanks go to Khazar Dargahi Nobari, Faiza Tabassum, and Martin Krüger, with particular thanks for the most memorable scientific and personal conversations during our shared work commutes.

My special thanks as well are due to Jan Braun for his encouragement and mental support during the difficult writing phase. Moreover, I am thankful to my colleagues Katharina Bartsch, Manuel Schmidt, and Jan Braun for their proofreading support and valuable input on the presentation slides. Richard Scherping deserves

my thanks for having initiated my first contact with the institute early on, and also for his friendship since then. In addition, I would also like to thank Franz Albers, Christopher Diehl, Philip Dorpmüller, Robert Gonschorek, Pascal Janke, Maximilian Krämer, Dr.-Ing. Christoph Krimpmann, Dr.-Ing. Jörn Malzahn, Freia Irina Muster, Dr.-Ing. Krishna Kumar Narayanan, Dr.-Ing. Luis Philipe Posada, Niklas Stannartz and Mirko Waldner for the good suggestions and discussions in the doctoral seminars, coffee rounds and lunch breaks.
Naturally, I would also like to mention the technical support for the experimental setups and the IT infrastructure, for which I thank Jürgen Limhoff, Rainer Müller-Burtscheid, Sascha Kersting and Halit Cicek sincerely. The same goes for Gabriele Rebbe and Nicole Czerwinski, whom I thank for their friendly support in all administrative matters.
Some ideas of the present work also arose in the context of student work. For this, I thank the participating students, namely Patrick Weyers, Carlos Miguel Treviño Campa, Marvin Rühl, Björn Polenz, Dizhao Jiang, Zichao Hu and Bernd Möllenbeck.
Gratitude is also owed to all my friends whom I have not mentioned by name for always helping me to find distraction and relaxation when my thoughts were at my work more than due.
Finally, I would like to thank my entire family for their unwavering, unconditional support and encouragement of my personal and professional growth throughout my life. Most importantly, I would like to thank my dear wife Ezgi. Your loving support and patience over the last years were crucial for the completion of this thesis and confirm once more, that as a team we can accomplish just about anything.

Thank you!

Hildesheim, June 2021 Malte Oeljeklaus

Abstract

This thesis investigates methods for traffic scene perception with monocular cameras as a foundation for a basic environment model in the context of automated vehicles. The developed approach is designed with special attention to the practical application in two experimental systems, which results in considerable computational limitations. For this purpose, three different scene representations are investigated. These consist of the prevalent road topology as the global scene context and the drivable road area, which are both associated with the static environment. In addition, the detection and spatial reconstruction of other road users is considered to account for the dynamic aspects of the environment. In order to cope with the computational constraints, an approach is developed that allows for the simultaneous perception of all environment representations based on multi-task convolutional neural networks.

For this purpose methods for the respective tasks are first developed independently and adapted to the special conditions of traffic scenes. Here, the recognition of the road topology is realized as general image recognition. Furthermore, the perception of the drivable road area is implemented as image segmentation. To this end, a general image segmentation approach is adapted to improve the incorporation of the a-priori class distribution present in traffic scenes. This is achieved through the inclusion of element-wise weight factors through the Hadamard product, which resulted in increased segmentation performance in the conducted experiments. Also, a task decoder for the perception of vehicles is designed based on a compact 2D bounding box detection method, which is extended by auxiliary regressands. These are used for an appearance-based estimation of the orientation and dimension ratio of detected vehicles. Together with a subsequent method for the reconstruction of spatial object parameters based on constraints derived from the backprojection into the image plane, a scene description with all measurements for a basic environment model and subsequent automated driving functions can be generated. From the examination of alternative multi-task approaches and considering the computational restrictions of the experimental systems, an integrated convolutional neural network architecture is implemented, which combines all perceptual tasks in a single end-to-end trainable model. In addition to the definition of the architecture, a strategy is developed in which alternated training of the perception tasks, changing with each iteration, enables simultaneous learning from several single-task datasets in one optimization process. On this basis, a final experimental evaluation is performed in which a systematic analysis of different task combinations is conducted. The obtained results clearly show the importance of a combined approach to the perception tasks for automotive applications. Thus, the experiments demonstrate that the integrated multi-task architecture for all relevant representations of the scene is indispensable for practical models on realistic embedded processing hardware. Regarding this, especially the existence of common, shareable image features for the perception of the individual scene representations, which are clearly evident from the results, is to be mentioned.

Kurzfassung

Die Arbeit untersucht Wahrnehmungsmethoden mit monokularen Kameras für die Erzeugung eines grundlegenden Umfeldmodells im Kontext automatisierter Fahrzeuge. Der entwickelte Ansatz wird dabei mit Fokus auf die praktische Anwendung in zwei Versuchssystemen ausgelegt, woraus strikte Beschränkungen der rechentechnischen Ressourcen resultieren. Zu diesem Zweck werden drei verschiedene Szenenrepräsentationen untersucht. Diese bestehen aus der Straßentopologie als globalem Szenenkontext und dem befahrbaren Straßenbereich, welche beide dem statischen Umfeld zugerechnet werden. Darüber hinaus wird die Detektion und Rekonstruktion von anderen Verkehrsteilnehmern zur Berücksichtigung der dynamischen Umfeldanteile einbezogen. Um die rechentechnischen Einschränkungen zu berücksichtigen, wird ein Ansatz basierend auf Multi-task Convolutional Neural Networks entwickelt, welcher die gleichzeitige Wahrnehmung aller Umfeldrepräsentationen erlaubt.

Hierzu werden Ansätze für die Wahrnehmungsaufgaben unabhängig voneinander ausgearbeitet und an die Gegebenheiten von Verkehrsszenen angepasst. Die Erkennung der Straßentopologie wird dabei als allgemeine Bilderkennung realisiert. Darüber hinaus wird die Wahrnehmung des befahrbaren Straßenbereichs als Bildsegmentierung umgesetzt. Hierfür wird ein allgemeiner Ansatz zur Bildsegmentierung angepasst um eine stärkere Berücksichtigung der in Verkehrsszenen vorhandenen a-priori Klassenverteilung zu erzielen. Dies erfolgt durch elementweise Gewichtungsfaktoren mittels des Hadamard Produkts, was im Experiment zu einer gesteigerten Segmentierungsgüte führte. Ebenso wird zur Wahrnehmung anderer Fahrzeuge ein Verfahren zur Detektion von 2D Bounding Boxen um zusätzliche Hilfsregressanden erweitert. Diese dienen zur Erscheinungs-basierten Schätzung der Dimensionen sowie der Orientierung detektierter Objekte. Zusammen mit einer Rekonstruktion der räumlichen Parameter durch aus der Rückprojektion in die Bildebene abgeleitete Zwangsbedingungen kann eine für nachfolgende Fahrfunktionen geeignete Objektbeschreibung erzeugt werden. Weiterhin erfolgt, hergeleitet aus der Betrachtung alternativer Multi-task Ansätze und unter Berücksichtigung der rechentechnischen Beschränkungen, die Integration in ein Convolutional Neural Network welches alle Wahrnehmungsaufgaben kombiniert. Zudem wird eine alternierende Trainingsstrategie vorgestellt, welche durch mit jeder Iteration wechselnde Wahrnehmungsaufgaben das simultane Anlernen von mehreren Single-task Datensätzen ermöglicht. Auf dieser Grundlage erfolgt eine abschließende Evaluation, bei welcher eine systematische Untersuchung verschiedener Aufgabenkombinationen erfolgt. Die erzielten Ergebnisse zeigen klar die Bedeutung einer kombinierten Betrachtung der Wahrnehmungsaufgaben für eine Anwendung in der Fahrzeugtechnik auf. So ergibt sich in Hinsicht auf die betrachteten Versuchssysteme, dass eine integrierte Wahrnehmung aller Szenenrepräsentationen für praxistaugliche Modelle unabdingbar ist. In diesem Zusammenhang ist besonders das aus den Ergebnissen ersichtliche Vorhandensein gemeinsamer, mehrfach nutzbarer Bildmerkmale für die Wahrnehmung der einzelnen Szenenrepräsentationen zu nennen.

Contents

Nomenclature

AOS	average orientation score
α	observation angle
$b, \mathbf{b}, \mathbf{B}$	bias scalar, bias vector, bias tensor
β	position angle
CS	cosine similarity
$\mathrm{concat}\left(\square\right)$	concatenation operator, stacks tensors along the u_3 dimension
$\square_{\mathrm{C}}$	marks the use of camera coordinates in [m]
χ	count of frequent classes according to the 85%-15%-rule
$\mathbf{d} = \left(d_{x_1}, d_{x_2}, d_{x_3}\right)^{\top}$	vector of 3D bounding box dimensions and its elements in [m]
$\mathrm{diag}\left(\square\right)$	diagonal matrix
$\mathcal{E}$	local environment in image space, receptive field
f	focal length of the camera system
$F1$	$F1$-score, harmonic mean of precision and recall
FP	number of false positive samples
FN	number of false negative samples
ϕ	roll angle
$\boldsymbol{\varphi}_{\mathrm{BP}}\left(\square\right)$	camera backprojection-line in parametric form
$\varphi_{\mathrm{a}}\left(\square\right)$	nonlinear activation function
$\varphi_{\mathrm{s}}\left(\square\right)$	softmax function
γ	learning rate for gradient descent optimization
g_{MA}	moving average of the squared gradients
$h, \mathbf{h}, \mathbf{H}$	feature scalar, feature vector, feature tensor (feature map)
$\mathbf{H}^{\diamond}$	feature map with additional rows and columns of zeros
$\mathcal{H}_{\mathrm{Rec}}$	set of features relevant for the topology recognition task
$\mathcal{H}_{\mathrm{Seg}}$	set of features relevant for the road segmentation task
$\mathcal{H}_{\mathrm{Det}}$	set of features relevant for the vehicle detection task
IoU	intersection over union, Jaccard index
IoU_{2D}	2D bounding box IoU in image space
IoU_{BEV}	2D BEV bounding box IoU in world coordinates
IoU_{3D}	volumetric cuboid IoU in world coordinates
i	gradient descent iteration count
$\square_{\mathrm{I}}$	marks the use of image coordinates in [px]
η	road topology class weight
j	general counter index
$\mathbf{K}$	intrinsic camera calibration matrix
κ	class index, discrete category in classification problems
l	indicates the depth of a given neural network layer
$L\left(\square\right)$	optimization loss
$L_{\mathrm{L1}}\left(\square\right)$	smooth L1 optimization loss

$L_{\text{nll}}\left(\square\right)$	negative log likelihood optimization loss, multi-class cross entropy
$L_{\text{Rec}}\left(\square\right)$	topology recognition task optimization loss
$L_{\text{Seg}}\left(\square\right)$	road segmentation task optimization loss
$L_{\text{Det}}\left(\square\right)$	vehicle detection task optimization loss
$L_{\text{2Dbox}}\left(\square\right)$	2D bounding box optimization loss
$L_{\varrho}\left(\square\right)$	vehicle dimension ratio optimization loss
L_{total}	total optimization loss that combines all perception tasks
$L_{\alpha}\left(\square\right)$	observation angle optimization loss
λ_{MA}	moving average decay factor
λ_{LR}	learning rate decay factor
mAP	mean average precision, area under the *pre-rec* curve
$\square_{\mu}$	index denoting a per-sample average, micro average
$\square_{\text{M}}$	index denoting a per-class average, macro average
N_{P}	number of neurons in a neural network layer
N_{L}	number of layers in a neural network
N_{batch}	batch size
N_{Θ}	number of all trainable model parameters
N_{train}	number of samples in the training dataset
N_{test}	number of samples in the test dataset
N_{val}	number of samples in the validation dataset
N_{k}	convolution kernel dimensions
N_{Det}	number of detected bounding boxes
N_{i}	total number of gradient descent iterations
N_{κ}	number of distinguished classes
$\mathbf{n}_{\text{C}}$	scene point position vector in camera coodinates
$\mathbf{n}_{\text{I}}$	scene point position vector in image coodinates
$\mathbf{n}_{\text{W}}$	scene point position vector in world coodinates
$\overset{\infty}{\mathbf{n}}_{\text{W}}$	vanishing point to a given scene point in world coordinates
$\mathbf{n}_{\text{W}}^{C}$	centroid of a 3D bounding box in world coordinates
ν	free parameter of the backprojection line in parametric form
$\mathbf{o}_{\text{C}}$	camera center in camera coodinates in [m]
$\mathbf{o}_{\text{W}}$	camera center in world coodinates in [m]
$\mathbf{o}_{\text{I}} = \left(o_{u_1}, o_{u_2}\right)^{\top}$	principal point of the camera system
$\mathbf{P} = \left(\mathbf{p}_1^{\shortmid}, \mathbf{p}_2^{\shortmid}, \mathbf{p}_3^{\shortmid}, \mathbf{p}_4^{\shortmid}\right)$ $= \left(\mathbf{p}_1^{-}, \mathbf{p}_2^{-}, \mathbf{p}_3^{-}\right)^{\top}$	camera projection matrix of size 3 × 4 and its column and row vectors
pre	precision, positive predictive value
pre_{interp}	interpolated precision
ψ	yaw angle
q	integer multiple of 2π, e.g. $q \in 2\pi \cdot \mathbb{Z}$
$\mathbf{R}$	general rotation matrix of size 3 × 3
rec	recall, true positive rate
ρ_{κ}	segmentation class a-priori probability

ϱ	aspect ratio of 3D bounding box width and length, e.g. $\varrho = \dfrac{d_{x_2}}{d_{x_1}}$
ϱ_{NMSE}	normalized mean squared error of the vehicle dimension ratio
s	stride, sliding window step size
TP	number of true positive samples
TN	number of true negative samples
$\mathbf{t}$	general translation vector
τ	decision threshold
θ	general notation of a trainable model parameter
$\Theta = \{\theta_1, \theta_2, \dots \theta_{N_\Theta}\}$	entirety of all trainable model parameters
ϑ	pitch angle
$\tilde{\square}$	accentuation for indicating the use of homogeneous coordinates
$\mathbf{u} = (u_1, u_2, u_3)^\top$	horizontal, vertical and feature channel dimension in image or feature map coordinates
$\mathbf{u}_{\mathrm{mid}} = (u_{1,\mathrm{mid}}, u_{2,\mathrm{mid}})^\top$	2D bounding box midpoint
v	general counter index
$\mathrm{vec}\,(\square)$	vectorization operator, converts a tensor into a column vector
$w, \mathbf{w}, \mathbf{W}$	weight scalar, weight vector, weight tensor
$\mathbf{W}_{\mathrm{k}}$	convolution kernel
w_{k}	convolution kernel element
w_{Rec}	weight of the topology recognition task in the optimization loss
w_{Seg}	weight of the road segmentation task in the optimization loss
w_{Det}	weight of the vehicle detection task in the optimization loss
$\square_{\mathrm{W}}$	marks the use of world coordinates in [m]
x_1, x_2, x_3	position in spatial world or camera coordinates in [m]
ξ	road topology class occurrence frequency
$y, \mathbf{y}, \mathbf{Y}$	target value, target vector, target tensor
$y_{\mathrm{mid},u_1}, y_{\mathrm{mid},u_2}$	2D bounding box midpoint target variables
$y_{\mathrm{w}}, y_{\mathrm{h}}$	2D bounding box dimensions target variable
ζ	gradient momentum weight factor
$\circ$	Hadamard product, element-wise product
$\lfloor\square\rfloor$, $\lceil\square\rceil$	floor and ceiling functions, Gauss brackets

Abbreviations and acronyms

ACC	adaptive cruise control
ADAS	advanced driver assistance system
BEV	bird's-eye-view
BL	back left
BR	back right
CNN	convolutional neural network
CPU	central processing unit
CRF	conditional random field
CUDA	compute unified device architecture
FCN	fully convolutional network
FL	front left

FR	front right
GPU	graphics processing unit
ILSVRC	ImageNet large scale visual recognition challenge
InVerSiV	Intelligente Verkehrsinfrastruktur für sicheres vernetztes Fahren in der Megacity
IPM	inverse perspective mapping
KITTI	Karlsruhe Institute of Technology, Toyota Technological Institute
LIDAR	light detection and ranging
LLS	linear least squares
MAC	multiply-accumulate operation
ML	maximum likelihood
MLP	multi-layered perceptron
NAS	neural architecture search
NMSE	normalized mean squared error
px	pixel, image point
RELU	rectified linear unit
SGD	stochastic gradient descent
SSD	single shot detection
YOLO	*here*: you-only-look-once

1 Introduction

The present dissertation investigates the topic of semantic scene understanding from monocular camera images. To this end, the focus is on automotive applications in the context of the future development of fully automated road vehicles. In the introductory chapter, firstly, a motivation of this research field is presented and secondly, a description of the goals and contributions of the present dissertation is given. Furthermore, a short overview of the subsequent structure of the thesis is outlined.

1.1 Motivation

Road vehicles are one of the most important means of transport in Germany and worldwide. Consequently, the number of vehicles in active use has been rising continuously for years. For example, [Fed20a] describes an increase of 16.5 % for the number of registered passenger cars in Germany from 1995 to 2019. Due to the resulting increase in traffic density, it could be assumed that the requirements for safe vehicle guidance have also risen equally. Counter intuitively, the traffic statistics indicate a contrary development of the reported number of accidents. This becomes particularly clear when the relative change in the number of registered passenger cars and the number of fatal accidents is considered. A corresponding illustration of these developments is shown in Figure 1.1. The statistics reveal that during the above-mentioned increase in the number of registered passenger cars, a reduction of 67.6 % in the number of road fatalities was recorded over the same period.

An often stated reason for this seemingly contradictory trend is the increased market penetration of modern vehicles with advanced safety systems. In the past, this development initially took place in the area of passive safety systems such as seat belts and airbags, which are intended to protect vehicle passengers in the event of accidents. In the further progress, a growing number of active systems were introduced, which aim to prevent or mitigate accidents. Their range of action is thus set in the early pre-collision accident phase before the first impact. Some examples of early active safety systems are the anti-lock braking system and the electronic stability program. These primarily serve to stabilize the vehicle and are activated only in emergency situations such as emergency braking or loss of control in curves. The development of active safety systems has then continued to evolve towards even earlier accident phases, to be

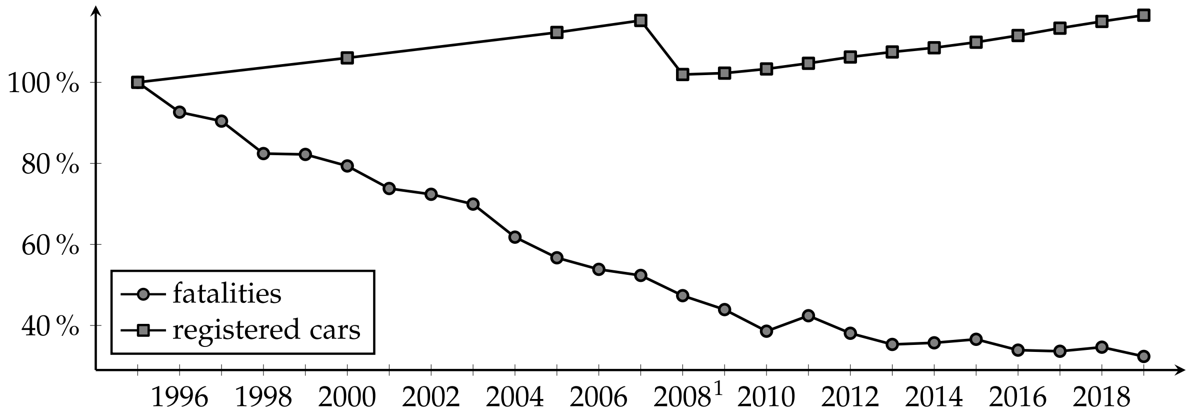


Figure 1.1: Development of the number of registered passenger cars and traffic fatalities in Germany up to the year 2019, relative to the level of 1995. Data taken from [Fed20a] and [Fed20b].

able to prevent critical situations from arising in advance. Examples of such systems include lane keeping assist and *adaptive cruise control* (ACC). The latter, for example, prevents the vehicle from falling below the safety distance, so that the occurrence of accidents based on insufficient distance can be completely avoided.

A characteristic feature of these systems is that, in addition to a safety effect, they also offer a certain increase in comfort by supporting the driver in everyday driving situations. Technologically, their development is accompanied by incorporating environment information about the present traffic scene into the systems. Moreover, a trend is apparent in current developments that gives rise to the requirement of an increasingly detailed and comprehensive perception of the given traffic scene. For example, existing implementations of an ACC focus on the perception and tracking of the vehicle directly in front of the ego vehicle. In this respect, newer systems such as emergency evasion or lane change assistant need to take all potential road users into account and also require a detailed acquisition of the drivable road area and the course and topology of the road.

Additionally, it is expected that the trend towards further increased demands on the perception of traffic scenes will continue in the future. Towards this, an analysis of the remaining potential for increased traffic safety through automatic systems reveals that further opportunities for improvement are almost exclusively based on systems with high demands on environment perception. For example, a study conducted by the German Federal Highway Research Institute (Bundesanstalt für Straßenwesen) expects the most significant effect on traffic safety from the market penetration of universal chauffeur systems in the sense of fully automated vehicles [Rös+19]. The Society of Automotive Engineers draws a similar course of the future development with its definition of the stepwise increase of automation levels up to the final vision of fully automated vehicles [Soc18]. These envisioned systems place the highest demands on the automatic perception of traffic scenes, which are equal to or exceed those of human drivers. Beyond the actual perception, further requirements have to be considered,

[1] From 2008 onwards, cars with a temporary deregistration were excluded from the official statistics of registered passenger cars.

which result from technical restrictions due to the use of mobile-capable hardware and the intended integration into mass products. These, in part contradictory, requirements can be summarized as follows:

- Provide a detailed and high performance environment perception sufficient for the safe execution of the driving task
- Compliance with the computational constraints of embedded automotive hardware systems with respect to computing power and runtimes
- Consideration of economic limitations regarding the unit costs in large-scale series production

Although these initial requirements are stated in very general terms, they already indicate the unique potential of camera-based systems for the practical perception of traffic scenes, which will be discussed in detail in the following.

Traffic scene perception with automotive cameras

One of the main questions in developing the perception of traffic scenes for automated vehicles is that of sensor technology. Here, cameras have significant cost advantages compared to other technologies. This is due to the advance in the technological development of image sensors but also due to the economies of scale resulting from the ubiquitous use of imaging sensors within the last years. As a result, high-quality cameras are now available at low cost in numerous, widely used electronic devices such as laptops, tablets, and smartphones. Apart from economics, there are also other arguments in favor of cameras for traffic scene perception.

This becomes particularly apparent by a comparison with the biological predecessors of automated vehicles. For example, the horse-drawn carriage, the preferred vehicle of earlier centuries, has been known to mankind for a long time as a highly autonomous means of transport. Here, the horses have their own sensory organs allowing them to perceive their surroundings. This enables them to exhibit appropriately adapted behavior and to perform the driving task with an increased degree of autonomy compared to today's cars. For instance, horses can recognize impending dangers and prevent accidents largely by themselves. Likewise, horses do not deviate from the road even without intervention by the coachman. In familiar surroundings, horses can even perform navigation and find the way home on their own.

To this end, the perception of traffic elements in their surroundings, such as the course of the road or the position of other road users, is predominantly based on visual information [Fle+03]. Also, the comparison with human drivers, e.g. in user studies on teleoperated driving [Geo+18], indicates that visual data contains basically all relevant information required for the driving task. Even if the actual image sensors are already fully developed and available at low cost, a crucial challenge of the perception task remains, which is the extraction of the information relevant for the driving task from the image data. In fact, it is the prevailing opinion among most experts that environment information extraction is one of the greatest remaining challenges in the

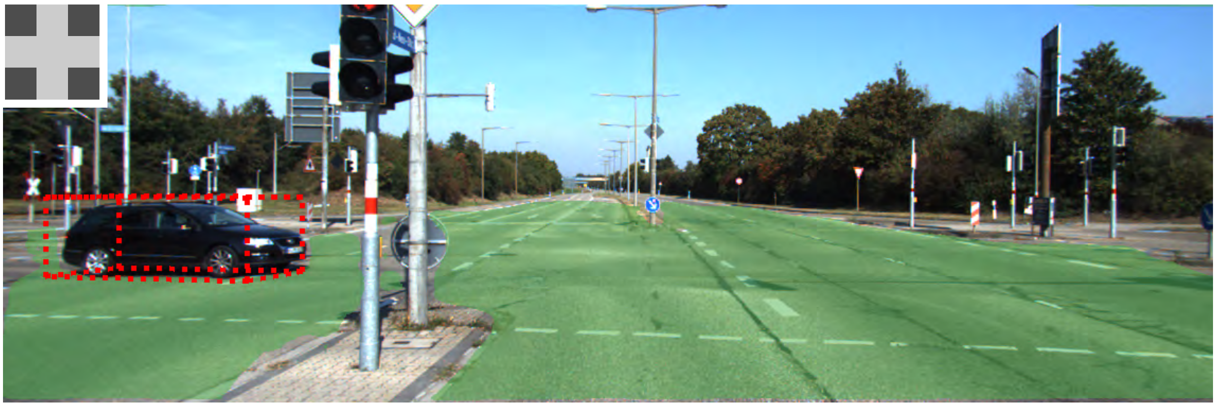


Figure 1.2: Illustration of three camera-based environment representations. Red: enclosing 3D bounding box of other present road users. Green: segmentation of the drivable road area. Top left: global context of the prevailing road topology

technological development of automated driving, see for example [Mat+15, p. 1145] or [Van+18].

From the perspective of a typical system architecture, the essential information about the traffic scene extracted by the perception system is passed on to downstream processing systems such as sensor fusion, maneuver or trajectory planning, etc. Therefore, the concrete goal consists in finding a representation of the relevant information that can easily be passed on to and be parsed by subsequent systems. Thus, the generated representation should be as compact and as meaningful as possible. To accomplish this goal, the principle of divide and conquer is applied to achieve a modularization into sub-problems. To this end, meaningful and compact representations of a traffic scene are firstly identified and secondly combined into a so-called environment model. The optimal combination of environment representations is still the subject of ongoing research. However, some representations can be named which are used particularly often in existing applications and research projects due to their outstandingly high relevance.

The corresponding camera-based environment representations, which will also be considered throughout the present dissertation, are illustrated in Figure 1.2. Here, the dashed lines (red) indicate the detection of an enclosing 3D bounding box around other road users within a scene as the first important representation of the environment. Furthermore, the Figure indicates the pixel-wise segmentation of the drivable road area in the image with a highlighted overlay (green). In addition, a third representation of the environment is given in the upper left corner of the image, which represents the global context of the prevailing road topology (here: intersection). For further clarification, an additional distinction is often discussed between dynamic, object-based representations (e.g. bounding boxes) and representations of the static environment components (e.g. free space map, road topology). Besides the definition of the environment representations, the consideration of methods and procedures which can generate these representations from camera images plays a vital role. Here, the biggest shortcoming of camera sensors becomes apparent, namely the complex data processing associated with their use. This is due to the comparatively large amounts of data due to the millions of image pixels of a camera, which all generate new measurements several times per second. This large amount of data results in complex processes for the extraction of the actually useful information. To address this challenge, again, the comparison with biological models found in nature can be helpful. For example, the visual cortex, which is part of the brain of higher mammals, is a known powerful visual processing system [Goo+16, pp. 353–359]. Following on from this, biologically inspired machine learning methods enable the highest quality of camera-based environment perception according to the current state of research. In particular, approaches based on so-called *convolutional neural networks* (CNNs) became popular, in which camera images are filtered through a hierarchical, multi-level network of simulated neurons. However, their increased perception performance is also accompanied by an increase in model complexity, which results in a substantial computational burden. This contradicts the limited computational resources available in embedded hardware systems used for driver assistance systems and automated driving.

Furthermore, it can be observed that significant development steps towards improving the perceptual ability have been made in the past by independently improving the methods of perception for individual environment representations. This was often based on a general formulation of the perception problem, without taking unique aspects of an application in the automotive field into account. However, as will be shown in the further course of this work, these aspects enable approaches that allow for a computational simplification or provide other advantages for the perception system. In this regard, particularly the simultaneous perception of several environment representations offers a promising opportunity to exploit synergy effects through a systematic avoidance of repeated computations. Through this, the computational burden of powerful CNN methods can be influenced positively towards practical automotive hardware requirements. The goals and contributions of the present thesis, which are established in relation to these preceding considerations, are discussed in detail below.

1.2 Outline and contributions

The goal of this thesis is the systematic development of a resource-constrained system for camera-based environment perception in the context of automotive applications in driver assistance and automated driving. The main focus here is on the practical applicability to the utilized experimental systems, which are geared to the technical requirements of common industrial applications in terms of hardware selection and accessible computational resources. In addition to computational efficiency, the emphasis is on the full integration of the scene representations and perception tasks required for a basic environment model. To achieve this goal, firstly, approaches for the independent prediction of individual scene representations are investigated and adapted to the specific conditions of traffic scenes. Secondly, a multi-task CNN architecture is developed, which consists of a common encoder stage and task specific decoders. The purpose of the encoder stage is to compute common image features that allow the simultaneous extraction of multiple scene representations. Compared to using separate models for each task, this avoids a redundant computation of image features and thus drastically increases the computational efficiency in comparison to the use of single-task models. The remaining structure of the present thesis is given as follows:

Chapter 2: This chapter contains a presentation of the current state of the art in the field of image processing with a focus on methods based on deep neural networks. Furthermore, it relates these methods to the most prevalent environment representations used in automotive applications. Moreover, a short overview of the fundamental definitions and principles in the context of CNNs is given.

Chapter 3: The practical implementation of the environment perception system presented in this thesis was carried out with two experimental systems, which are presented in this chapter. In addition to the technical overview, a short definition of the perspective projection model is given.

Chapter 4: This chapter covers the development of the basic CNN architecture used throughout this thesis. In this context, the general architecture decisions are discussed, and the multi-task approach is reviewed with respect to its impact on the network architecture and under consideration of its computational implications. Furthermore, a comparison of alternative encoder architectures and the final selection of the feature encoder are discussed.

Chapter 5: To represent the global context of a given traffic scene, firstly, the general context term is explained and the specific taxonomy used to represent the road topology is elaborated. Secondly, the classifier for road topology recognition is described. This is followed by a presentation of a road topology annotation dataset and a detailed evaluation to estimate the achieved recognition performance.

Chapter 6: The perception of the drivable road area through image segmentation is the subject of this chapter. First, the available alternatives are discussed, and the decoder architecture used for the subsequent implementation is selected. Next, an extension of the decoder architecture by explicit incorporation of the spatial class distribution is developed, and a comprehensive evaluation of the road area segmentation performance is carried out.

Chapter 7: For the perception of road users within a given traffic scene, an object detection is realized and described in this chapter. For this purpose, again, the discussion of possible alternatives, and the initial selection of the general decoder architecture is carried out. Subsequently, an approach to extend the 2D object detection decoder with auxiliary regressands for the purpose of geometrically reconstructing object viewpoints and spatial parameters is presented. Finally, an evaluation is performed to estimate the detection performance.

Chapter 8: A description of the implemented multi-task CNN architecture, including a discussion of the practical training strategy, is given. Furthermore, a comprehensive evaluation of all tasks for the fully integrated CNN and an investigation of the effects of different task combinations in the sense of an ablation study are carried out.

Chapter 9: This chapter summarizes the main results, supplements concluding remarks on the developed approaches, and suggests possible directions for future work.

In the course of these considerations, some particularly noteworthy insights have emerged. These main contributions of the present work can be summarized as follows:

- Comprehensive investigations on the recognition of the global traffic scene context are carried out. For this purpose, the scene context is classified to reflect the prevailing road topology. Furthermore, a dataset with corresponding annotations is prepared for the experimental evaluation.

- Studies on spatial priors present in traffic scenes as part of the drivable road area segmentation are conducted. To this end, a novel approach to explicitly model the spatial priors by integrating a position-dependant weight matrix (Hadamard multiplication) into a CNN architecture is developed.

- A thorough experimental analysis of a concrete CNN architecture following this approach and its impact on the drivable road area's segmentation performance is performed. In this context, particular emphasis is placed on compact and shallower CNN architectures that can maintain sub-sampled and thus computationally efficient feature maps.

- An exploration of possible assumptions and simplifications in the context of the spatial reconstruction of object detections is carried out. Here, special attention is paid to the particular requirements of traffic scenes.

- Based on this, a procedure for the spatial reconstruction of detected vehicles in traffic scenes is developed and analyzed. In doing so, a distinction is established between 3D parameters that can be reconstructed through geometric considerations and those that can be derived from the visual appearance of objects.

- A novel architecture that, by integrating multiple perception tasks, provides all image processing capabilities for a basic automotive environment model in one combined CNN is developed. Due to the resulting elimination of repeated computations, it can be demonstrated that this architecture can run even under the computational limitations of embedded hardware systems suitable for automotive applications.

- A systematic experimental analysis of the integrated multi-task architecture is performed. For this, firstly, the examination of hypothetical interactions between the perceptual tasks is made by leaning on the method of an ablation study. Secondly, the beneficial effects on the computational efficiency of the resulting system are investigated by means of a runtime analysis. Most notably, the results reveal that the integrated multi-task model requires 53 % less runtime compared to the sequential execution of single-task models.

2 Related Work and Fundamental Background

Due to, among other reasons, its high relevance for environment perception in the automotive context, image processing continues to arouse a high level of research interest. The development of modern methods based on CNNs can be traced back several decades, and only the accumulated knowledge of many incremental improvements enables models on par with today's state of research. Therefore, this chapter first gives an overview of the recent developments in image processing research. Subsequently, a description of the formal basis for the exact specification of CNNs, as they are applied in this work, is given. Moreover, the state of research is further divided into a presentation of the general developments on the one hand, and a grouping of specialized inventions in the context of automotive environment representations on the other hand.

2.1 Advances in CNN architectures for image processing

The current great interest in CNNs for image processing in research and industry began with the work of [Kri+12]. It achieved a reduction in the error rate of the annual *ImageNet large scale visual recognition challenge* (ILSVRC) [Rus+15] from 26% to 16%. This success is mainly due to two important factors. Firstly, extensive annotated datasets were now publicly available for the training, so that the problem of overfitting became manageable even with complex high capacity[2] models with many free parameters. Secondly, a high parallelization of the computations employed in CNNs using GPUs (*graphics processing units*) had a very advantageous effect.

As a result, the relative computational speedup on common hardware can be more than one order of magnitude, which accordingly supported the training of more complex networks. In the following time, CNNs became the general state of the art for image processing and camera perception, while new innovations were now driven by more sophisticated architecture design choices and training methods. Important architectural advances include [Lin+14], which proposes 1×1 convolutions, and [SZ15],

[2] The model capacity controls the scope of the types of mapping functions that the model can learn, see [Goo+16, pp. 107–113] or [Sha+17] for a detailed discussion.

which reveals the importance of network depth. Expanding on these works, [Sze+15] won the ILSVRC in 2014 by using the newly proposed inception-v1 architecture for CNNs. It employs submodules dubbed *inception modules*, and one of its major features is its efficiency in terms of computational resources. In [IS15] this architecture is refined to build the inception-v2 architecture. Furthermore, [He+16] introduces residual network connections that are utilized to feasibly train deeper CNNs than previously possible. The authors provide experimental results and theoretical foundations to justify their method, proposing a variant of their architecture that involves 152 layers and 60 million parameters. Following this, [Sze+17b] introduces residual inception modules and derives the inception-resnet-v2 architecture, which set a new record on the ILSVRC benchmark.

Unfortunately, as networks become deeper and larger, they require more memory and computation time, which oftentimes exceeds the capability of automotive embedded systems. Consequently, this was followed by introducing new architectures focusing specifically on computational efficiency and embedded applications. Notable examples include SqueezeNet [Ian+16], MobileNet [How+17], ShuffleNet [Zha+18], and Xception [Cho17]. A common feature of these architectures is the separation of pixel wise and dense convolutions by applying grouped and 1×1 kernels, and their competitive accuracy with smaller memory consumption and processing times compared to previous works.

Another more current trend is termed as *neural architecture search* (NAS) in the respective literature. This refers to the process of generating efficient network architectures within a carefully designed search space through an automatic search algorithm. The existing approaches differ primarily in the employed search strategy. Among others, there are methods based on evolutionary optimization [Rea+17; Rea+19], reinforcement learning [ZL17; Zop+18; TL19], and gradient based methods [Liu+19a; Xie+19]. The NAS-generated architectures have shown promising results, but they are subject to the major limitation of highly increased computational demands during the training phase compared to manually designed architectures.

2.2 Traffic scene representations from monocular cameras

In addition to the image processing methods, how the traffic scene is represented internally in an environment model is of great importance for implementing automated driving functions. Following [Die+05], an environment model is understood as a knowledge base that describes a theoretical and model-like representation of traffic elements in the real world. Usually, an automotive environment model combines several representations that differ according to the properties or effects of certain subsets of real world traffic elements. Road-related representations, as well as the representation of dynamic objects, offer both the highest relevance and the highest generalizability for the implementation of subsequent automated driving functions.

This is evident not only from the discussions in [Die+05] but also from the fact that these representations have been included in practically all relevant research work throughout the last years, see for example [Kas+11; Gre+12; Gre+14a; Sch+15; Eng+18;

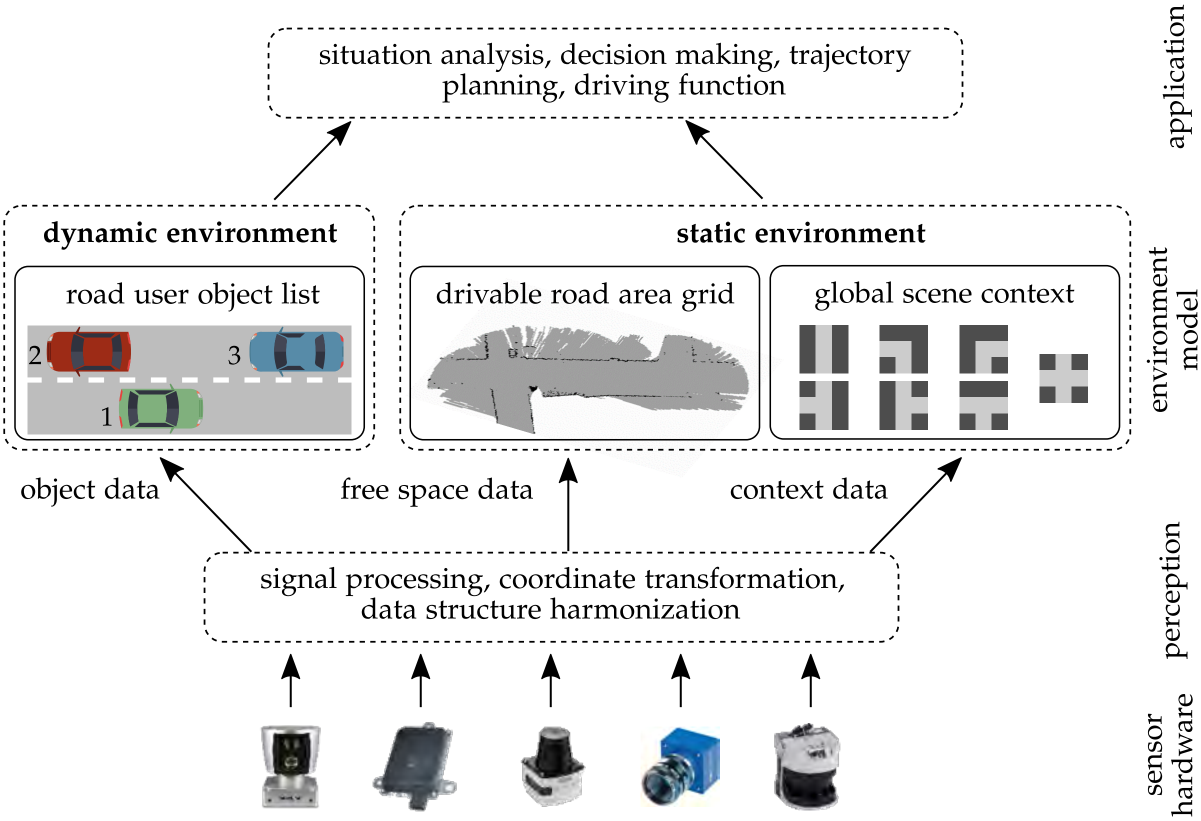


Figure 2.1: General structure of a basic automated driving processing stack with emphasis on the environment model according to [Gre+14a] and [Lie+19], supplemented by the global scene context as part of the static environment.

Lie+18; Aeb19], among others. To this end, road-related representations include the dimensions of the road surface, but oftentimes also extend to other classification parameters such as the general road topology or the number of lanes. Typically, the road-related representations are associated with the static environment components. In contrast, the description of dynamic objects usually contains different object model classes such as pedestrians or vehicles to represent other road users.

In addition to the comprehensive and complete representation of the traffic scene, the uniform abstraction of sensor data is an essential goal of environment models. An illustration of this uniform abstraction is shown in the generalized structure of a basic automated driving processing stack in Figure 2.1. This data processing structure has proven advantageous in particular due to the reusability of submodules and the independence of implemented functions from specific sensor configurations. Since this thesis addresses the efficient inference of scene representations from camera images for use in automotive environment models, the following sections contain further explanations on the corresponding state of research with respect to specific traffic scene representations.

Scene recognition for global-scale context information

In the literature, global context information on traffic scenes is often given as a taxonomy of discrete sets of characteristic properties [Fas+95]. Thus, the perception of context information is given by a classification of traffic scenes into typical categories representing some discrete properties of the scenes. Depending upon the application or driving function, the desired information can differ substantially. It may comprise, for example, the general road topology, the number of lanes, the presence of a construction site, or other predominant characteristics. For practical applications, global context information has been used both to implement plausibility checks for additional validation and to directly support driving functions [Wei+13; Rui+15; Sch16]. Primarily due to the safety requirement of self-contained operation, but also because the global context information may be subject to changes in the medium-term, e.g. when considering construction sites, the need for context perception with online sensors arises. Accordingly, several approaches for camera-based acquisition of context information are known in the literature. For clarity and since the general problem of image recognition has already been covered in the previous section, the following discussion focuses on works related explicitly to automated vehicles.

Towards this, the study in [Ess+09] analyses a two-step approach, in which a superpixel representation defines characteristic scene features. The context classification is carried out with respect to distinct road topology classes. With a similar objective, the work in [Kas+09] presents a different method for camera-based road type perception. It is based on feature engineering using the bag of visual words concept [FP05]. In the considered application, three different road types are distinguished and evaluated for a context-dependant adaptation of subsequent *advanced driver assistance systems* (ADAS). In another approach, [Sik+14; Sik+19] examine traffic scene context recognition for a vehicle fleet management application. They focus on compact feature descriptors to enable computation in remote back-end systems.

[Di+16; Di+17] examine global traffic scene context for landmark location recognition, using images taken under different weather or light conditions. Their approach is to extract CNN features and transfer the annotations from the retrieved, best fitting image based on a cross-domain, dense correspondence, where the domains reflect different light and weather conditions. The works in [Sch16; Tei+18] both consider global road type perception, here [Sch16] examines four context classes using traditional feature engineering while [Tei+18] employs a CNN-based approach to distinguish between two road type classes, e.g. highway and non-highway.

Moreover note, that the perception of global context information is not limited to camera sensors. To this end, the studies in [See+16; Hsu+17] present approaches for road type perception based on alternative or fused sensor data.

Semantic segmentation for spatial environment layout perception

Grid-based environment modeling was originally developed in the field of mobile robotics [Elf89] but was later also established in the automotive industry [Kas+11; Gre+14b; Sch+15; Eng+18]. The fundamental idea of this representation is to divide the environment into discrete cells on a horizontal 2D grid. The cells are assigned one of the states free, occupied, or unknown, often using the Dempster–Shafer framework [Dem68; Sha76]. Thus, measurements are not assigned to objects but to subregions of the vehicles environment. The absence of geometric or kinematic assumptions helps prevent false positive classifications (ghost targets) and the choice of simple features such as the cell-states enables simple sensor data fusion schemes [Gre+14a]. The generation of such a representation from camera data naturally corresponds to the problem of semantic segmentation. Here, semantic segmentation refers to the problem of assigning categories of objects or traffic elements to all image pixels.

Towards this, [Bru+15] proposes to run an ordinary recognition CNN multiple times for different sections of an image to obtain a dense segmentation output. A more sophisticated approach is presented by [She+17], who introduces a CNN architecture, named the *fully convolutional network* (FCN), that is designed to solve the task of segmentation in an end-to-end trainable manner. Herein, a recognition CNN is adapted for the task of segmentation by branching the network at intermediate layers and combining these branches into a pixel-dense new output path with preserved spatial resolution. Building on this, another approach combines CNNs and *conditional random field* models (CRFs) to perform semantic segmentation, see [Lin+16; Che+17]. Here, it is generally argued that CNNs perform especially well for feature representations, while CRFs capture contextual relation modeling.

Another semantic segmentation method relies on the FCN architecture in conjunction with dilated convolutions [YK16] and very deep residual models. [Wu+19c] refines the ResNet architecture by inserting additional residual units in a parallel manner and dub their approach as wide ResNet. [Zha+17] also builds upon the ResNet architecture, adding parallel computation of multiple pooling layers of different dimensions. An alternative approach is proposed by [Hon+15]. Here, the classification and segmentation tasks are decoupled and independently performed by two CNN models. Both models are trained separately such that a scalar class label is determined first, and the object contours are computed afterward. Additionally, [Har+15] introduces an approach termed hypercolumn that, similar to [She+17], combines information from coarsely resolved deeper layers and information from finer resolved shallow layers to form pixel descriptors. These descriptors constitute the input for the final classification step that obtains a dense segmentation output. More recently, [Wu+19b] addresses the heavy computational demand of dilated convolutions by replacing them with an approximation strategy based on regular convolutions. The approach is dubbed as joint pyramid upsampling and shows promising results by being able to reduce computation times many times over.

Related to semantic segmentation is the problem of instance segmentation, which is the task of detecting and delineating each individual object that appears in an image. Thus, unlike regular semantic segmentation, objects that are close to each other are not assigned to the same region. The approach in [Pin+15] learns to propose candidate segments, which are subsequently classified by an object detector stage. Another approach, which has been dubbed as Mask R-CNN [He+17], uses the exact reverse processing sequence by first evaluating an object detector stage and subsequently using an additional FCN-like network branch to predict object contours. Furthermore, a mixed approach dubbed panoptic segmentation has been proposed, which only provides explicit instance information for countable object categories [Kir+19]. Other (background) categories are dealt with using regular semantic segmentation. Therefore panoptic segmentation is a combination of instance and semantic segmentation. However, examples of automotive applications that benefit from instance-based contour information are hardly documented in the literature.

The scene representation predicted by semantic segmentation yields a 2D description of the environment in the image space. As explained before, it is often necessary to derive a geometric description in the form of a grid-based occupancy map for further use in subsequent driving functions. Moreover, a full semantic segmentation is unnecessarily complicated for the desired representation since it results in a rather detailed differentiation of categories, see for example [Cor+16; Sen+12]. In fact, a binary differentiation into free or occupied areas is sufficient for the generation of occupancy grids. Therefore, in the automotive context, semantic scene segmentation is often defined differently by considering only the binary problem and evaluating performance measures on a geometric scene description instead of an image space description [Fri+13].

Bounding box detection for object-based environment modeling

Object-based environment modeling aims to detect dynamic traffic elements such as road users in the ego vehicles surroundings and determine a spatial description of their position and, in some cases, also their orientation and dimension. An object representation from camera images is obtained using bounding box detection methods; however, these often focus on 2D bounding boxes in the image space as a first intermediate representation.

Modern image-based bounding box detection methods using CNNs fall into one of two categories: region-based detectors and non-region-based detectors. Generally, region-based detectors are accurate but relatively slow [Hua+17]. Their fundamental idea is to frame localization as a classification problem by finding image regions that correspond to object hypotheses and classifying each region individually. For example, the R-CNN method [Gir+14] generates region proposals, uses a CNN to extract features from these proposals and an output classifier for the final evaluation. Fast-RCNN [Gir15], Faster-RCNN [Ren+17], and Mask-RCNN [He+17] are modified versions that improve certain steps but generally share the same logic.

In contrast, the main idea of a non-region-based detector is to directly map image pixels to coordinates of bounding boxes. This category includes the approach of [Liu+16a] dubbed as *single shot detection* (SSD) and [Red+16] dubbed as *you-only-look-once* (YOLO). Another more recent work modifies the YOLO approach by using a deeper and more sophisticated CNN for feature encoding [RF18]. In comparison, these methods are more efficient in terms of speed and memory consumption. However, non-region-based models do not perform as well due to the background-class imbalance problem, which stems from the fact that many image locations are evaluated, but only few locations contain objects. The work in [Lin+20] addresses this problem with notable success by modifying the standard cross entropy loss to decrease the weight assigned to background locations.

Several methods are known in the literature to move from 2D bounding boxes to a spatial reconstruction purely based on image cues. [KK19] performs 2D bounding box detection after first applying a bird's-eye-view (BEV) transformation. Thus the spatial layout can be directly recovered. Similarly, [Rod+19] uses BEV features for bounding box detection. Another method is to apply point cloud object detection methods after first generating pseudo point cloud features from a learned depth map. Some notable examples among the works exploring this idea are [XC18], [Wan+19a], and [WK19].

The works in [Lep+09], [Kun+18], [Man+19], and [HS19] all follow the general pattern of predicting correspondences between image keypoints and a 3D model and fit the 3D pose. An alternative approach is to render 3D object models at different poses, backproject into the image plane and measure the similarity between the rendering and the detection window online, see for example [Mot+15], [Cha+17], and [Bar+20].

Another basic idea is to assume a tight fit between 2D bounding boxes and backprojected 3D bounding boxes and exploit this as an additional geometric constraint, as introduced by [Mou+17]. The works in [Liu+19b] and [Nai+19] somewhat relax the tight fit assumption by allowing for explicit offsets of the backprojected bounding box. In a related approach, [Ku+19] establish geometric constraints based on the backprojection of the bounding box centroid coordinates. [Li+19] presents an approach where the tight fit constraint is augmented by an additional orientation prediction based on the object's visual appearance.

To directly predict 3D bounding boxes, [Che+16b] lends from the idea of proposal based 2D detectors by sampling proposal boxes in 3D space. In another approach, [Sim+19] modifies the traditional object detection loss functions to include a 2D *IoU* (intersection over union) component and a 3D corner alignment component that are trained alternatingly. Similarly, [BL19] directly predicts all necessary information for 3D bounding box detection within one end-to-end model.

2.3 Fundamental principles and general framework

The subject of this section is to provide a general understanding of CNNs as a basis for the present work, with some simplifications for the sake of clarity. Many fundamental concepts can be explained using the historical perceptron neuron model [Ros58]. According to this, a *multi-layered perceptron* (MLP) consists of several layers of simulated neurons arranged as nodes in a directed acyclic graph. In a MLP, all neurons of one layer are fully connected to all neurons of the following layer. Furthermore, each neuron computes a weighted linear combination of the previous layers output values and subsequently applies a nonlinear activation function. The notation used in the following is inspired by the works [Far+13; Vog18], where scalars are given as italic letters, vectors as bold lowercase letters, and matrices and tensors as bold uppercase letters. Accordingly, the mapping of a layer l with N_P neurons can be described as:

$$\mathbf{h}_l = \begin{pmatrix} h_{1,l} \\ \vdots \\ h_{N_\mathrm{P},l} \end{pmatrix} = \begin{pmatrix} \varphi_\mathrm{a}\left(\mathbf{w}_{1,l} \cdot \mathbf{h}_{l-1} + b_{1,l}\right) \\ \vdots \\ \varphi_\mathrm{a}\left(\mathbf{w}_{N_\mathrm{P},l} \cdot \mathbf{h}_{l-1} + b_{N_\mathrm{P},l}\right) \end{pmatrix} = \varphi_\mathrm{a}\left(\mathbf{W}_l \cdot \mathbf{h}_{l-1} + \mathbf{b}_l\right) \ . \tag{2.3.1}$$

The respective feature maps (input and output values) are given by $\mathbf{h}$. The vector $\mathbf{b}$ contains the bias and the matrix $\mathbf{W}$ the weights, which together form the model's trainable parameters. The nonlinear activation function is represented by $\varphi_\mathrm{a}(\square)$. The description of a MLP with N_L network layers is therefore given as follows.

$$\mathbf{h}_l = \varphi_\mathrm{a}\left(\mathbf{W}_l \cdot \mathbf{h}_{l-1} + \mathbf{b}_l\right) \ \ \forall \ l \in \{1, 2, \dots, N_\mathrm{L}\} \tag{2.3.2}$$

From this formal description, it can be seen that basic neural networks are mathematically rather simple entities consisting mostly of stacked matrix-vector multiplications interleaved with nonlinear activation functions.

Convolutional neural networks and their components

CNNs have some specific characteristics that distinguish them from regular MLP networks. Compared to MLPs, CNN architectures consist not only of the matrix vector multiplication and the nonlinear activation function but additionally incorporate pooling (sub-sampling) and convolution components. Usually, these components are referred to as layers, synonymous with the MLP network's previously discussed layers. Furthermore, when applied to camera data, it becomes clear that the input measurements are structured in two dimensions. It follows that the features are now represented by $\mathbf{H}$ to reflect the multi-dimensional nature of the data.
In fact, in any practical implementation $\mathbf{H}$ becomes a tensor with at least three dimensions (u_1, u_2, u_3). This is because, additionally to image coordinates, camera data is usually also structured along a channel dimension, which is encoded in the third tensor coordinate u_3. For the input images, this corresponds to the color channels, in the network's remainder the term feature channels is used. Furthermore, the pooling and convolution layers are characterized by the fact that only input values within a local environment are evaluated for the computation of an output value. Originally, a similar structure has been termed as the receptive field in neurophysiological studies [HW59; HW62] and it can be illustrated as shown in Figure 2.2. A detailed description of the individual components is provided in the following paragraphs.

Convolutional layers: As the name indicates, these layers perform a convolution operation on the input feature maps $\mathbf{H}_{l-1}$. The convolution kernels $\mathbf{W}_\mathrm{k}$ form the trainable model parameters of this layer, which creates the local connectivity consistent with Figure 2.2. The convolution kernel's elements are identical for all image positions, a mechanism also known as weight sharing. Formally, the relationship between the input and output data of a convolutional layer is given by:

$$\mathbf{H}_l = \mathbf{H}_{l-1} * \mathbf{W}_{\mathrm{k},l} + \mathbf{B}_l \ . \tag{2.3.3}$$

Note that the 2D convolution can be formulated as a matrix multiplication using the im2col approach [RX15], so that CNNs too can be considered as stacked matrix multi-

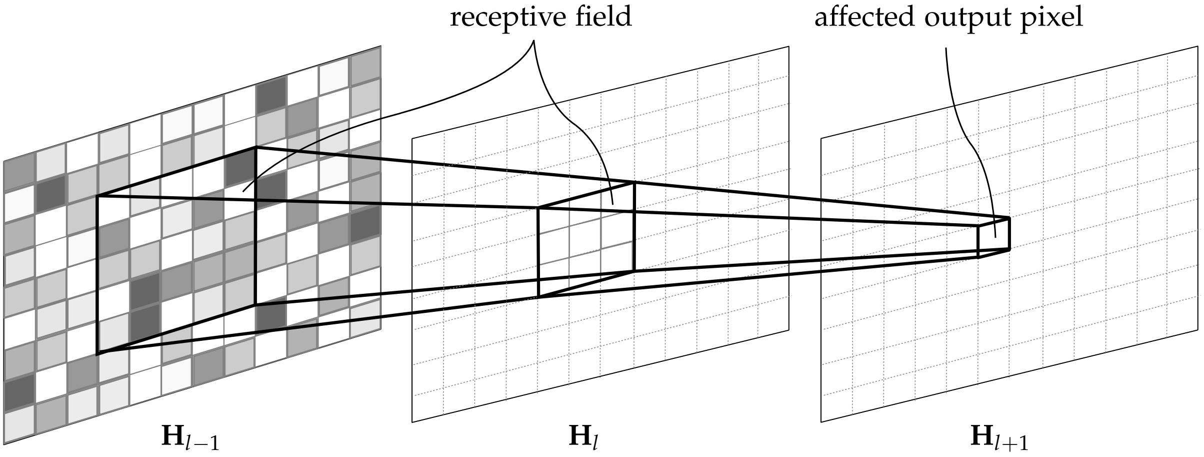


Figure 2.2: Illustration of the local connectivity pattern resulting from two successive 3 × 3 convolutions. Inspired by biological analogies, these are referred to as receptive fields.

plications. In convolutional layers, the requirement for identical parameters in all local environments also holds for the bias **B**. Due to this, the convolutional layer becomes spatially invariant, since a spatial translation of the input data leads to a corresponding translation of the output data but otherwise leaves the output unchanged. The number of free parameters therefore corresponds to the size of the convolution kernel $\mathbf{W}_\mathrm{k}$ and the number of independent elements of **B**.

Fully connected layers: These correspond to conventional MLP neural networks, since every neuron processes all outputs from the previous layer. Therefore, receptive fields do not apply in this type of layer. As noted previously, the full connection of succeeding network layers corresponds to a matrix-vector multiplication. In the case of multidimensional data, the feature maps therefore have to be considered in vectorized form. If $\mathrm{vec}\left(\Box\right)$ denotes a vectorization operator, the fully connected layer is defined by:

$$\mathbf{h}_l = \mathrm{vec}\left(\mathbf{H}_{l-1}\right) \cdot \mathbf{W}_l + \mathbf{b}_l \quad . \tag{2.3.4}$$

The number of trainable parameters of the fully connected layer is equivalent to the product of its input and output dimensions accounting for the weight matrix **W**, plus the bias parameters **b**. The comparably huge increase in computational complexity means that typically only a small part of the network is implemented as fully connected layers.

Activation functions: In the historical developments on neural networks, the step function and later the hyperbolic tangent function were used as activation functions due to their high similarity with biological models. However, these activation functions are not well suited for gradient-based training of the model parameters due to their non-existing or extremely small gradients. This is particularly problematic since, in deep neural networks, gradients are generally computed by the chain rule. Effectively, this results in a multiplication of many small gradient values, leading to the vanishing gradient problem [Hoc91]. Therefore, in recent works, the activation function is more closely selected according to its mathematical properties. Towards this, [Hah+00] propose the use of the *rectified linear unit* function (RELU). Other variants include the leaky RELU function [Maa+13], the parametric RELU [He+15a] and the exponential linear unit function [Cle+16]. Due to its empirical success for image processing problems, the original RELU function still is most widely used today for CNNs [Nwa+18]. Mathematically, it can be described as follows.

$$\mathbf{H}_l = \varphi_{\mathrm{a}}\left(\mathbf{H}_{l-1}\right) = \max(0, \mathbf{H}_{l-1}) \tag{2.3.5}$$

Herein, φ_{a} denotes the respective activation function. Even though their primary motivation stems from their mathematical properties, they have also been related to biological processes [Glo+11].

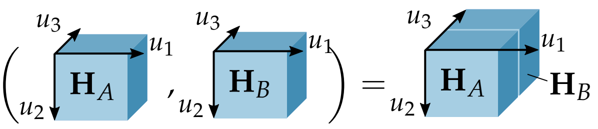
Pooling layers: This type of layer is commonly used to reduce the spatial dimensions of the input feature maps for the following layers. Note that the number of feature channels u_3 is not affected by pooling. The dimension reduction is achieved through sub-sampling, which results in a loss of information. However, it also leads to a reduction of the overall computational demand, which can play a decisive role and outweigh any information loss. Similar to the convolutional layer, pooling operates on a local environment of the feature map. Various strategies for this are documented in the relevant literature. The obvious approach is to implement pooling by averaging over a local environment. Other variants discussed in the literature are stochastic pooling [WG15] and spatial pyramid-pooling [He+15b]. It is, however, more common to implement pooling using the plain maximum operator [Sch+10; Bou+10]. Following the notation of [Gra14], a local environment (receptive field) of size s is given by $\mathcal{E}$. Then, the maximum pooling layer can be expressed as:

$$\begin{aligned} h_l\left(\mathbf{u}\right) &= \max_{(\Delta u_1, \Delta u_2) \in \mathcal{E}} h_{l-1}\left(\mathbf{u} + (\Delta u_1, \Delta u_2, 0)^\top\right) \quad , \\ \mathcal{E} &= \left\{0 - \left\lfloor \frac{s}{2} \right\rfloor, 1 - \left\lfloor \frac{s}{2} \right\rfloor, \ldots, s - 1 - \left\lfloor \frac{s}{2} \right\rfloor \right\}^2 \quad . \end{aligned} \tag{2.3.6}$$

Herein, $\mathbf{u} = (u_1, u_2, u_3)^\top$ applies. The effect of sub-sampling is achieved by shifting the local environment of the input feature maps using a sliding window with a fixed step size. Often this step size is equal to s, then s is also referred to as stride.

Concatenation: Besides, the concatenation of features will become relevant in various places throughout this thesis. Here, no actual processing of the input features occurs, as the input feature's values are retained and merely rearranged to form the output feature tensor. More precisely, concatenation describes the stacking of several input tensors along the dimension of the feature channels u_3. If concat $(\square)$ describes the concatenation operator, its effect can be illustrated as follows.

$$\operatorname{concat}\left(\mathbf{H}_A, \mathbf{H}_B\right) = \operatorname{concat}\left(\ldots \right) = \ldots \tag{2.3.7}$$



Identification and loss functions

All network layer's parameters are learned based on a set of N_{train} input images $\{\mathbf{H}_{0,1}, \mathbf{H}_{0,2}, \dots, \mathbf{H}_{0,N_{\text{train}}}\}$, the corresponding output data $\{\mathbf{H}_{N_{\text{L}},1}, \mathbf{H}_{N_{\text{L}},2}, \dots, \mathbf{H}_{N_{\text{L}},N_{\text{train}}}\}$, and a ground truth annotation of the desired output data $\{\mathbf{Y}_{N_{\text{L}},1}, \mathbf{Y}_{N_{\text{L}},2}, \dots, \mathbf{Y}_{N_{\text{L}},N_{\text{train}}}\}$. For this purpose, a loss function $L(\square)$ is minimized using the training examples. If the entirety of all trainable model parameters (weights **W** and biases **B**) is denoted as $\Theta = \{\theta_1, \theta_2, \dots \theta_{N_\Theta}\}$, then this is formally expressed as follows.

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \left(\frac{1}{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} L\left(\mathbf{Y}_{N_{\text{L}},j}, \mathbf{H}_{N_{\text{L}},j}\right) \right) \tag{2.3.8}$$

For continuous target variables, the smooth $L1$ norm is commonly used as a loss function. If $y_{N_{\text{L}},j}$ and $h_{N_{\text{L}},j}$ denote the elements of $\mathbf{Y}_{N_{\text{L}},j}$ and $\mathbf{H}_{N_{\text{L}},j}$, it is given as follows.

$$L_{\text{L1}}\left(\mathbf{Y}_{N_{\text{L}},j}, \mathbf{H}_{N_{\text{L}},j}\right) = \sum_{\mathbf{u}} \begin{cases} 0.5 \cdot \left(y_{N_{\text{L}},j}(\mathbf{u}) - h_{N_{\text{L}},j}(\mathbf{u})\right)^2, & \text{if } \left|y_{N_{\text{L}},j}(\mathbf{u}) - h_{N_{\text{L}},j}(\mathbf{u})\right| < 1 \\ \left|y_{N_{\text{L}},j}(\mathbf{u}) - h_{N_{\text{L}},j}(\mathbf{u})\right| - 0.5, & \text{else} \end{cases} \tag{2.3.9}$$

For classification tasks, the distinguished classes κ are typically encoded in the u_3 dimension of the last feature map $\mathbf{H}_{N_{\text{L}}}$ and its ground truth $\mathbf{Y}_{N_{\text{L}}}$. The values of $y_{N_{\text{L}}}, h_{N_{\text{L}}}$ are interpreted as confidence scores. This corresponds to a one-hot encoding where $\kappa = u_3$ applies, e.g. assume the case that for a given sample of a dataset, the classification into seven different categories is examined. Then, for an example ground truth class $\kappa = 2$, the following $1 \times 1 \times 7$ vector $\mathbf{y}_{N_{\text{L}}}$ results.

$$\kappa = 2 \rightarrow \mathbf{y}_{N_{\text{L}}} = (0, 1, 0, 0, 0, 0, 0) \tag{2.3.10}$$

Generally, the confidence scores are normalized using the softmax function $\varphi_{\text{s}}(\square)$ so that the model outputs can be interpreted as class probabilities.

$$\varphi_{\text{s}}\left(h_{N_{\text{L}}}(\mathbf{u})\right) = \frac{\mathrm{e}^{h_{N_{\text{L}}}(\mathbf{u})}}{\sum_{\kappa=1}^{N_\kappa} \mathrm{e}^{h_{N_{\text{L}}}(u_1, u_2, \kappa)}} \tag{2.3.11}$$

Herein, again $\mathbf{u} = (u_1, u_2, u_3)^\top$ applies and N_κ denotes the total number of distinguished classes. Training the model then corresponds to a maximum likelihood (ML) problem, since the model parameters are selected to maximize the likelihood across all training examples. In practice, the negative log likelihood, also known as multi-class cross entropy, is minimized because of its advantages for representing the joint probability distribution of multiple independent variables [Goo+16, p. 128]. For classification tasks, a suitable loss function therefore is given by:

$$L_{\text{nll}}\left(\mathbf{Y}_{N_{\text{L}},j}, \mathbf{H}_{N_{\text{L}},j}\right) = -\sum_{\mathbf{u}} \ln\left(y_{N_{\text{L}},j}(\mathbf{u}) \cdot \varphi_{\text{s}}\left(h_{N_{\text{L}},j}(\mathbf{u})\right)\right) \ . \tag{2.3.12}$$

By repeatedly applying the chain rule, the gradient of the optimization loss with respect to the model parameters can be determined for all layers of the network. An iterative gradient-descent approach is then used to minimize the loss and fit the model parameters to the training data. In practice, using all available training examples to determine the gradient is very inefficient. Therefore, it turned popular not to use the entire training dataset but only a (pseudo-)random subset to determine the loss gradient. This method is called *stochastic gradient descent* (SGD), and the subset of training examples is referred to as a mini-batch with size N_{batch}. Since only an approximation of the gradient can be determined this way, further measures are used to support a robust convergence. For this, the standard SGD is extended by a momentum term, so that parameter updates also consider information about gradients of past iterations [Pol64]. Different adaptation strategies were proposed to facilitate the manual choice of the learning rate and the momentum term's weight, which scale the loss gradient. Prominent methods for this are AdaGrad [Duc+11], RMSProb [TH12], and Adam [KB15]. In the context of this thesis, the update rule according to [TH12] is employed. Herein, a weighted moving average, denoted as g_{MA}, is computed for the squared gradient in each iteration i with respect to each trainable model parameter θ_j as follows.

$$g_{\text{MA},j,i} = \lambda_{\text{MA}} \cdot g_{\text{MA},j,i-1} + (1 - \lambda_{\text{MA}}) \cdot \left(\frac{\partial L_i}{\partial \theta_j} \right)^2 \tag{2.3.13}$$

When determining the parameter update, the gradient is then divided by the root of the corresponding moving average value, which yields the following parameter update per iteration.

$$\Delta\theta_{j,i} = -\gamma \cdot \frac{\partial L_i}{\partial \theta_j} \cdot \frac{1}{\sqrt{g_{\text{MA},j,i}}} + \zeta \cdot \Delta\theta_{j,i-1} \tag{2.3.14}$$

The optimization parameters are thus given by the learning rate γ, the moving average decay parameter λ_{MA}, and the momentum parameter ζ. It has also been shown to be advantageous if feature maps consist of values from approximately the same range. This has consequences for the initialization of the model parameters, which are often adapted to the layer sizes of the network [GB10; He+15a]. Furthermore, it is state of the art to normalize the feature maps batchwise to explicitly generate values in the same range [IS15].

3 Experimental Setup and Data Acquisition

A camera system was set up and integrated into both a mobile and a static test platform to record the traffic scene images for the experimental investigations carried out within this work. A discussion of these two test platform's logical and physical structure is the subject of this chapter. Also, the used coordinate systems are defined, and a description of the transformations between these coordinate systems is given.

3.1 Outline of the camera system and test platforms

An overview to illustrate the interaction of the camera system's various components is shown in Figure 3.1. The Figure shows the distinction of the employed systems into an offline part and an online part. For the training of the developed models, the offline part performs the corresponding parameter optimization based on a database of annotated camera images using desktop PC hardware. For the execution of the trained models on real-time video data recorded by the camera system, the embedded hardware mentioned above is used by contrast.

The development of a reliable perception of the traffic environment sets unique challenges for the utilized camera. As a result, the image sensors used in automotive applications generally differ significantly from conventional cameras or smartphones. For example, unlike in consumer hardware, it is irrelevant for the application whether the camera system can produce visually appealing, good-looking images. Instead, machine perception of the traffic environment requires an image that is as close to reality as possible. For instance, the image sensor must have sufficient dynamic range to display scenes outdoors and in changing light and weather conditions. In addition, advanced features for adjusting the exposure time control and for synchronizing the acquisition play an important role. It should also be noted that the wide adoption of camera systems in the automotive field is only possible with particularly robust, durable, and cost-effective sensors.

A corresponding sensor was selected for the experimental setup taking these requirements into account, whose detailed technical specifications are listed in the appendix in Table A.2.a. At the time of writing, these conform to the state of the art available to the automotive industry, especially concerning dynamic range and color depth, see also [Cor17, p. 59]. However, it must be noted that this sensor is particularly suitable

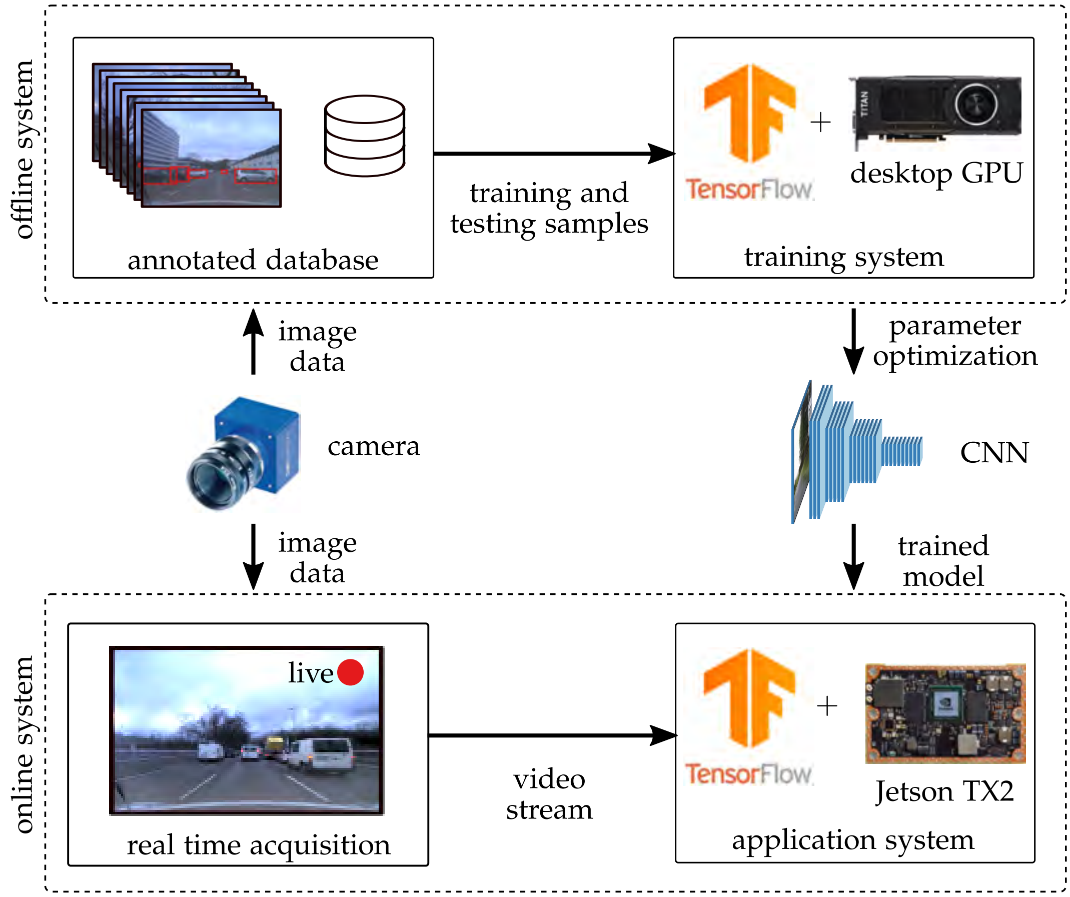


Figure 3.1: Technical overview of the employed camera system. First, the CNN models are generated in an offline part, then they are used in an online part for processing live camera views.

for use at low speeds and less for highway scenarios due to its operating principle of the rolling shutter[3]. The image sensor is used as an integral part of an industrial-grade camera. Furthermore, appropriate lenses for traffic scene perception should have as little optical distortion as possible, so that a pinhole camera model can easily reproduce their properties. This is to ensure that the captured image's peripheral areas remain usable for the environment perception and that the available image resolution can be fully utilized. Also, the achievable detection range resulting from the aperture angles of the lenses must be taken into account. For the employed camera system, the technical details of the lenses are given in the appendix in Table A.2.b. These are characterized by fixed aperture angles and a particularly low optical distortion.

Since automotive applications are often safety critical, a camera system must also provide the computing power required for highly accurate models and fast processing times. At the same time, restrictions in installation space and power consumption must be taken into account. Given this conflict of objectives, the choice of processing hardware is an important factor. A promising way to meet these requirements is to use embedded multicore systems, which can accelerate mathematical calculations re-

[3] A detailed investigation of the sensor characteristics was conducted in the master's thesis "Entwicklung einer Multi-Sensor-Datenfusion zur Umfelderfassung automatisierter Fahrzeuge" written in 2018 by A. Dikarew at the TU Dortmund Institute of Control Theory and Systems Engineering.

lated to signal and image processing through massive parallelization. In fact, several manufacturers have taken note of the emerging market for embedded machine learning applications and started developing specific multicore hardware for automated vehicles. In addition to the provided computing power, the available solutions also differ in their support for common machine learning libraries. At the time of writing of the present thesis, the processing hardware termed as Jetson TX2 and detailed in the appendix in Table A.2.c represents one of the leading available embedded systems. It is based on a multicore CPU + GPU setup and supports the CUDA (*compute unified device architecture*) programming interface's full instruction set [Nic+08]. This allows the use of virtually all state of the art machine learning libraries without extensive customization, especially without reducing the set of supported functions as is common with other alternatives [Goo19a; Goo19b]. Therefore, this processing hardware is used as the online application system in the following.

The implementation of the developed image processing models is largely based on the open source Tensorflow software framework [Aba+16] for the definition, training, and execution of deep neural network models. Here, the decisive advantage of Tensorflow over other alternatives is the support of the large community of active researchers and developers. This is also reflected in statistics such as the analysis of search engine trends or the number of development projects forked from Tensorflow [SJ19].

Mobile platform: test vehicle

The examination of the image processing models developed throughout this work is carried out mainly using two different test platforms. The described online part of the camera system was integrated into a test vehicle of the type Nissan Leaf ZE0 to set up the first test platform. Thus, the test vehicle's main purpose is to record some of the required datasets for training and evaluation of the developed perception system. For the positioning of the camera, multiple different aspects need to be considered. On the one hand, the mounting position should be as high as possible, since from this perspective, the mutual occlusion of road users occurs less frequently, thus enabling a more robust perception. On the other hand, the camera should be protected as much as possible from dirt and weather influences. This is why the installation behind the windscreen at the rear-view mirror's height has proven itself in practice. This position

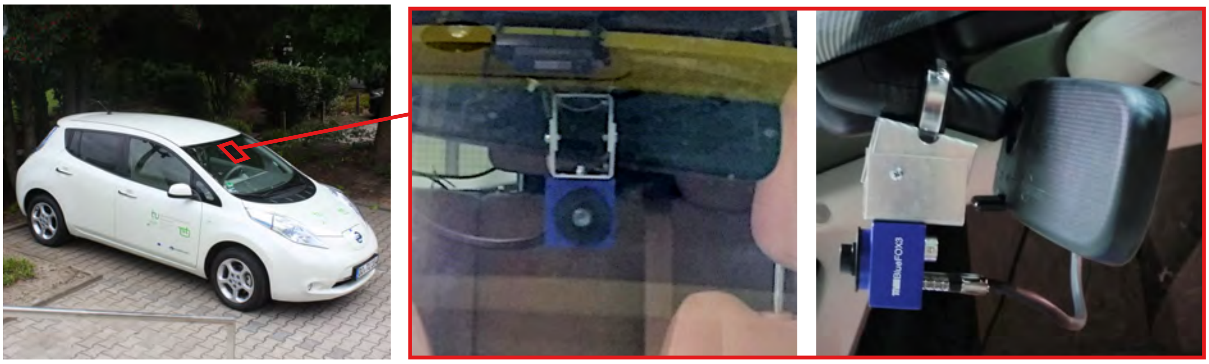


Figure 3.2: Overview of the used test vehicle with the camera fixture. The camera is positioned as high as possible while still being covered by the windscreen wiper's cleaning area.

allows the recording of the traffic scenes in the direction of travel, which is the most relevant area for the driving task. Furthermore, it is advantageous that this position is covered by the cleaning area of the windscreen wiper, so that disturbances caused by rain and other weather influences are reduced. Therefore, the camera is attached to the rear-view mirror mount using a specially designed fixture. An overview of the setup in the test vehicle is shown in Figure 3.2. This mobile platform allows to capture the traffic scene while driving, which has the advantage of allowing a realistic examination of the developed models with recordings from actual road traffic. In contrast, the influence of ego-movements on the measurements is disadvantageous. In particular, any rolling and pitching movements of the test vehicle can have substantial effects on the obtained spatial reconstructions of the scene representations.

Static platform: road side unit

The second employed test platform was developed in the context of the interim field test of the InVerSiV project[4]. This platform is referred to as road side unit. It is intended for sensor-based traffic monitoring using elements of the permanently installed traffic infrastructure such as traffic lights or street lamps. Within the framework of a radio-supported, cooperative perception of the traffic scene, the road side units provide a means of extending the detection range of a vehicle's own sensors. Thus it becomes possible to also detect occluded road users, for example, behind the corner of a building at an intersection. For this purpose, the camera system was set up in a field experiment on a closed-off test track, whereby a truss construction was used for mounting the camera system and other sensor components.

Figure 3.3 gives an impression of the field test. This illustration shows that the operating conditions are different from those of the mobile test platform. In this regard, the road side unit concept represents a purely static platform with no inherent movements. Correspondingly, the sensor platform's orientation does not change and a static transformation can be used to estimate distances and spatial positions within the depicted scene. Furthermore, the installation is again carried out in a custom housing that minimizes weather influences such as rain or direct sunlight through a protective screen. Compared to the moving test platform, the intended mounting position on common infrastructure elements such as traffic lights, lanterns, or overhead sign gantries offers considerable advantages. Generally, the traffic scene is shown from a much higher perspective when viewed from this position. Therefore, when the camera system is used in the road side unit, direct lines of sight to most road users exist even in crowded scenes, so that mutual occlusions occur less frequently and have fewer adverse effects. Due to the closed-off test track and the synthetic traffic scenes simulated by test drivers, a disadvantage is that the resulting image data generally offers less variety and realism. For this reason, the second test platform is mainly used for a qualitative assessment of the generalizability of the developed image processing models towards altered operating conditions.

[4] *Intelligente Verkehrsinfrastruktur für sicheres vernetztes Fahren in der Megacity*, see https://www.inversiv.de/ . Accessed August 6th, 2020

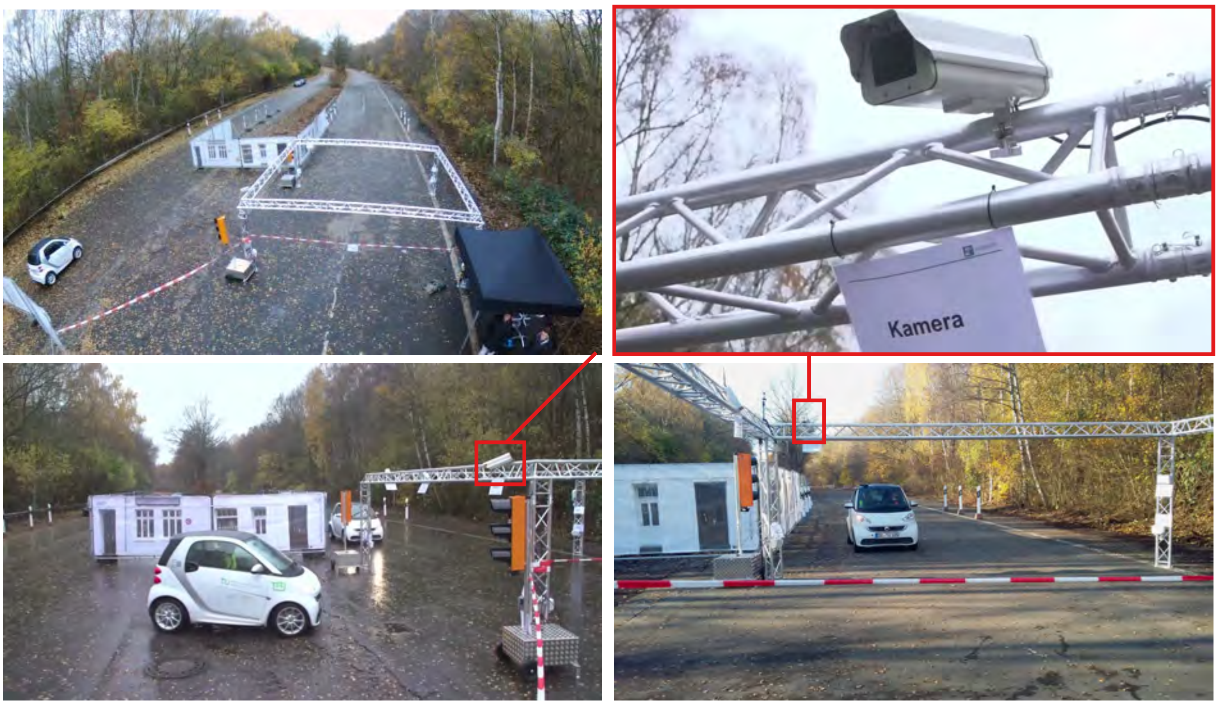


Figure 3.3: Overview of the InVerSiV field test with the road side unit camera platform. The cameras monitor the traffic scene from a stationary truss structure.

3.2 Inferring scene points from image space measurements

Since the planning of driving maneuvers and vehicle control algorithms typically work in metric 2D space aligned to the road surface, it is necessary to describe the location of relevant traffic elements within the scene in spatial coordinates. In the camera's perspective image, the spatial relationship between traffic elements cannot be determined directly. Instead, it is constrained indirectly from their 2D positions in the image.

This applies regardless of the type of representation. Therefore, this section first discusses the definitions of the used coordinate systems and the mapping of scene points from a 3D coordinate system into a 2D image coordinate system. Subsequently, the actually desired inverse mapping of image coordinates to spatial 3D coordinates is considered, and it is discussed how this mapping can be realized depending on the scene representation.

Camera model and coordinate systems

The mapping of 3D scene points to image coordinates is described by a camera model. The explanations in the following are based on [Pri12, pp. 359–363] and [HZ03, pp. 153–161].

The pinhole camera model is conceptually simple but widely used in practice. It essentially makes use of projective geometry to describe the coordinate mappings. This can be easily understood using an example object positioned at the 3D scene point $\mathbf{n}_C$, as shown in Figure 3.4. The depicted 3D coordinate system $(x_1, x_2, x_3)^\top$ is

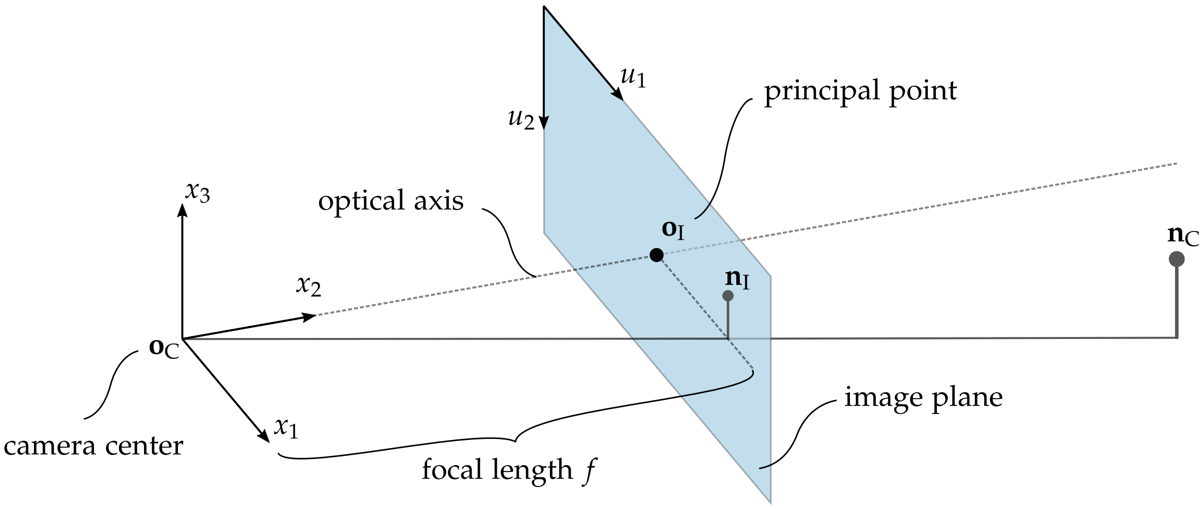


Figure 3.4: Schematic illustration of the central projection in the pinhole camera model. The object at the spatial position $\mathbf{n}_\mathrm{C}$ is mapped to the image point $\mathbf{n}_\mathrm{I}$.

named the camera coordinate system and marked as $\square_\mathrm{C}$. In the 2D representation of the scene, measurements are available in image coordinates $(u_1, u_2)^\top$, which are denoted as $\square_\mathrm{I}$ respectively. This image coordinate system is aligned with the image plane, such that the origin lies in the upper left corner. As in a real pinhole camera, an optical axis is defined orthogonal to the center of the image plane, and a focal length f is specified, which indicates the distance between the camera center and the image plane. If the camera coordinate system is aligned with the optical axis and the camera center, as shown in Figure 3.4, the mapping can be obtained by the following equation:

$$\tilde{\mathbf{n}}_\mathrm{I} = \begin{pmatrix} f & 0 & o_{u_1} \\ 0 & f & o_{u_2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n_{x_1} \\ n_{x_2} \\ n_{x_3} \end{pmatrix} = \mathbf{K} \cdot \mathbf{n}_\mathrm{C} \ , \tag{3.2.1}$$

wherein the focal length f and the image coordinates of the principal point $\mathbf{o}_\mathrm{I} = (o_{u_1}, o_{u_2})^\top$ represent the camera model's parameters. The vector $\mathbf{n}_\mathrm{I}$ denotes the resulting image point and $\tilde{\square}$ marks the use of homogeneous coordinates. Furthermore, the matrix $\mathbf{K}$ is referred to as the intrinsic calibration matrix. Apart from inexact calibration parameters, the accuracy of the outlined camera model is also limited because it does not take sufficient account of all occurring phenomena. For instance, lens distortion or a skew between the image plane and the optical axis due to manufacturing tolerances are not sufficiently considered. For this reason an extended camera model is used in practice, which allows a more precise mapping, see [KB17, pp. 637–687]. For the following explanations, however, it is assumed that all effects were taken into account, and that equation 3.2.1 applies.

In general, subsequent motion planning and driving functions make use of a coordinate system that is not aligned with the camera sensor. Instead, world coordinates $\square_\mathrm{W}$ are defined that are measured in a right-handed coordinate system aligned at fixed points. For the considered test platforms, these are given by the base of the road side unit or, in the case of the test vehicle, by a road-level point below the number plate's center. Consequentially, a rotation and a translation component must be taken into account to describe the relative offset from the world to the camera coordinate systems. Based on the equation 3.2.1, the following expression can be established to map between world and image coordinates.

$$\tilde{\mathbf{n}}_\mathrm{I} = \mathbf{K} \cdot \mathbf{n}_\mathrm{C} = \mathbf{K} \cdot \, (\mathbf{R} \mid \mathbf{t}) \cdot \tilde{\mathbf{n}}_\mathrm{W} = \mathbf{P} \cdot \tilde{\mathbf{n}}_\mathrm{W} \tag{3.2.2}$$

Herein $\mathbf{R}$ is the rotation matrix and $\mathbf{t} = -\mathbf{R} \cdot \mathbf{o}_\mathrm{W}$ describes the translation of the rotated camera center in world coordinates. Together they represent the extrinsic camera parameters, and the combined matrix $\mathbf{P}$ is termed as the camera projection matrix.

Inverse perspective mapping of image points

Next, it has to be considered how the scene representations measured in image coordinates can be transformed back into the spatial world in order to be usable for driving functions. It should be noted that capturing a 3D scene in a 2D image is always associated with a loss of information. Therefore, it is impossible to determine spatial coordinates directly, but only a backprojection line can be determined. This line describes the set of all points $\mathbf{n}_\mathrm{W}$ that are mapped to a given image point $\mathbf{n}_\mathrm{I}$. The backprojection line is defined to run through the camera center $\mathbf{o}_\mathrm{W}$ of the camera system. Note that $\mathbf{P}$ can be rewritten as:

$$\mathbf{P} = \mathbf{K} \cdot \, (\mathbf{R} \mid \mathbf{t}) = (\mathbf{K} \cdot \mathbf{R} \mid \mathbf{K} \cdot \mathbf{t}) \ . \tag{3.2.3}$$

Furthermore, the translation vector $\mathbf{t}$ can be expressed with $\mathbf{p}'_4$, the fourth column vector of $\mathbf{P}$, as follows.

$$\begin{aligned} \mathbf{p}'_4 &= \mathbf{K} \cdot \mathbf{t} \\ \Leftrightarrow \quad \mathbf{t} &= (\mathbf{K})^{-1} \cdot \mathbf{p}'_4 \end{aligned} \tag{3.2.4}$$

Then, with $\mathbf{t} = -\mathbf{R} \cdot \mathbf{o}_\mathrm{W}$, it follows that the camera center can be expressed as:

$$\mathbf{o}_\mathrm{W} = -\,(\mathbf{R})^{-1} \cdot \mathbf{t} = -\,(\mathbf{R})^{-1} \cdot (\mathbf{K})^{-1} \cdot \mathbf{p}'_4 = -\,(\mathbf{K} \cdot \mathbf{R})^{-1} \cdot \mathbf{p}'_4 \ . \tag{3.2.5}$$

The vanishing point $(\overset{\infty}{\mathbf{n}}_\mathrm{W}, 0)^\top$ belonging to a given $\mathbf{n}_\mathrm{I}$ provides another point on the backprojection line. It can be expressed as:

$$\tilde{\mathbf{n}}_\mathrm{I} = \mathbf{P} \cdot \left(\overset{\infty}{\mathbf{n}}_\mathrm{W}, 0\right)^\top = (\mathbf{K} \cdot \mathbf{R}) \cdot \overset{\infty}{\mathbf{n}}_\mathrm{W} \tag{3.2.6}$$

$$\Leftrightarrow \quad \overset{\infty}{\mathbf{n}}_\mathrm{W} = (\mathbf{K} \cdot \mathbf{R})^{-1} \, \tilde{\mathbf{n}}_\mathrm{I} \ . \tag{3.2.7}$$

The set of all scene points mapped to $\mathbf{n}_\mathrm{I}$ is then defined by the backprojection line in parameter form $\tilde{\boldsymbol{\varphi}}_\mathrm{BP}$ as follows.

$$\tilde{\boldsymbol{\varphi}}_\mathrm{BP}(\nu) = \begin{pmatrix} \mathbf{o}_\mathrm{W} \\ 1 \end{pmatrix} + \nu \begin{pmatrix} \overset{\infty}{\mathbf{n}}_\mathrm{W} \\ 0 \end{pmatrix} = \begin{pmatrix} -\,(\mathbf{K} \cdot \mathbf{R})^{-1} \cdot \mathbf{p}'_4 \\ 1 \end{pmatrix} + \nu \begin{pmatrix} (\mathbf{K} \cdot \mathbf{R})^{-1} \, \tilde{\mathbf{n}}_\mathrm{I} \\ 0 \end{pmatrix} \tag{3.2.8}$$

As is obvious from these considerations, additional constraints have to be formulated to derive concrete scene coordinates.

Bird’s-eye-view through pixel-dense inverse perspective mapping

So far, a selective mapping of image points to scene coordinates was discussed. For the representation of shape-based scene elements such as the drivable road area, however, a pixel-dense reconstruction is more feasible. For this purpose, it can be specified that all points depicting the road area must lie on a common, flat ground surface in 3D space. By assuming a constant value for the road surface’s vertical position and then intersecting $\tilde{\boldsymbol{\varphi}}_{\mathrm{BP}}$ with the corresponding plane, actual 3D world coordinates can be obtained. From the previous descriptions, it is obvious that this initially leads to a sparse representation of the road due to the finite image resolution. Therefore, to obtain a dense spatial representation of the drivable road area, bilinear interpolation is used.

As shown in Figure 3.5, the obtained image gives the impression of a view of the scene from above. This is why this pixel-dense inverse perspective mapping is also referred to as bird’s-eye-view (BEV). It is apparent that this technique produces a feasible reconstruction especially of those image areas that depict the road surface. However, it can also be seen that for image areas depicting other road users, significant distortions arise due to the violated assumptions about the vertical position of the scene points. Therefore, these require a more sophisticated analysis, which will be discussed in more detail later.

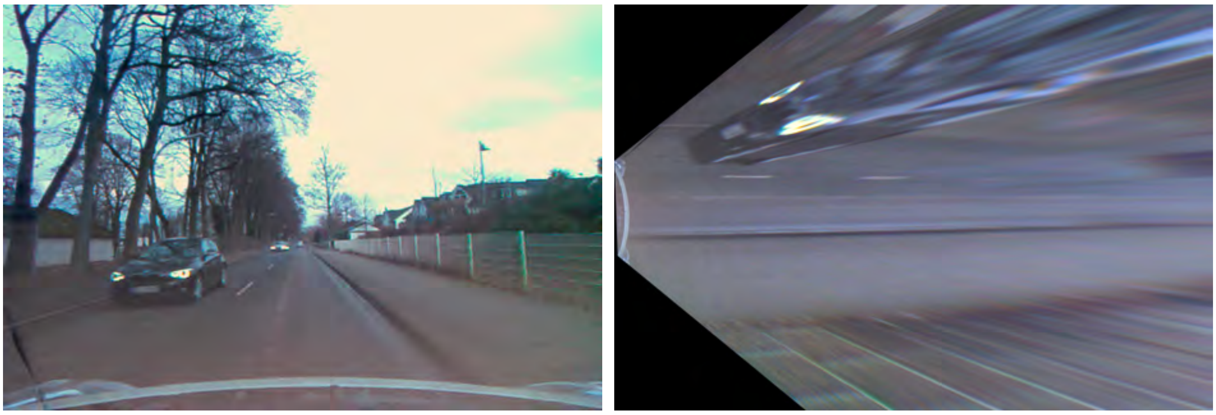


Figure 3.5: Example image of a traffic scene recorded from the test vehicle and the corresponding image after applying the BEV transformation.

4 Network Architecture for Multi-task Feature Sharing

An important design consideration for neural networks is the determination of the architecture, which is the subject of this chapter. In this context, the term architecture describes the general network structure regarding the number of neurons and how they are connected. Besides the general principles of designing deep CNN architectures, this chapter also discusses suitable strategies to reduce the experimental burden to a practical level during the design phase. Furthermore, the unique potential of multi-task architectures for traffic scene perception is discussed, and the encoder architecture used in the further course of this thesis is determined.

4.1 General design considerations

The basic composition of CNNs follows a layer-wise interconnected structure. In such a network architecture, several essential design decisions have to be made. An important design aspect is given by the network's depth, i.e. the number of successive layers N_{L}. Furthermore, the size of the individual network layers has to be determined. This includes, for example, the number of neurons N_{P} which for convolutional layers results from the size and number of filter kernels $\mathbf{W}_{\mathrm{k}}$, or the resolution of the feature maps computed by the pooling layers. Primarily, these design choices concern the network's part, which generates the intermediate representations serving as features for the output predictions.

This part of the network is dubbed as the *feature encoder* and typically accounts for the largest share of the computational cost. It is detrimental for the actual architecture design that only general estimates and empirical results are known so far in many respects. Often, these tend to have the character of broad guidelines, and their theoretical or even empirical validation is incomplete, see for example the remarks in [Sze+16] or [He+19] on general design principles. Most notably, the current state of research does not allow to formulate theoretical proof (or disprove) of the superiority of specific network architectures. Moreover, the universal approximation theorem [Cyb89; Hor+89] is often seen as an example of how, in some cases, theoretical considerations even suggested conclusions that were somewhat misleading in the light of later empirical findings [Goo+14]. Therefore, a network architecture's practical development is still commonly based on extensive experimentation, guided by monitoring the test set error [Goo+16, p. 192]. This inevitably implies a respective burden in terms of development time and trial-and-error effort. While there are approaches to automate this process [Zop+18; TL19], they have been successfully applied only to a limited number of tasks, and their computational demand remains a decisive disadvantage.

As an alternative strategy, which is able to reduce the associated experimental burden somewhat, it is suggested to align the design of the architecture heavily with known effective approaches, whose performance was already adequately investigated. This general technique is also known as *transfer learning* [Yos+14]. Due to the outlined practical benefits, a corresponding approach will be pursued in the following. Therefore, the further design procedure's main objective is the selection of a network architecture under consideration of existing empirical findings.

For this, general design considerations and strategies are reviewed first to make the underlying rationale for the final architecture choice transparent. To this end, similarities and frequently used design elements in common CNNs can provide a basis for deriving effective architectural concepts, some of which are discussed below. Furthermore, any real-world experimental system is subject to certain practical and technical restrictions, which are also examined.

Expressive features from repeated network submodules

Meaningful and expressive features that make a given task accessible to an automatic solution are at the core of any image processing or general machine learning system. This immediately raises the question of how such features can be characterized and how they can be computed. Both studies from the field of neurophysiology [IK04; Hyv+05] and studies in the field of machine learning [Lee+11; Far+13] show that meaningful visual features have an inherent hierarchical structure. Furthermore, this hierarchy can also be explained by intuition. For instance, in pictures of a certain object, it is easy to identify parts of the object that, in turn, consist of simple shapes, geometric primitives, and finally of the simplest contours such as edges or local curvatures. For example, suppose a task requires the detection of vehicles as an object class. In that case, it is useful to discover class-specific patterns (e.g. wheels or body panels) as features[5]. These considerations already suggest that models for image processing should have a hierarchical structure. In fact, this corresponds well to the outlined layered structure of CNNs, and this fact is often held responsible for their high performance in computer vision applications. When reviewing common successful CNN architectures, this hierarchical structure can also be identified at a more general level.

More precisely, it can be stated that the most performant architectures virtually always consist of repeating submodules [Mil+02; San+17], which can be understood as small, fundamental building blocks of complex architectures and are sometimes also referred to as microarchitectures. In a CNN, these submodules define patterns that include the various processing layers and a careful design of the connections between them.

[5] See for example [Lee+11] for more examples and empirical findings.

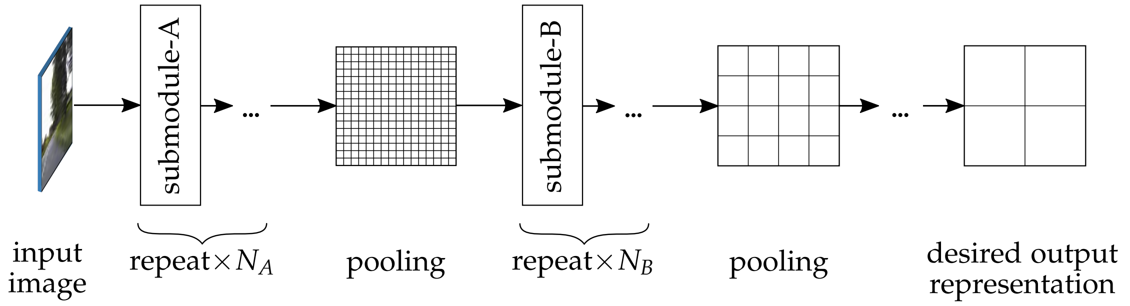


Figure 4.1: Repeated network submodules in common CNN feature encoder architectures. A repeating structure of identical microarchitectures connected in a sequence is interrupted by pooling layers for sub-sampling.

The resulting general architecture is illustrated in Figure 4.1. Herein, an architecture takes on a form in which identical repeating submodules are connected in a sequence interrupted by pooling layers that implement a gradual sub-sampling.

Notable examples of this general principle include [Sze+16; He+16; Ian+16; How+17] and [Zop+18]. Due to their empirical support, it is argued that the employed feature encoder should have a corresponding architecture to achieve an optimal performance. Conversely, approaches that do not consider this general structure, as in [Kri+12; SZ15; Tei+18], should be excluded.

Practical view on network depth and translation invariance

Practical limitations in implementing CNNs often pose a decisive factor that introduces constraints for a technically feasible network architecture.

Generally, empirical results indicate a connection between the network complexity, given by its number of successive layers N_L, and the resulting classification performance, see for example [Zop+18]. In addition to empirical results, there are also more intuitive arguments for the choice of deeper CNNs, as such a choice generally reflects the idea that a combination of many simple functions can solve a complex task. This corresponds well to the previously discussed inherent hierarchical structure of useful and expressive features in image processing. The consequential tendency towards deeper architectures gives rise to an essential technical constraint, which is given as the conflict of objectives between the computational cost of a CNN and its achieved performance. The computational cost is reflected in the computation time and the memory requirements, which are both related. The exact estimation of computational requirements depends on the individual implementation and architectural details. However, for common implementations, excluding dedicated approaches for very specialized applications, an approximately linear relationship between network depth N_L and the associated memory and computation time can be inferred [Che+16a]. For the memory requirement, this is largely determined by the feature maps [Rhu+16], the number of which increases linearly with increasing depth.

In addition, deep network architectures have more free parameters requiring particularly large datasets to avoid overfitting. Corresponding publicly available large datasets mainly exist for the problem of image recognition, since this is a relatively simple image representation with comparatively low annotation effort. Consequently, the architectures of large, high-performance networks are usually characterized by the task of image recognition, which involves maximal sub-sampling up to a scalar class label. Since the present work follows the transfer learning strategy, it is necessary to employ pre-trained image recognition networks and modify them for other perception tasks through a partial transfer of the learned parameters. Here it should be noted that the strong sub-sampling mainly stems from the translation invariance present in the task of image recognition [Zhu+19]. When adapting recognition architectures for other tasks that require a more precise localization of scene elements, this invariance must therefore be explicitly compensated.

Overall it can be stated that the network's depth is technically limited by the employed computing hardware. Furthermore, due to the availability of datasets, there are mainly image recognition architectures available for the transfer learning approach, which require an explicit adaptation to other perception tasks.

Residual skip connections and consecutive images

In deep CNNs, the training gradients are computed by applying the chain rule, which can result in very small gradient values. These can lead to a slowing or stagnation of the training process, which has been termed as the vanishing gradient problem [Hoc91]. As a possible mitigation approach, so-called residual skip connections [He+16] can bridge network layers in parallel, which breaks up the chained network structure. However, some empirical studies report inconclusive results [Sze+17b], and it is unclear whether significant benefits can be achieved specifically in a transfer learning approach, which already eases the model training due to the associated knowledge transfer.

In the context of traffic scene perception, it can also be assumed that measurements are given in the form of a video stream. Therefore, it is obvious to examine whether an explicit consideration of successive measurements in the sense of a structured prediction offers advantages over the independent processing of single images. The underlying rationale is that supplementary information from slightly different perspectives in consecutive images can be exploited to increase the perception performance.

Both strategies were examined in a preliminary study associated with the present thesis[6]. The investigation of residual skip connections was based on the work of [Sze+17b]. For the causal processing of image sequences, a recurrent network structure according to [Fay+16] and a structure with external dynamics following the works of [Tra+15] were investigated. However, the obtained results did not indicate a decisive benefit, so these strategies are not further considered in the following.

4.2 Multi-task learning and architectural implications

Multi-task learning aims at simultaneously predicting multiple output representations from an integrated model so that input data and (some of the) intermediate feature

[6] Further details are described in the master's thesis "Deep residual networks for causal semantic segmentation of traffic scene videos" written in 2017 by D. Jiang at the TU Dortmund Institute of Control Theory and Systems Engineering.

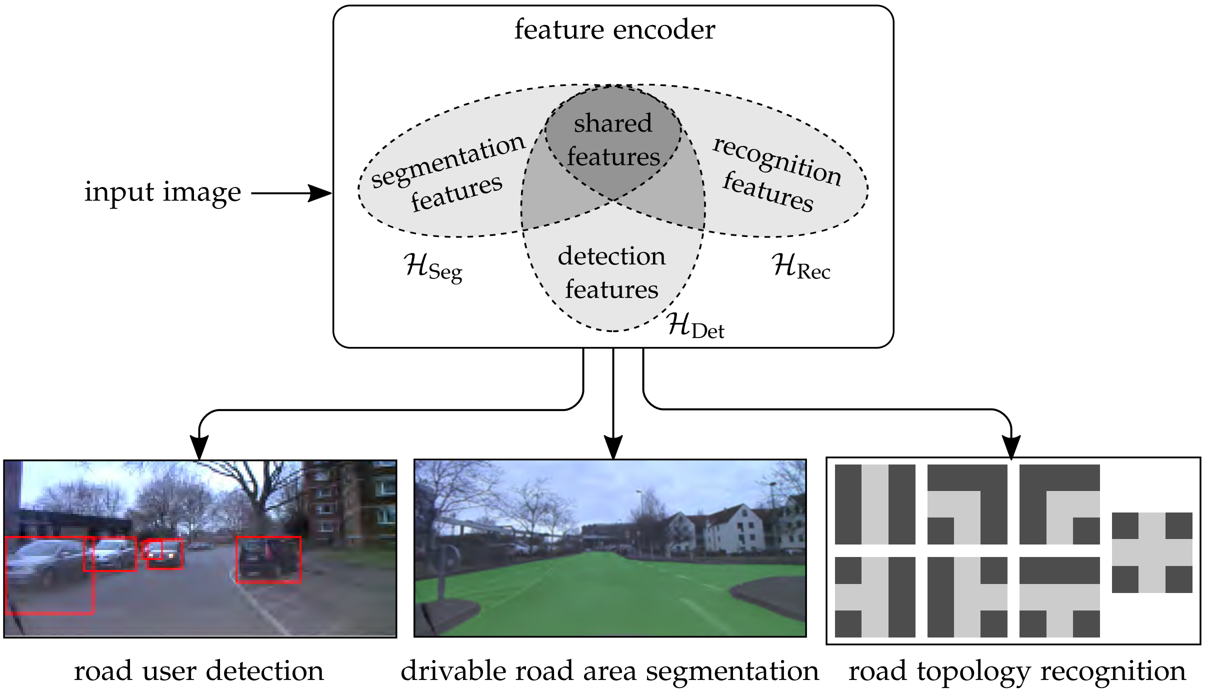

feature encoder
input image
segmentation features
shared features
recognition features
detection features
$\mathcal{H}_{\text{Seg}}$
$\mathcal{H}_{\text{Rec}}$
$\mathcal{H}_{\text{Det}}$
road user detection
drivable road area segmentation
road topology recognition

Figure 4.2: The multi-task concept, according to [Car97], illustrated for traffic scene perception. Different tasks share the same input as well as some of the intermediate features. However, they involve different target values for their respective output representations. $\mathcal{H}_{\text{Seg}}$, $\mathcal{H}_{\text{Det}}$ and $\mathcal{H}_{\text{Rec}}$ denote the feature sets relevant for the respective tasks.

representations are shared. As a motivation for this approach, it has been hypothesized that shared representations can offer some form of beneficial knowledge transfer between multiple tasks [Car93; Thr95; Car97]. Put simply, the basic assumption is that what is learned from one task is also useful for other tasks. This view resonates well with the concept of hierarchical features, that shift from simple, generic patterns to more complex and specific features. In the context of traffic scenes, the simultaneous determination of multiple complementary perception tasks offers a unique potential for a multi-task architecture. Figure 4.2 illustrates the basic concept where the tasks are given as the various considered environment representations.
Herein, the different tasks share the same input, and some of the feature maps generated by the intermediate CNN layers. However, they differ in their output representations and their associated target values, which are specific to each respective task. Following [Goo+16, pp. 237–239], the multi-task approach and its parameters can therefore be divided into two types of components:

1. The first type refers to the encoder part, which consists of the shallow to intermediate network layers. Since, as shown in Figure 4.2, it is shared across all tasks and its purpose is to generate expressive features, this part is termed the *shared feature encoder*.

2. The second type represents the components that are not shared and consist of the task specific layers located after the branch in the architecture outlined in Figure 4.2. As their function is to generate the output representations, they are termed *task specific decoders*.

Note that the overall objective of the simultaneous perception of complementary traffic scene representations already narrows the possible multi-task architectures as a matter of principle. Under this constraint, some of the general multi-task learning approaches are not applicable or not feasible for technical reasons. For instance, due to the previously discussed reasons, it is necessary to use the transfer learning approach with pre-trained models to achieve a practicable perception performance with realistic experimental and annotation effort. As a result, approaches based on fully customized multi-task model architectures, which cannot integrate previous knowledge from pre-trained models, are not suitable in this context. This applies, for example, to the approaches from [Lu+17] and [Has+17].

Furthermore, several existing strategies are based on homogeneous task structures with similar output variables, see for example [BS03] and [Arg+08]. Regarding automotive environment perception, some applications indeed exhibit homogeneous task structures, for example, when considering multi-sensor object detection [Lia+19]. However, in the case of the perception of complementary environment representations, this prerequisite is generally not met. Eventually, a differentiation can be made in terms of the extent of feature sharing, which is discussed in more detail below.

Efficient feature sharing architecture

Another design decision is the concrete form of the feature sharing within the multi-task architecture. In this context, [Rud19, pp. 48–49] distinguishes between hard and soft parameter sharing strategies, see Figure 4.3 for an illustration. Hard parameter sharing generally refers to an architecture, where all feature encoder layers are directly shared between all tasks. In fact, this strategy is most commonly applied in the context of traffic scenes, see for example [CC17; Tei+18; Rod+19; Wan+19b], and it was originally introduced by [Car93].

In contrast, in the case of soft parameter sharing, a separate model with its own model parameters is maintained for each task, and an additional regularization minimizes only the deviation of the parameters between the tasks. Corresponding work can be found, for example, in [Duo+15] and [YH17].

Generally, the main advantage of the multi-task approach is its increased efficiency, which has been discussed in the context of computational resources and prediction performance. In systems for visual scene perception that are based on embedded hardware and, at the same time, require compliance with runtime constraints, the computational advantages of multi-task learning are of particular importance.

This relates to the multi-task model's characteristic, that all task specific decoders are based on a shared set of features. Thus, compared to the use of separate models, the computational demands can be reduced if repeated computations of redundant features can be avoided. In this context, redundant features are given if some of the relevant features associated with the different tasks can be shared between two or more tasks, see also [Goo+16, pp. 237–239]. As long as it can reasonably be assumed that some redundant features exist, this can formally be expressed as:

$$\left|\mathcal{H}_{\text{Seg}} \cap \mathcal{H}_{\text{Det}} \cap \mathcal{H}_{\text{Rec}}\right| > 0 \ , \tag{4.2.1}$$

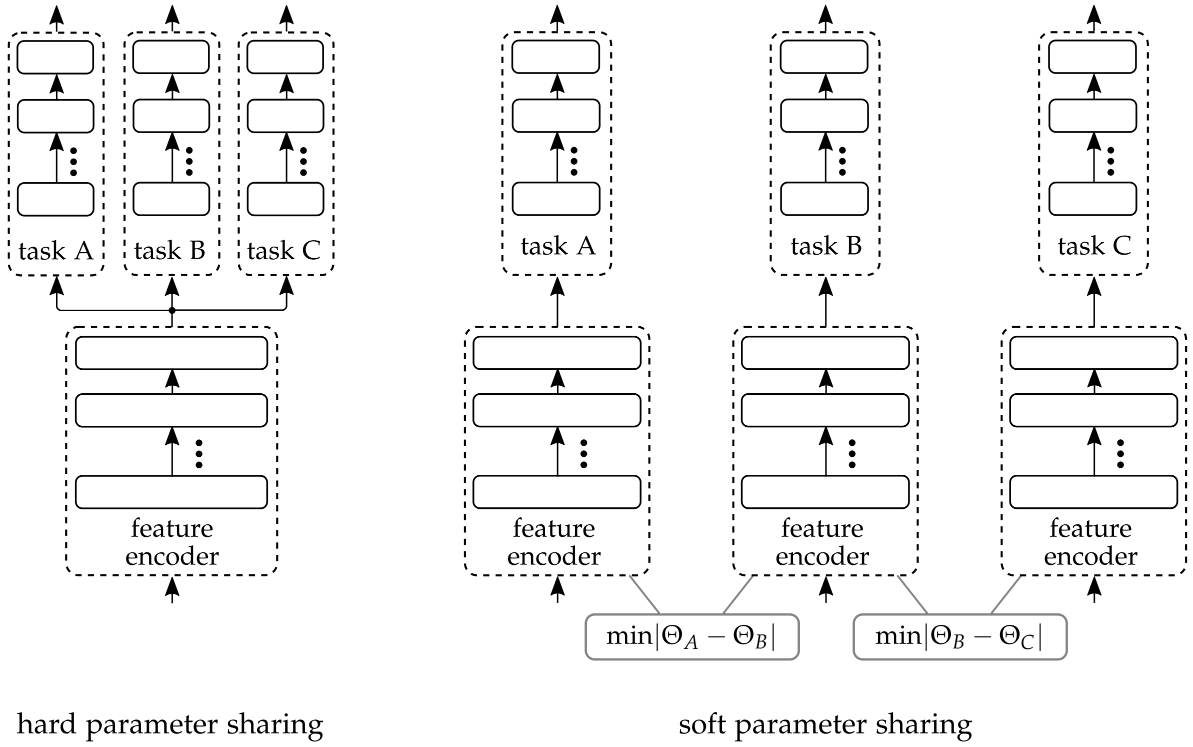


Figure 4.3: Comparison of the concepts of hard and soft parameter sharing, see also [Rud19, pp. 48–49]. Hard parameter sharing directly shares all hidden layers between the tasks, whereas soft parameter sharing minimizes the deviation of the model parameters Θ between the tasks.

wherein $\mathcal{H}_{\text{Seg}}, \mathcal{H}_{\text{Det}}$ and $\mathcal{H}_{\text{Rec}}$ denote the respective feature sets relevant for the road segmentation, vehicle detection and topology recognition tasks. Therefore, it is beneficial to use a multi-task approach solely for the increased efficiency of shared feature computations.

It is already evident from the illustration in Figure 4.3 that this computational advantage of the shared features does not apply in the case of soft parameter sharing. This is because the parameters for the tasks are similar but not identical, and thus repeated computation of the shared features cannot be omitted. For this reason, the multi-task approach used in this thesis follows the hard parameter sharing strategy shown in Figure 4.3 on the left.

Besides the computational efficiency, it is also essential to consider the effects of a multi-task approach on the resulting model performance. However, the related effects are less apparent, and a distinction can be made between potentially beneficial and detrimental influences on the model's performance. For example, it has been hypothesized that the aggregation of training examples from different tasks helps to learn the shared features better than it would be possible for a single-task model, due to the larger relevant training dataset, see [Bax00] and [Goo+16, p. 237]. Similarly, multi-task architectures inherently exhibit a preference for features that are useful for all tasks. This effect has been termed as inductive bias [Car97, p. 52], and some empirical results suggest that there is a relationship between features that are shareable between tasks and features that generalize well to unseen data [Tei+18; Kok17]. However, in contrast to the previous reasoning, it can be argued that the total model capacity in several separate single-task models can be higher than in one integrated multi-task model. Here, the capacity of a model is its ability to perform a wide variety of functions. In the discussion it should be noted that the term model capacity is not a theoretically founded and directly measurable quantity. Rather, it is an abstract concept, which is often associated with network depth N_L or the number of model parameters N_Θ, see also [Goo+16, pp. 111-112]. Since a multi-task architecture potentially reduces the available model capacity per task, this effect may adversely affect the resulting performance. Therefore, the performance effects in a multi-task approach largely depend on whether sufficient similarities exist between the different tasks, such that the model can make use of a significant amount of shared features. Moreover, it is decisive whether the total model capacity available for all different tasks becomes a limiting factor.

4.3 Comparison and choice of the feature encoder architecture

As described in the previous sections, an approach based on transfer learning and using a multi-task architecture provides an effective means of achieving a robust perception of traffic scenes with reasonable development effort and practical hardware constraints. For a multi-task architecture, as outlined in Figure 4.2, the computational burden is assumed to lie mostly with the shared feature encoder. Therefore, the choice of the feature encoder is of great importance, as it has a decisive influence in balancing the conflicting objectives of low computational requirements and high perception performance. Furthermore, a sufficient measurement accuracy should be ensured regarding the perception performance and also the spatial resolution of the resulting scene description.

Note that the employed image resolution also affects the computational requirements significantly. In practice, the image resolution specification determines the achievable spatial reconstruction accuracy, but it also needs to take the observable part of the scene into account. Consequently, aspects such as cropped traffic elements in the near field or slopes and grades in the course of the road should also be considered. Due to this, an estimation of the required resolution can be made based on well established and proven effective parameters from existing applications. For example, the system in [Gei+13] uses an image resolution of $1242 \times 375\,\mathrm{px}$. For the mobile experimental test platform presented in section 3.1, this results in a longitudinal resolution of $\approx 25\frac{\mathrm{px}}{\mathrm{m}}$ and a lateral resolution[7] of $\approx 1.5\frac{\mathrm{px}}{\mathrm{m}}$ at a distance of $x_2 = 30\,\mathrm{m}$.

Due to the comparatively slightly smaller opening angle of the employed camera system (cf. Table A.2.b in the appendix), this accuracy is slightly higher than in [Gei+13]. However, to allow for a consistent and comparable evaluation of the model architecture, an image resolution of $1242 \times 375\,\mathrm{px}$ is adopted in the following. With the discussed constraints, the choice of a possible feature encoder architecture is already significantly

[7] This estimate was determined based on the camera model in section 3.2 for positions on the ground plane.

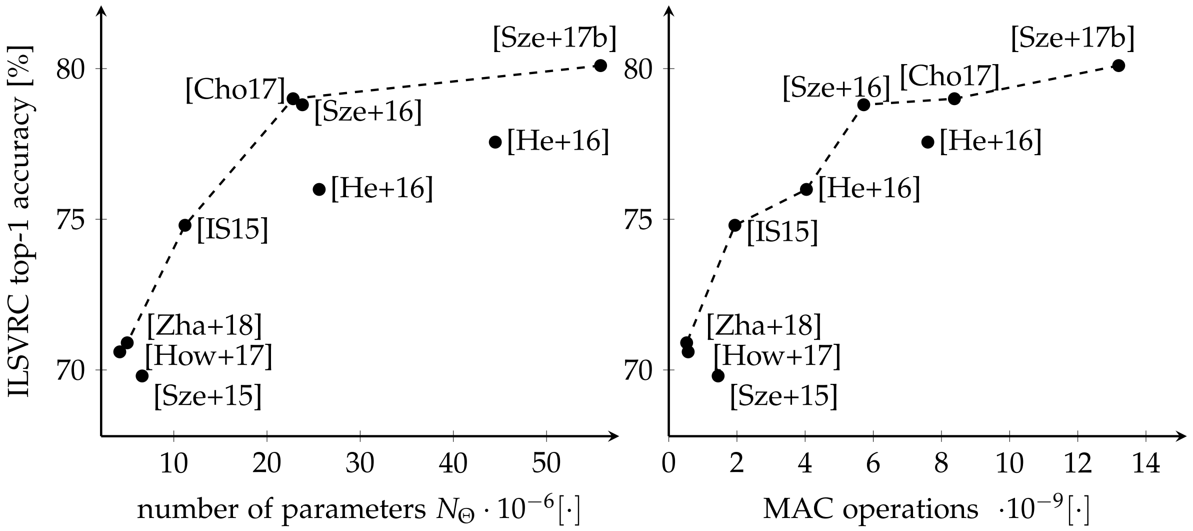


Figure 4.4: Performance versus computational resources of selected feature encoder architectures measured on the ILSVRC benchmark [Rus+15]. Left: the total number of free model parameters N_Θ. Right: the total number of multiply-accumulate (MAC) operations measured at the originally reported resolutions. The dashed line marks Pareto optimal approaches. The data stems from the corresponding publications as well as [Zop+18; Sze+17a] and [TL19].

narrowed. The performance of the model can be used as a further basis for a more comprehensive assessment. The main goal here is that the best possible performance is achieved for a given computational load. More precisely, none of the alternative architectures should require the same or a less computational resources while providing better performance. The feature encoder should, therefore, be Pareto optimal with respect to the conflict of objectives between performance and computational load.

In addition, the chosen approach should be designed according to the previously established general architecture guidelines. With this pre-selection criterion, several known architectures can be considered as alternatives for choosing the feature encoder. Instead of directly measuring the runtime and memory requirements of a particular feature encoder, the architecture selection can more easily be based on an indirect estimate by comparing characteristic parameters. For this purpose, the number of trainable model parameters N_Θ and the number of required *multiply-accumulate* (MAC) operations are considered. A corresponding comparison of potential architectures based on the benchmark of [Rus+15] for general image recognition is included in Figure 4.4. This again shows the general conflict of objectives as well as the relationship between the architecture complexity and the achieved perception performance. Furthermore, it is apparent that no linear relationship exists between the architectural complexity and the achieved performance, but instead the curve shows a flattening trend.

From the set of Pareto optimal alternatives, the architecture according to [Sze+16], which has been termed as inception-v2, can utilize the employed online hardware platform to the fullest extent while still maintaining the desired image processing resolution. Larger architectures generally exceed the memory resources of the available hardware system, while for smaller architectures, it can be assumed that the model

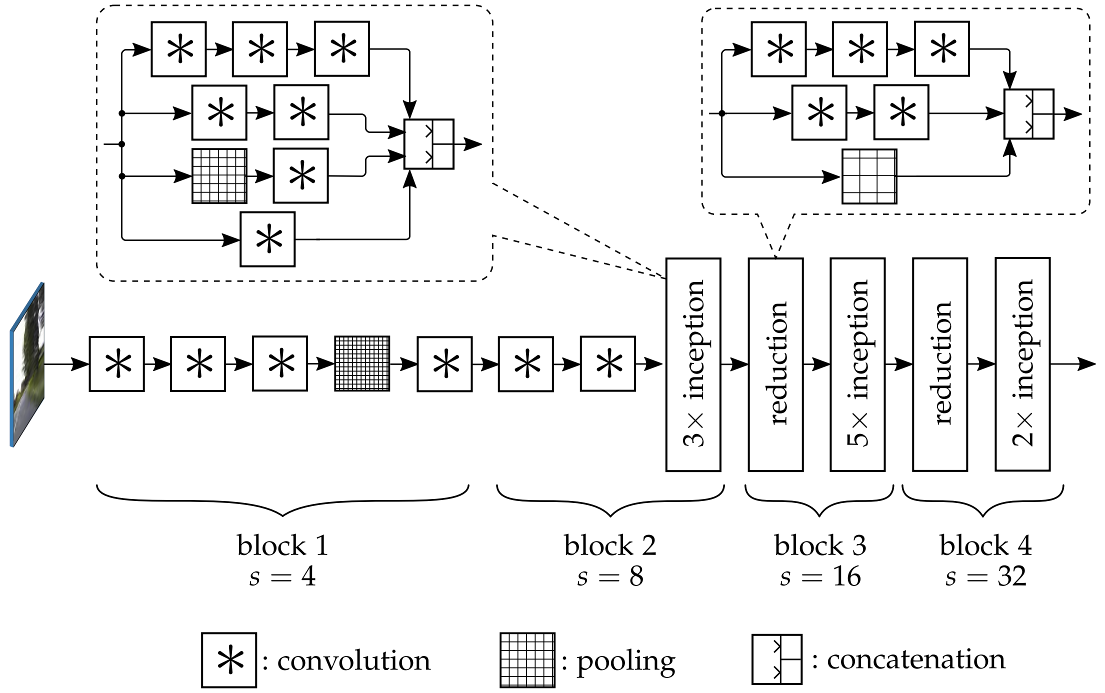


Figure 4.5: The employed feature encoder based on the inception-v2 architecture according to [Sze+16]. The inception modules reflect the idea of repeated network motifs. Separated convolutions used in the original architecture are illustrated as single blocks, and activation layers are not shown for clarity.

performance is overly limited due to unused computational resources. Besides, this architecture is located at a prominent bend in Figure 4.4, so that a favorable compromise of the conflicting goals can be assumed. Consequently, this architecture is used in the present work[8] as a basis for all considered perception tasks.

A simplified overview of the inception-v2 architecture is shown in Figure 4.5, note that the stride s here denotes the sub-sampling factor of the corresponding feature maps with respect to the input resolution. From the illustration it is evident, that this architecture makes use of several of the outlined design strategies. Thus, it follows the basic structure of successive sub-sampling and relies heavily on the so-called inception submodules, which are illustrated in the top left part of Figure 4.5. Within these inception submodules, convolutional layers with different receptive fields and a maximum pooling layer are computed in parallel. Subsequently, their output feature maps are concatenated along the channel dimension u_3. Furthermore, the gradual subsampling in deeper layers is not exclusively done by pooling, but by a combination of pooling and convolution layers (see Figure 4.5 top right).

[8] this work employs a variant of the architecture published at https://github.com/tensorflow/models/blob/master/research/slim/nets/inception_v2.py. Accessed: September 21st, 2020

5

Global Road Topology from Scene Context Recognition

The general objective addressed in this chapter is to classify the wider context of a given traffic scene. Although the definitions of what is considered relevant context information may vary depending on the intended application, they all have in common that the context information has a global character, as opposed to local information that only describes a particular region in the image. The following section discusses how the global context can be defined to categorize traffic scenes, how the context information can be used to support subsequent driving functions and how the corresponding image recognition can be implemented in a practical model. Also, an evaluation of the proposed model is presented. Parts of the considerations in this chapter have been published in the papers [Oel+14; Oel+15a; Oel+16b; Oel+17] and [Oel+18b].

5.1 Use and taxonomies of the traffic scene context

Since the concrete form of context information may vary with the intended application, a definition of what can be considered as context information is discussed first. Concerning traffic scenes, [Fas+95, p. 44–45] defines the subjective situational context as the directly perceivable part of the traffic-related influences from the driver's perspective. Similarly, [CK00] define that "context is the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user". [Mat+15] notes that for the purpose of automotive engineering, the user is given by the ego vehicle.

To further concretize the examined context information, it is helpful to take a closer look at the possible objectives associated with its perception. Here, among others, the following objectives are identified:

- Situational adjustment of the executed driving strategy
- Additional supplementary input features for other perception tasks
- Cross-checks and plausibility tests to support a safe function

The most important objective results from the need for adaptive, situational adjustments of subsequent driving strategies build on top of an environment perception. With regard to ADAS it can be observed that on the one hand, the number of available functions has strongly increased over time, while on the other hand single, specialized ADAS functions cannot support the driver permanently during the whole journey. Instead, the activation of ADAS depends on the context of the respective scene [Kas+11; Wei+13]. In contrast, modern approaches in automated driving often pursue an integrated system architecture, in which a strict differentiation into individual ADAS functions is no longer feasible. Nevertheless, it can be assumed that the traffic scene's context has a significant influence on the execution of the driving task, see also [Sch12; Fra+15].

Besides, global context information also plays an important role in the task of environment perception itself. Firstly, this can be verified by an explicit examination of the occurrence frequencies of certain objects, which can vary depending on the global scene type, see for example [Hoi+08; Nie14]. Secondly, empirical observations on models with integrated global context information also support this argument, where the incorporation of context information can improve the generalization capabilities through an additional inductive bias [Liu+16b; Sch16; Tei+18].

Moreover, another objective concerns the often discussed problem of functional safety of automated driving systems [Win15; Lüt+18]. Here, global context information provides an opportunity to implement additional cross-checks and plausibility tests, which are particularly useful for dealing with rare situations and corner cases. Especially in this context, the necessity of context perception with online sensors becomes apparent, for example with regard to the safety requirement of self-contained operation or the possibility of medium-term changes in global context information, e.g. when considering construction sites.

Traffic scene related global context taxonomies

Early on, [Fas+95] demonstrated that the driving task is affected by the discrete class of the given traffic scene. The same assumption about the discrete nature of context information was also adopted in several later studies, see for example [Wei+13; Sch16; Sch+18].

For further differentiation, it can therefore be assumed, that a discrete taxonomy with clear semantic meaning can represent global information relevant to the driving task and perceivable by cameras. A model for camera-based context perception, therefore, generally resembles a global image recognition classifier. Some examples of corresponding traffic scene context attributes include the lane count [Sch+18], the road type [Kas+09; Tei+18; Sch16], weather and visibility information [Wei+13], and the road topology [Ess+09], among others. These examples suggest that the objectives of the given application determine the context taxonomy.

However, it can be stated that context information is particularly useful in complex scenes and that among the discussed examples, road topology information generally offers a very direct benefit for subsequent driving functions. For example, this is evident from the discussions in [Sti+15] on the intersection perception module employed

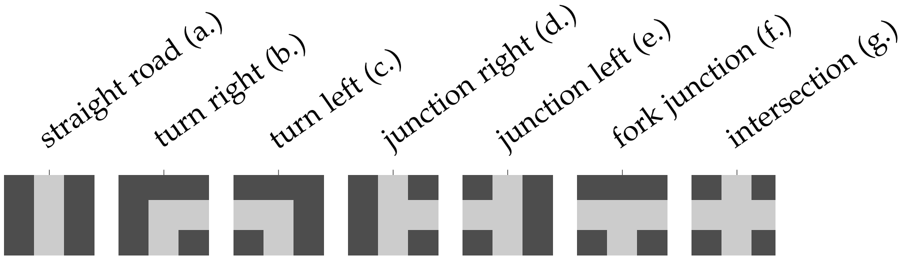


Figure 5.1: The traffic scene taxonomy used for global context recognition throughout the present thesis. Seven distinct classes describe the road topology in urban, inner-city traffic scenes.

in the Bertha Benz drive [Zie+14]. As a further example, the work in [Sch12] demonstrates that the set of feasible driving maneuvers depends on the road topology, i.e. the presence of junctions or intersections. Therefore the following investigations consider a taxonomy that describes the road topology in urban, inner-city traffic scenes. Due to the diversity of the occurring traffic elements and the high variability of the visual appearance of inner-city scenes, context perception here is particularly meaningful. The corresponding traffic scene taxonomy is presented in Figure 5.1. Here, roads without junctions are distinguished from those with one, two, or three junctions, and further the course of the road, e.g. to the left or right, is taken into account. However, it shall be noted that the general approach of image recognition is also directly applicable to most other taxonomies, except for hierarchical context attributes [Bin+09; Wu+19a] which require dedicated approaches.

5.2 Recognition decoder and architecture integration

In the employed feature encoder architecture, the resolution of the computed intermediate feature maps decreases with increasing depth of the network layer. Furthermore, since the problem of road topology recognition requires the prediction of scalar context class confidences, no spatially resolved feature maps are required for this task. Therefore, the last (deepest) layer of the feature encoder is well-suited to derive the scene context recognition. A corresponding illustration of this decoder architecture, which again omits the nonlinear activation functions for clarity, is provided in Figure 5.2.
In order to predict scalar class labels from the low-resolution features, the dimensions must be further reduced. For this purpose, confidence scores are generated by applying a fully connected layer that eventually discards 2D resolution information. Since the original approach of the inception-v2 architecture also investigates the problem of image recognition, the findings reported in [Sze+16] were taken into account for the design of the road topology recognition decoder. This concerns, in particular, the number of fully connected layers that are used. Although approaches that built recognition decoders from a sequence of multiple fully connected layers have also been proposed [Kri+12; SZ15], image recognition with a single fully connected layer reportedly yields just as good performance [Sze+15; Sze+16]. Due to this and also for its lower computational complexity, this design is adopted in the following.

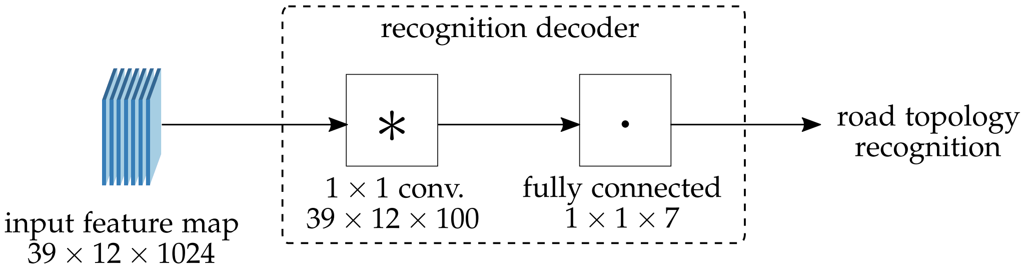


Figure 5.2: Schematic overview of the recognition decoder. A fully connected layer computes the classification confidence scores for the seven distinguished road topologies. It is preceded by a 1×1 convolutional bottleneck, which compresses the feature channel dimension u_3 to the fixed value of $u_3 = 100$.

As a fully connected layer resembles the inner matrix product, which is computationally expensive and adds a large number of free parameters to the model, an additional 1×1 convolutional layer is first used as a preceding step. This convolutional layer reduces the feature channel dimension u_3. Therefore, it acts as an additional bottleneck to augment the computational burden of the output classification stage. If $\mathbf{H}_l$ denotes the recognition decoder's input feature map, the corresponding prediction of the road topology is thus obtained as follows.

$$\mathbf{h}_{N_\mathrm{L}} = \varphi_\mathrm{a}\left(\mathrm{vec}\left(\varphi_\mathrm{a}\left(\mathbf{H}_l * \mathbf{W}_{\mathrm{k},l+1} + \mathbf{B}_{l+1}\right)\right) \cdot \mathbf{W}_{N_\mathrm{L}} + \mathbf{b}_{N_\mathrm{L}}\right) \tag{5.2.1}$$

Here, $\mathbf{W}_{\mathrm{k},l+1}$ and $\mathbf{B}_{l+1}$ are the kernel and bias parameters of the convolutional bottleneck layer, $\mathbf{W}_{N_\mathrm{L}}$ and $\mathbf{b}_{N_\mathrm{L}}$ denote the weight and bias parameters of the fully connected output classifiers, and φ_a is the activation function. Furthermore, $\mathbf{h}_{N_\mathrm{L}}$ contains the class confidence scores in one-hot notation and without softmax normalization.

End-to-end recognition network architecture

Due to the seven different considered road topologies, the dimension of $\mathbf{h}_{N_\mathrm{L}}$ is given as $1 \times 1 \times 7$. Furthermore, the bottleneck layer is chosen such that the channel dimension of its output feature map takes a value of $u_3 = 100$. This value was found to be convenient in preliminary studies [Oel+17] as it results in a good compromise between runtime and classification performance, as the follow-up evaluation will show.

With these arrangements, it is possible to define an end-to-end CNN architecture for implementing the road topology recognition. Here, the term end-to-end refers to an architecture with one continuous processing path from the recorded input image to the final classification result. Consequently, all model parameters of this architecture can be optimized in one single training process.

Figure 5.3 depicts an overview of the resulting architecture with the inclusion of the recognition decoder. For the sake of clarity, the illustration of the encoder in the left part of the Figure condenses subparts of equal spatial resolution into single blocks. This is because the architecture can be represented mainly by the feature map connections, while the details of the feature encoder are generally interchangeable. Note that the

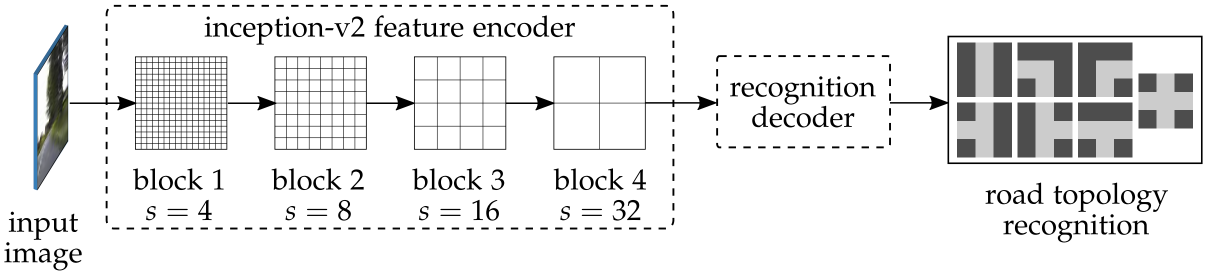


Figure 5.3: Overview of the end-to-end trainable recognition CNN architecture. The underlying inception-v2 feature encoder is condensed into subparts of equal stride s (equal spatial resolution). The deepest feature map with $s = 32$ of the encoder provides the input to the recognition decoder.

stride s denotes the sub-sampling factor of the corresponding feature maps with respect to the input resolution.

5.3 Road-topology recognition experiments

The following sections provide an examination of the presented approach's performance in the task of road topology recognition. For this, first, the description of a corresponding dataset is given. Subsequently, the corresponding experimental evaluation is carried out.

Utilized dataset and labeling strategy

At the time of conducting the following experiments, no specific dataset with the necessary annotations for road topology recognition in urban environments was publicly available. However, since this type of traffic scene was already investigated for other perception tasks, a corresponding dataset can be supplemented by annotating the road topology. For this purpose, a subset of the Cityscapes dataset [Cor+16] is used in conjunction with the annotation available from [Oel+17]. The new dataset thus obtained is used for the subsequent experimental investigations. The contained images result from a front facing camera, such that the setup roughly corresponds to the test vehicle described in section 3.1.

Furthermore, all images were recorded during daytime and in clear weather, and the considered images stem from multiple German cities, whose share of the dataset is illustrated in the appendix in Figure A.1. Moreover, the diversity of the inner-city traffic scenes contained in the dataset has to be highlighted. For example, the roads are partly demarcated by buildings and partly by landscaped areas. Furthermore, specific traffic elements such as pedestrian crossings, parking lanes, or traffic lights are present in some but not all of the images. Altogether 1599 images were supplemented by an annotation of the road topology, which were further divided into $N_{\text{train}} = 1199$ training and $N_{\text{test}} = 400$ test images. The original dataset of [Cor+16] was furthermore cropped and scaled to allow for a consistent image resolution. [BT12] found that the intuitive visual understanding of human annotators often is subjective and ambiguous when

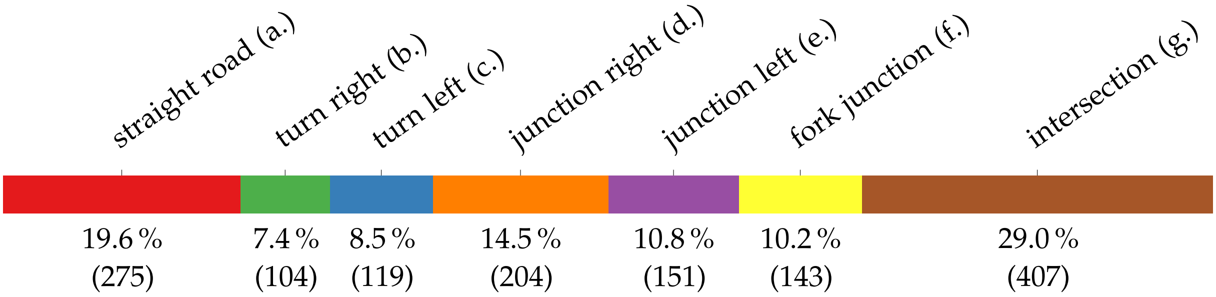


Figure 5.4: Statistics of the road topology recognition dataset with respect to the distinguished topology classes. Notably, the occurrence frequencies are not equally balanced.

semantically meaningful terms are assigned to an image. Therefore, creating feasible image annotations firstly requires the definition of a clear, objective labeling protocol. In order to achieve a reproducible and objective definition of class categories, the respective annotations were thus determined according to fixed criteria. To this end, using the original resolution of [Cor+16], the road topology predominantly visible in the lower third of the image was annotated. In the real spatial scene, this corresponds to a distance of about $\approx 20\,\mathrm{m}$ and also accurately reflects the subjective impression when viewing the scene. Besides, similar strategies for defining the topology categories can also be found in other works [Ess+09].

The number of recorded examples for each road topology class is shown in Figure 5.4. From this, it can be deduced that the dataset is unbalanced with respect to the occurrence of the various considered road topologies. For instance, most of the images depict intersection scenes. The statistics also reveal that left and right turns and junctions are included with slightly different occurrence frequencies.

Therefore, the loss function is adjusted to weight training samples according to their class frequency in the dataset, accounting for the unbalanced class occurrences. However, [Shu+16] observed a classification performance deterioration if class weights are merely adjusted according to their inverse frequencies. Thus they propose the 85%-15%-rule introduced in [Mos+15] to define class weights. Following [Shu+16] the weight η_κ of class κ is defined as:

$$\eta_\kappa = 2^{\left\lceil \log_{10}\left(\frac{\chi}{\xi_\kappa}\right)\right\rceil} \quad . \tag{5.3.1}$$

Herein χ is the count of the most frequent classes that account for at least 85% of the dataset (thus the 85%-15%-rule), and ξ_κ denotes the class occurrence frequencies as given in Figure 5.4. To account for these weighting factors during model training, the negative log likelihood loss function according to equation 2.3.12 needs to be adequately modified. The modified loss function for the task of road topology recognition is thus obtained as follows:

$$L_{\mathrm{Rec}}\left(\mathbf{y}_{N_{\mathrm{L}},j}, \mathbf{h}_{N_{\mathrm{L}},j}\right) = -\sum_{\mathbf{u}} \eta_\kappa \cdot \ln\left(y_{N_{\mathrm{L}},j}\left(\mathbf{u}\right) \cdot \varphi_{\mathrm{s}}\left(h_{N_{\mathrm{L}},j}\left(\mathbf{u}\right)\right)\right) \quad , \tag{5.3.2}$$

where again $\mathbf{u} = (u_1, u_2, u_3)^\top$ and $\kappa = u_3$ holds, the vector $\mathbf{y}_{N_\mathrm{L},j}$ denotes the road topology ground truth, $\mathbf{h}_{N_\mathrm{L},j}$ the output feature map, $y_{N_\mathrm{L},j}$ and $h_{N_\mathrm{L},j}$ their respective elements, and φ_s denotes the softmax function.

Classification performance evaluation

For an experimental examination, the model parameters of the network architecture shown in Figure 5.3 are trained on the dataset described in the previous section. Noteworthy, common data augmentation techniques, such as performing a mirror or crop operation on the original images, may alter the road topology depicted in the scene. Therefore, no data augmentation techniques were used for the following experiments. All model parameters belonging to the original inception-v2 architecture are initialized from a model pre-trained on the ILSVRC dataset [Rus+15] of 2012 and published alongside the works of [Aba+16] to make use of the transfer learning technique. The remaining model parameters of the recognition task decoder were initialized from a uniform distribution according to the method of [GB10]. Furthermore, the RMSprop method [TH12] with momentum term is used to determine the model parameters. Additionally, an exponentially decaying learning rate γ is employed to scale the parameter updates according to the following formula.

$$\gamma_i = \gamma_0 \cdot (\lambda_\mathrm{LR})^{i/N_i} \tag{5.3.3}$$

Following the works of [Aba+16], and in reference to equation 2.3.14, the initial learning rate is chosen to $\gamma_0 = 0.004$, the learning rate decay factor is chosen to $\lambda_\mathrm{LR} = 0.95$, the moving average decay factor is chosen to $\lambda_\mathrm{MA} = 0.9$, and the momentum weight factor is set to $\zeta = 0.9$. The batch size is chosen to $N_\mathrm{batch} = 12$, and the recognition CNN is trained for a total number of $N_i = 100\,000$ iterations.

The resulting model is evaluated with respect to the test partition of the road topology dataset to assess the achieved performance. Since the recognition of the road topology is a multi-class problem, the use of performance measures for binary classification is not feasible without further adaptations. To this end, the obtained classifications are distinguished into the cases true positive (TP), true negative (TN), false positive (FP), and false negative (FN), analogous to binary classification tasks. The different cases are evaluated first combined for all samples, termed as the micro-strategy and denoted as $\square_\mu$. Secondly, each of the different cases is counted separately for all investigated classes, dubbed as the macro-strategy $\square_\mathrm{M}$ respectively. In the case of the macro-strategy, another distinction must be made between the one-vs.-all and the one-vs.-one setting [Bis06, pp. 182–183].

Furthermore, the works in [HT01] and [DG06] argue that multi-class classifiers generally result in a disproportionate number of TN samples. This especially applies to unbalanced datasets such as the one considered here, see Figure 5.4. Due to this effect, the conclusiveness of performance measures that evaluate these TN cases should be considered as impaired. Therefore, the following evaluation is initially focused on the measures *pre* (precision) and *rec* (recall), as they do not include TN samples. For the derivation of a single comprehensive performance measure, the $F1$ score is reported as well. It is defined as the harmonic mean of *pre* and *rec*. To asses the overall performance of the obtained model, the average values of these performance measures are considered. Here, the aforementioned cases of the micro and macro average are examined. For the macro average, the values resulting from the one-vs.-all setting are reported. The corresponding formulas for all three performance measures pre_{M}, rec_{M}, and $F1_{\mathrm{M}}$ are given as follows.

$$
\begin{aligned}
pre_{\mathrm{M}} = \frac{1}{N_\kappa}\sum_{\kappa=1}^{N_\kappa}\left(\frac{TP_\kappa}{TP_\kappa + FP_\kappa}\right) \ , \ rec_{\mathrm{M}} = \frac{1}{N_\kappa}\sum_{\kappa=1}^{N_\kappa}\left(\frac{TP_\kappa}{TP_\kappa + FN_\kappa}\right) \ , \\
F1_{\mathrm{M}} = \frac{1}{N_\kappa}\sum_{\kappa=1}^{N_\kappa} 2\frac{\left(\frac{TP_\kappa}{TP_\kappa+FP_\kappa}\right)\cdot\left(\frac{TP_\kappa}{TP_\kappa+FN_\kappa}\right)}{\left(\frac{TP_\kappa}{TP_\kappa+FP_\kappa}\right)+\left(\frac{TP_\kappa}{TP_\kappa+FN_\kappa}\right)}
\end{aligned}
\tag{5.3.4}
$$

Herein, $N_\kappa = 7$ denotes the number of all considered classes, κ denotes the road topology class encoded in the feature channel dimension u_3 of $\mathbf{h}_{N_{\mathrm{L}}}$ and TP_κ, FP_κ and FN_κ the respective cases for the class κ. Note, that the definition of $F1_{\mathrm{M}}$ is not consistent in the literature, as a minority of publications uses a different formula that calculates $F1_{\mathrm{M}}$ based on the averaged values of pre_{M} and rec_{M}, see [SL09].

Also, observe that the total number of FP samples equals the total number of FN samples. Therefore, for the micro-average pre_μ, rec_μ, and $F1_\mu$ are identical, and only $F1_\mu$ is considered in the following. It can be computed as follows.

$$
F1_\mu = 2\frac{pre_\mu \cdot rec_\mu}{pre_\mu + rec_\mu} = \frac{\sum_{\kappa=1}^{N_\kappa} TP_\kappa}{\sum_{\kappa=1}^{N_\kappa}\left(TP_\kappa + FP_\kappa\right)} = \frac{\sum_{\kappa=1}^{N_\kappa} TP_\kappa}{\sum_{\kappa=1}^{N_\kappa}\left(TP_\kappa + FN_\kappa\right)}
\tag{5.3.5}
$$

A further performance measure results from the observation that often not all misclassifications are equally severe in the considered application context. For example, the correct classification of the number of road junctions can be of greater importance for landmark-based global planning and localization algorithms than the correct road curvature classification. To account for this, a decision threshold τ can be applied to the predicted class probabilities so that more severe misclassifications can be avoided at the expense of a higher number of less severe classification mistakes. By sampling different τ, *pre* and *rec* can be plotted against each other, so that the model's ability to balance between the different error cases can be evaluated.

To determine a *pre-rec* curve in a multi-class setup, a feasible strategy is to derive the *pre-rec* curve on the basis of a one-hot notation of the class probabilities $\varphi_{\mathrm{s}}\left(\mathbf{h}_{N_{\mathrm{L}}}\right)$ analogous to the binary case. In this way, the fact that topology recognition is indeed a multi-class problem is effectively ignored, so that all calculations are performed in a class-agnostic manner. In addition to assessing the actual *pre-rec* curve, the area under this curve is regarded as a further performance measure and is referred to as mAP_μ (mean average precision). Following [Eve+10], *pre* is interpolated for each point on the *rec* axis by determining the maximum *pre* with a higher *rec*:

$$
pre_{\mathrm{interp}}\left(rec\right) = \max_{rec^\diamond \geq rec} pre\left(rec^\diamond\right) \ ,
\tag{5.3.6}
$$

where $rec^\diamond$ is an auxiliary variable to cover the range of higher *rec* levels. Then, again following [Eve+10], mAP_μ is determined using 11 equidistantly distributed points according to the formula:

Table 5.1: Multi-class *pre*, *rec*, $F1$ and mAP measures for the examined road topology recognition problem. The first row reports the results obtained from the proposed approach. Additionally, alternative approaches are included to provide a comparative basis. The runtimes were measured on the hardware system outlined in Table A.2.d

model	# classes	micro-average		macro-average			runtime
		mAP_{μ}	$F1_{\mu}$	pre_{M}	rec_{M}	$F1_{\mathrm{M}}$	
this work	7	**74.97 %**	**70.61 %**	**69.02 %**	**66.85 %**	**67.39 %**	22.4 ms
[Oel+17]	7	71.39 %	68.02 %	67.30 %	64.22 %	64.82 %	20.3 ms
[Ess+09]	8	n/a	n/a	45.00 %	n/a	n/a	n/a

$$mAP_{\mu} = \frac{1}{11} \sum_{rec \in\, 0,0.1,\ldots,1} pre_{\mathrm{interp}}\left(rec\right) \quad . \tag{5.3.7}$$

With these definitions, the performance measures reported in Table 5.1 result. Note that Table 5.1 also includes alternative approaches for a further assessment of the results. To this end, an additional road topology recognition CNN with a less complex feature encoder is trained and evaluated on the same dataset for comparison. Due to the similar architecture and based on the investigations in [Oel+17], the inception-v1 encoder according to [Sze+15] is used for this purpose. Thus, the recognition decoder remains unchanged as in the Figures 5.2 and 5.3, and only the encoder part is modified. As another comparative basis, the work of [Ess+09] is considered due to the similarly formulated problem. It examines a related problem of road topology recognition with eight distinct classes, which are the seven classes shown in Figure 5.1 and an eighth class for roundabout scenes.

Table 5.1 reveals that the proposed approach consistently outperforms the other results. In the case of the work from [Ess+09], this is presumably because the approach there is based on manually designed image features instead of learned or optimized features. Therefore, compared to the recognition CNN investigated in the present work, it can meanwhile be regarded as obsolete. The relative difference in performance compared to a similar CNN based on [Oel+17] is indeed significantly smaller. The reason for the remaining margin is most likely the relatively higher model capacity, which offers an advantage over more compact CNN architectures due to more expressive and robust features.

Furthermore, the obtained results reveal slightly higher values for pre_{M} compared to rec_{M}. However, an overall balanced behavior of the model is apparent. Compared with other image recognition problems, such as the one described in [Rus+15], the achieved model performance may initially appear beneath expectations. However, it should be noted that the investigated traffic scenes exhibit a significantly higher degree of visual similarity and the examined classes differ only in fewer and smaller but nevertheless decisive scene elements. Also, it has to be mentioned that the used dataset can be described as comparatively small, which may act as a limiting factor. For the sake of clarity, the further evaluation is again based on one combined *pre-rec* curve, as explained above. The corresponding curve is shown in Figure 5.5. From this, it can

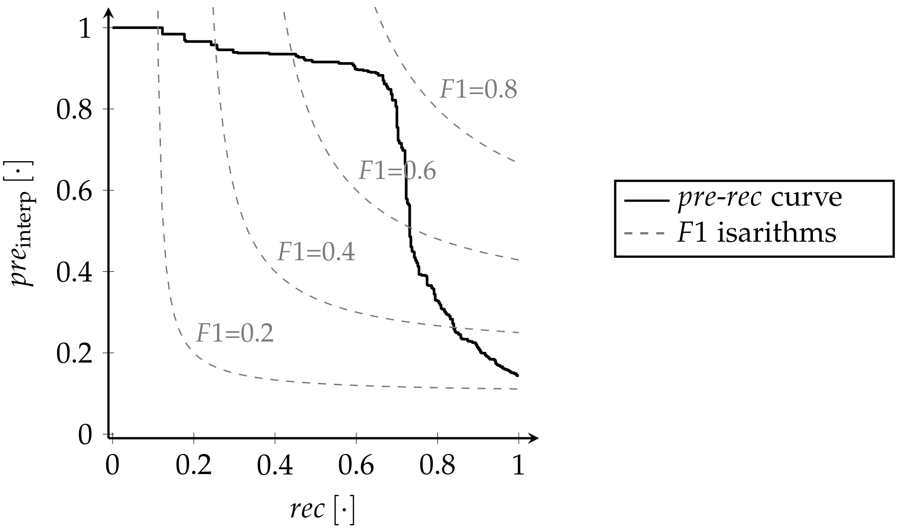


Figure 5.5: Interpolated *pre-rec* curve of the proposed road topology recognition CNN. Based on a one-hot notation, the curve was calculated per-sample in a class agnostic manner.

be seen that the shape of the curve generally approaches a step pattern. Furthermore, the curve indicates that the model does indeed provide some flexibility for balancing misclassifications. This is evident from the fact that the curve does not form a strict step (e.g. the maximum $F1$ score is not reached in a straight line), and a significant part of the curve lies in the area of high $F1$ values.

Moreover, the end point of the curve in the low *pre* range reveals that this evaluation is also subject to a certain bias due to the high number of TN cases. This is because the curve is determined from class-agnostic one-hot vectors, which produce a high share of negative ground truths. Note, that the full presentation of all *pre-rec* curves for the considered classes, determined in the one-vs.-all setting can be found in the appendix under section A.3.

In addition to the combined evaluations, the analysis is also supplemented by a consideration of the individual classes. For this purpose, Table 5.2 presents the complete confusion matrix for all investigated road topologies. In conjunction with Figure 5.4, it becomes evident that the classification accuracy is slightly better for classes with a comparatively large number of samples. Thus, the best results are obtained for the classes straight road and intersection, which are also the most frequent classes in the used dataset. In contrast, the least frequent class turn right yields the lowest classification accuracy.

These findings indicate again, that the overall size of the given dataset is a limiting factor that negatively affects the resulting classification accuracy. Furthermore, it is also evident that erroneous classifications are particularly often assigned to the two most frequent road topology categories. It can therefore be assumed, that a significant share of misclassifications results from an inherent bias of the obtained model towards frequent classes. The second highest percentage of misclassifications concerns the case of fork junctions, which are confused with left turns. This may be explained by the high degree of visual similarity between these two classes. It should be noted, however, that the spread of misclassifications between the categories is not overly large and thus

Table 5.2: Per-class confusion matrix of the proposed road topology recognition CNN. The numbers indicate the percentage of the corresponding classifications with respect to the total number of samples of a class. Furthermore, the numbers in parentheses indicate the absolute number of classifications respectively.

		prediction						
	[%]	a.	b.	c.	d.	e.	f.	g.
ground truth	a.	78.4 (58)	0.0 (0)	1.4 (1)	2.7 (2)	1.4 (1)	5.4 (4)	10.8 (8)
	b.	12.0 (3)	60.0 (15)	0.0 (0)	0.0 (0)	8.0 (2)	8.0 (2)	12.0 (3)
	c.	3.6 (1)	0.0 (0)	60.7 (17)	3.6 (1)	0.0 (0)	10.7 (3)	21.4 (6)
	d.	5.7 (3)	1.9 (1)	5.7 (3)	62.3 (33)	1.9 (1)	7.5 (4)	15.1 (8)
	e.	10.5 (4)	2.6 (1)	5.3 (2)	2.6 (1)	65.8 (25)	5.3 (2)	7.9 (3)
	f.	0.0 (0)	9.7 (3)	16.1 (5)	9.7 (3)	0.0 (0)	61.3 (19)	3.2 (1)
	g.	4.0 (4)	0.0 (0)	5.0 (5)	1.0 (1)	3.0 (3)	7.9 (8)	79.2 (80)

does not unduly affect the viability of the obtained road topology predictions.

As a qualitative assessment, individual example images and their corresponding classifications will be examined in the following. For this purpose, Figure 5.6 illustrates a selection of images of the test partition of the dataset with an overlay of the predicted classifications. These first of all confirm once again the great diversity of the recorded inner city-scenes. For instance, in some pictures, there are no lane markings, or the road boundary is characterized by parking strips with stationary vehicles in some but not in all images. In general, the selected examples demonstrate a qualitatively accurate prediction of the predominant road topology. As an exception, the center right example image contains a typical misclassification, in which a right turn was wrongly assigned to a straight road. This is presumably due to an over-representation of the straight road class in the training dataset, which further confirms the earlier conclusions. Overall, it is apparent that an adequate characterization of the road topology as a global context attribute of the traffic scene can be determined.

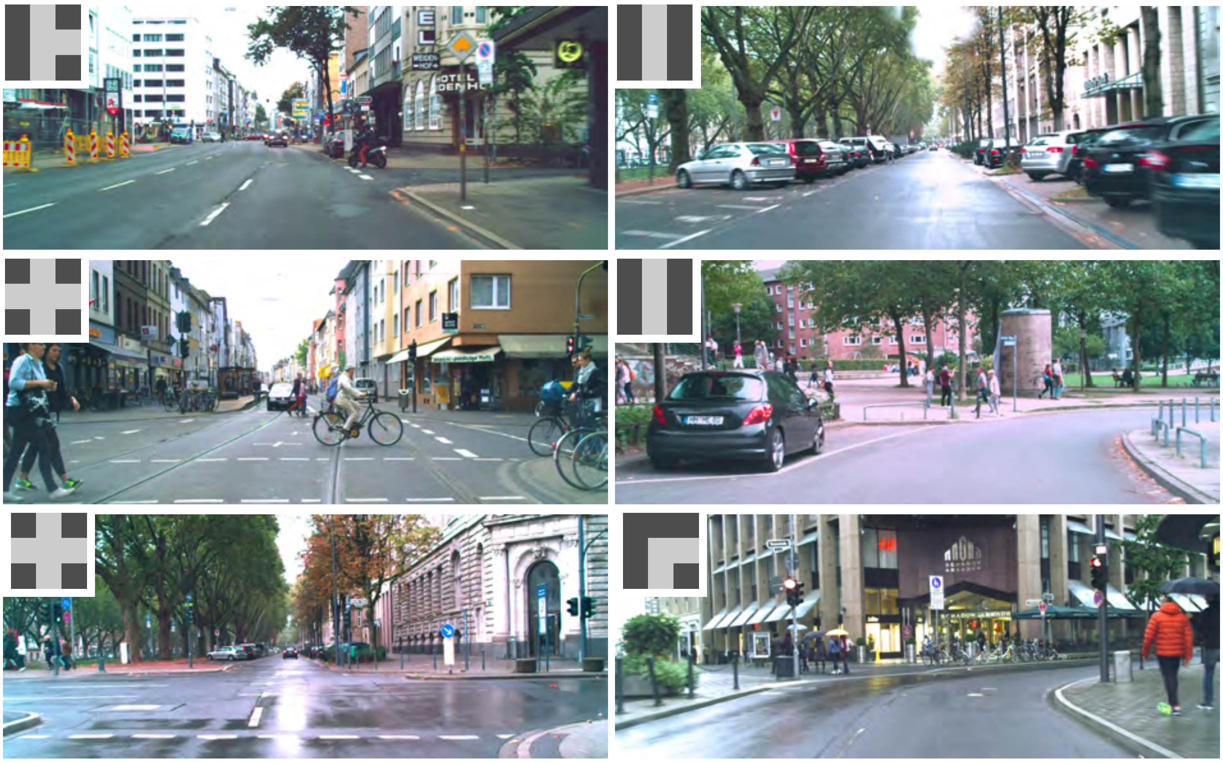


Figure 5.6: Example images from the test partition of the dataset with overlayed road topology predictions. From top left to bottom right, the corresponding ground truth classes are: junction right, straight road, intersection, turn right, intersection and turn right.

6 Drivable Road Area from Semantic Image Segmentation

Semantic segmentation identifies regions in the image that correspond to specific object categories. More precisely, it consists of assigning a class to each pixel corresponding to its surrounding object or area, so that segmentation can be understood as a pixel-wise classification. Thus, this representation is capable of capturing the exact contours of objects or scene elements. Generally, segmentation has the effect that scene elements that have the same class and are close to each other are grouped in one common region. Due to this, it is most suitable for the representation of traffic elements that cannot be distinguished solely by their position and size, but instead, the exact shape defines the essential information, and instance separation is negligible. Thus it naturally lends itself to the perception of the drivable road area, which indeed is one of the most relevant applications of image segmentation in the automotive field.

Figure 6.1 shows an example image recorded in the vicinity of TU Dortmund University. The perception of the illustrated road area segmentation is the subject of the present chapter. For this, the applicable methods are discussed, followed by the selection and design of the specific segmentation architecture used in this thesis. Furthermore, an evaluation of the obtained results is conducted. Some excerpts from this chapter have been published in the papers [Oel+15b; Oel+16a; Oel+17; Oel+18b] and [Oel+18a].

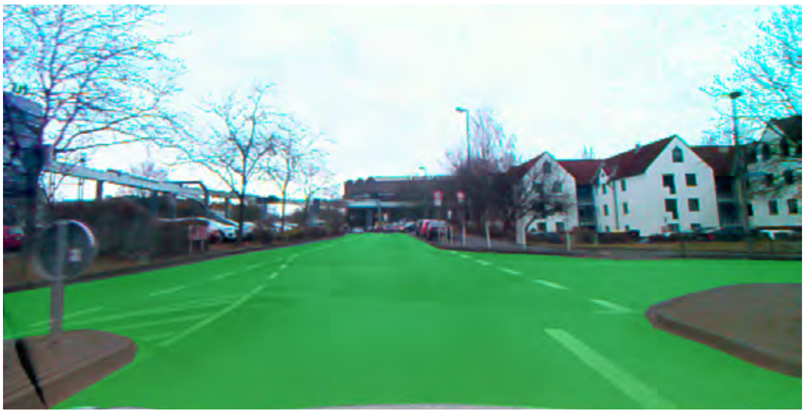


Figure 6.1: Example image recorded from the test vehicle near TU Dortmund University. The ground truth annotations of the drivable road area are highlighted in green.

6.1 Traffic scene segmentation as dense classification

From a technical point of view, segmentation corresponds to a dense classification. Instead of a single output path, as discussed for the road topology recognition, one classification output path is created for each image pixel. Furthermore, the transfer learning strategy is generally of crucial importance for image segmentation as the annotation effort is much higher when compared to other perception tasks. However, typical architectures for CNNs that are applicable to transfer learning include a successive sub-sampling up to a scalar output path. This contradicts the high-resolution output path required for segmentation. Since a semantic segmentation usually requires the same resolution as the input data, the feature maps following the first sub-sampling step thus require additional adjustments.
To this end, the following discusses methods to adapt common recognition CNNs into architectures that provide high-resolution output paths.

Discussion of segmentation strategies and general method

Various concepts are feasible for adapting a CNN architecture to the problem of segmentation in general. Below some of the basic alternative ideas are discussed and compared. Following this, a suitable approach for road segmentation is selected. The considered approaches, that will be discussed in the following, are given as:

- Patch-wise classification through an unmodified image recognition architecture
- Elimination of the gradual sub-sampling through dilated convolutions
- Skip connections and bilinear interpolation for fully convolutional fusion of high- and low-resolution features

Regarding patch-wise classification [Gir+14; Bru+15], a simple approach towards segmentation is to run an ordinary recognition network multiple times for different sections of an image. The repeated execution evaluates all network layers for multiple image patches, similar to a sliding window approach. Thus, this procedure allows the generation of a densely resolved output image. However, the sub-sampling stages contained in the architecture remain unchanged, so the resolution of the segmentation output cannot reasonably match that of the input images. Also, the naive implementation with repeated inference execution is very inefficient, because overlapping image areas must be evaluated for a practicable resolution of the obtained segmentation. This has the effect of re-computing the exact same image features multiple times. The main disadvantages of this approach are thus its computational inefficiency and the insufficient resolution of the obtained segmentation.
Another viable method is dilated convolution as in [YK16]. Herein, the convolution filter kernels are inflated and interleaved by additional rows and columns of zeroes. This approach allows applying the filter kernels to feature maps that have not been sub-sampled, without mathematically changing the results of the convolution. As a result, the respective pooling layers can be omitted entirely. All features, therefore,

need to be computed only once and this method does not involve redundant computations. The high resolution of the feature maps also allows for a comparatively high segmentation performance. However, this characteristic also massively increases the memory requirements of the corresponding CNN architectures. Therefore, it is hardly applicable for embedded hardware systems as the ones considered throughout this work.

Another approach, which maintains sub-sampling and at the same time does not involve any re-computation of feature maps, are the so-called fully convolutional networks (FCNs) [She+17]. FCNs constitute an end-to-end trainable CNN architecture designed specifically for the task of semantic segmentation. Herein, a recognition CNN is adapted for the task of segmentation, by branching the network at intermediate layers and combining these branches to form a pixel-dense new output path with preserved spatial resolution. For this, an adjustment of the feature resolutions is implemented through bilinear interpolation. The feature maps are thus scaled to match a specific resolution and then merged into combined features in a fusion step. The approach is inspired by the idea to combine high level semantic information from deep, low-resolution layers, with highly resolved spatial information from shallow layers near the network input. Similar to the dilated convolutions, a re-computation of the feature maps can be avoided due to the end-to-end nature of the approach. In addition, FCNs result in less high-resolution feature maps, since the pooling layers remain in place. The need for computational resources, especially the memory requirements during online execution, is thus substantially lower than with the other discussed approaches. FCNs, therefore, offer good prerequisites for the considered application in embedded systems. Due to this, they are used as a basis for the drivable road area segmentation in the following.

6.2 Segmentation decoder architecture and spatial priors

As outlined, the road segmentation approach is based on combining the intermediate feature maps by parallel network paths and thus generating a high-resolution output path. Consequently, it is necessary to align the dimensions of the feature maps step-wise, so that only features of the same dimension are combined and the output path can reach the original resolution of the input image. An overview of the decoder architecture designed according to these principles is shown in Figure 6.2. Herein, the 1×1 convolutions serving as input bottleneck layers in the segmentation decoder are chosen to compress the input features to $u_3 = 10$. Following the one-hot class encoding, the number of feature channels generated by the last bottleneck layer corresponds to the number of distinguished segmentation classes. For the examined segmentation of the drivable road area, this corresponds to $u_3 = 2$, accounting for the road area and the background class respectively.

The alignment of the feature map dimensions is based on bilinear interpolation, similar as described in section 3.2. However, a strict bilinear interpolation unnecessarily restricts the flexibility of the model. According to [She+17], it is advantageous in practice to use bilinear interpolation only for the initial model. Then, during the training

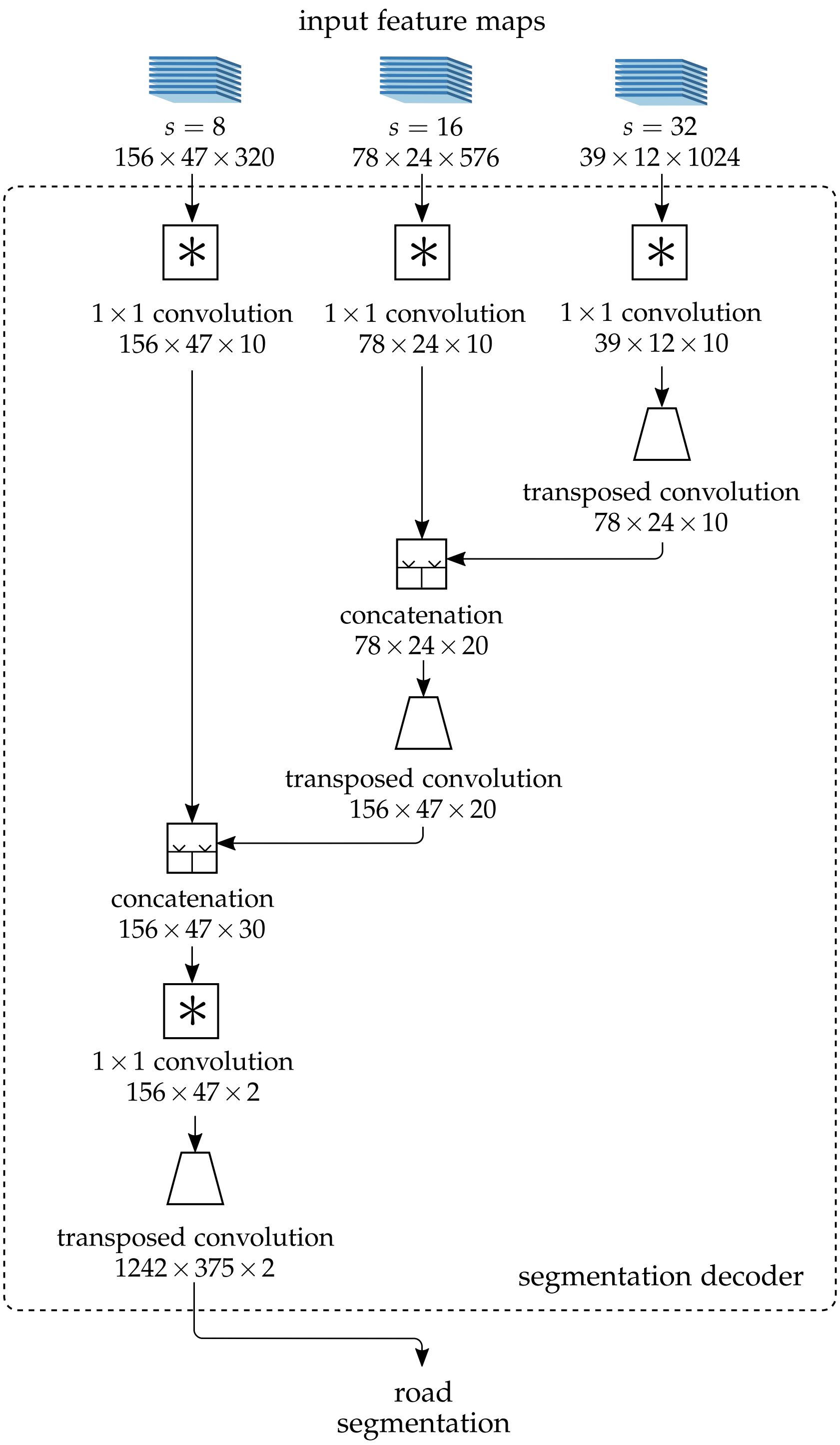


Figure 6.2: Schematic overview of the general segmentation decoder. Multiple inputs are fused through tensor concatenation. All bottlenecks layers compress the feature channel dimension to $u_3 = 10$.

process, deviating forms of interpolation are explicitly allowed.
For the technical implementation, the transposed convolution can be used, which switches the prediction inference (forward) and gradient (backward) computation of the regular convolution and is therefore also termed as deconvolution. This results in a mapping analogous to a regular convolution, however, the input features are interleaved with additional rows and columns of zeros. For a bilinear scale factor of two, the resulting modified feature map $\mathbf{H}_l^\diamond$ can thus be determined as follows[9].

$$\mathbf{H}_l^\diamond = \begin{pmatrix} h_l\left(0,0,u_3\right) & 0 & h_l\left(1,0,u_3\right) & 0 & \dots & h_l\left(u_{1,\max},0,u_3\right) \\ 0 & 0 & 0 & 0 & \dots & 0 \\ h_l\left(0,1,u_3\right) & 0 & h_l\left(1,1,u_3\right) & 0 & \dots & h_l\left(u_{1,\max},1,u_3\right) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_l\left(0,u_{2,\max},u_3\right) & 0 & h_l\left(1,u_{2,\max},u_3\right) & 0 & \dots & h_l\left(u_{1,\max},u_{2,\max},u_3\right) \end{pmatrix} \tag{6.2.1}$$

In analogy to the regular convolution, filter kernels $\mathbf{W}_\mathrm{k}$ are then evaluated for all coordinates of the feature map.

$$\mathbf{H}_l = \mathbf{H}_{l-1}^\diamond * \mathbf{W}_{\mathrm{k},l} + \mathbf{B}_l \tag{6.2.2}$$

Again, $\mathbf{H}_l$ denotes the output feature map and $\mathbf{B}_l$ accounts for the respective bias parameters. The weight initialization for the elements $w_{\mathrm{k},l}$ of the $N_\mathrm{k} \times N_\mathrm{k} \times u_3$ filter kernel $\mathbf{W}_{\mathrm{k},l}$ then results as follows.

$$w_{\mathrm{k},l}\left(u_1,u_2,u_3\right) = \frac{1}{\left\lceil \frac{N_\mathrm{k}}{2} \right\rceil^2} \cdot \left(\left\lceil \frac{N_\mathrm{k}}{2} \right\rceil - \frac{\left|2u_1 - N_\mathrm{k} + 1\right|}{2} \right) \cdot \left(\left\lceil \frac{N_\mathrm{k}}{2} \right\rceil - \frac{\left|2u_2 - N_\mathrm{k} + 1\right|}{2} \right) \tag{6.2.3}$$

For the merging of the parallel paths, [She+17] employ a simple addition of the feature maps. However, this does not take into account that the feature maps value ranges may differ, which would lead to an unintended preference for certain network paths. Furthermore, the addition is done element-wise, so that those features with the same channel dimension u_3 are added. Therefore, features may be combined even if they don't fit well semantically and better pairings would be available. These considerations suggest that improved approaches to feature fusion can support an enhanced segmentation performance.
Tensor concatenation along the u_3 dimension can be used instead to merge the feature maps, which mitigates the described disadvantages. A drawback of this approach, however, is the higher memory requirement resulting from the preservation of all original features. As compensation, bottleneck layers are again used to reduce the feature channel dimensions u_3. Bottleneck compression through a 1 × 1 convolutional layer results in a weighted sum of the concatenated features. The bottlenecks thus introduce another effect in that the weighting factors are learned during the training phase, which enables the fusion of semantically appropriate pairings of features. Hence, the precise mapping of the fusion is left to the optimizer.

[9] Note, that this simplified expression ignores some effects such as padding, for further details the interested reader is referred to [DV16].

Note, that this general segmentation decoder architecture results in significantly more loss function signals per image than in the case of the image recognition architecture. However, when constructing a multi-task network architecture, it is not desirable to prefer individual tasks through larger gradients. To account for this, the cross entropy loss function from equation 2.3.12 is normalized by the dimensions of the feature maps. Consequently, the segmentation loss results as follows.

$$L_{\text{Seg}}\left(\mathbf{Y}_{N_\text{L},j}, \mathbf{H}_{N_\text{L},j}\right) = -\frac{1}{u_{1,\max} \cdot u_{2,\max}} \sum_{\mathbf{u}} \ln\left(y_{N_\text{L},j}\left(\mathbf{u}\right) \cdot \varphi_\text{s}\left(h_{N_\text{L},j}\left(\mathbf{u}\right)\right)\right) \tag{6.2.4}$$

Herein, $\mathbf{Y}_{N_\text{L},j}$ and $\mathbf{H}_{N_\text{L},j}$ denote the road segmentation ground truth and output feature map tensors of training sample j, $y_{N_\text{L},j}$ and $h_{N_\text{L},j}$ their respective elements and φ_s is the softmax function. Altogether, the decoder used in this thesis for predicting the drivable road area follows a FCN architecture, in which tensor concatenation with a subsequent bottleneck layer is used for feature fusion.

Class distributions and spatial priors in traffic scenes

The segmentation decoder described so far was designed for a general problem definition. Special characteristics of traffic scenes, that can be exploited to improve the segmentation performance, have not yet been taken into account. An obvious characteristic is that in road area segmentation, the class distribution depends on the pixel position within the image. For example, it can be assumed that in a typical traffic scene the area directly in front of the ego vehicle belongs to the road area, whereas this is not the case for areas near the side of the image. Similar considerations can also be made for other traffic elements so that it generally seems sensible to take the spatial class distribution into account for the segmentation of traffic scene images. Therefore, even without knowledge of the image content, a position-dependant a-priori probability ρ_κ for the occurrence of certain classes κ can be empirically determined.

$$\rho_\kappa\left(u_1, u_2\right) = \frac{1}{N_\text{train}} \cdot \sum_{j=0}^{N_\text{train}} y_{N_\text{L},j}\left(u_1, u_2, u_3 = \kappa\right) \tag{6.2.5}$$

Again, $y_{N_\text{L},j}$ denotes the respective ground truth and N_train is the size of the training dataset. To ensure that this information can be utilized, a network architecture should therefore be designed to explicitly capture this spatial class distribution. In classical CNNs, the mappings of the different network layers have defined receptive fields that determine the influence of spatial information on the feature maps and thus also on the classification result. With the segmentation decoder outlined in Figure 6.2, only mappings whose receptive fields are significantly smaller than the typical resolution of the input images are incorporated. In theory, the stacking of multiple receptive fields can increase the global receptive field size of the overall model.

However, typical global receptive field sizes are still considerably smaller than the resolution of modern ADAS cameras. This holds especially, as practically observed global receptive fields are even smaller than their theoretical sizes. For example, [Zho+15] observed global receptive fields with a diameter of fewer than 100 px for a typical CNN, similar observations are also reported by [Liu+15; Luo+16]. Also, convolutional layers replicate identical weights at every spatial location of an image, an effect which is also known as weight sharing. This prevents efficient utilization of spatial priors, as they are neither captured in sufficiently large receptive fields nor encoded in position-dependant weights. Therefore a corresponding extension of the segmentation decoder described so far is outlined below.

Hadamard layer for pixel-wise weighting

An obvious attempt to improve the model architecture for the effective utilization of spatial priors would be to increase the receptive field sizes. The naive implementation leads to convolutional layers whose filter matrices $\mathbf{W}_\mathrm{k}$ have significantly increased dimensions. Maximum size convolution kernels are identical with fully connected layers and their number of parameters is the product of the input and output dimensions. This curse of dimensionality limits the practical applicability of this approach.
A more feasible method is the introduction of position-dependant feature encodings. For this, [Bru+15] proposes the inclusion of constant feature maps that directly encode the image coordinates. However, this manual engineering contradicts the concept of optimization-based feature learning, which is why a different approach is proposed in the following. As an alternative method, position-dependant weight factors can be introduced. These allow encoding the spatial class distribution directly in the form of a heatmap representation. This becomes apparent, when the connectivity patterns of the common network layers are compared with those of the Hadamard layer, see Figure 6.3 for an illustration. Since no adjacent image coordinates are evaluated for the calculation of the output features, the size of the receptive fields of the CNN is not affected. The heatmap can be multiplied element-wise with the feature maps, the resulting relationship is known as the Hadamard product. The corresponding notation of the Hadamard operator is shown below.

$$\begin{aligned}\mathbf{H}_l &= \mathbf{H}_{l-1} \circ \mathbf{W}_{l-1} \\ h_l\left(u_1, u_2, u_3\right) &= h_{l-1}\left(u_1, u_2, u_3\right) \cdot w_{l-1}\left(u_1, u_2, u_3\right)\end{aligned} \tag{6.2.6}$$

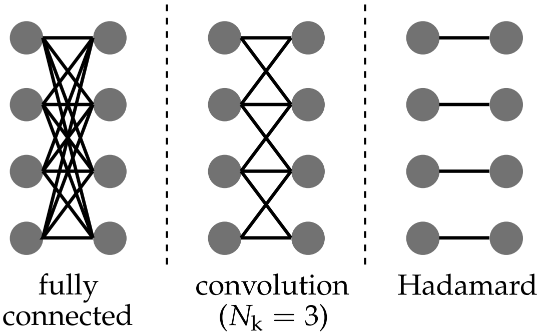


Figure 6.3: Comparison of the common connectivity patterns and the proposed Hadamard layer. Element-wise connections only influence features at the same spatial position, therefore they do not affect on the resulting receptive field sizes.

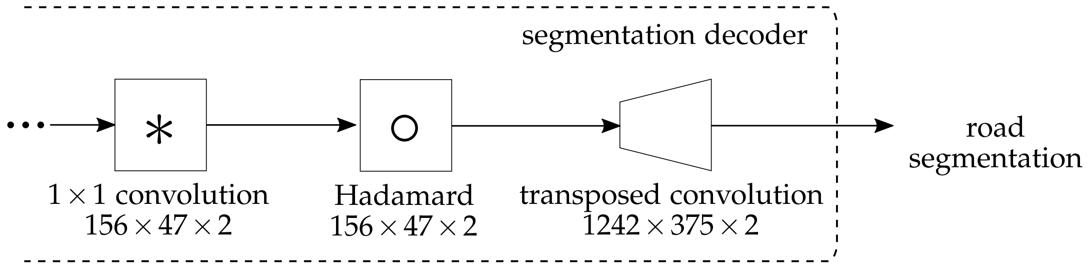


Figure 6.4: Modification of the segmentation decoder with the Hadamard layer (the full illustration can be found in the appendix in Figure A.4). The element-wise weights are added near the segmentation output. For reduced computational complexity, the Hadamard layer is located in-between the final fusion and the upsampling stage.

It follows, that the dimensions of the input feature map $\mathbf{H}_{l-1}$, the weighting matrix $\mathbf{W}_{l-1}$, and the output feature map $\mathbf{H}_l$ are identical. As a result, the Hadamard Layer can be understood as a mask, which emphasizes or suppresses certain image areas depending on their position.

An overview of the segmentation decoder with incorporation of the a priori class distribution is shown in Figure 6.4. Since the Hadamard layer aims to encode the position-dependant class distribution, it is reasonable to use this layer in conjunction with the final class representation near the output of the network. An integration before the last interpolation step of the generic segmentation decoder is suitable for this purpose, as the corresponding feature map already represents the output classes and the reduced resolution before the final interpolation positively influences the computational requirements of the resulting model.

End-to-end segmentation network architecture

For the implementation of this approach, the integration into an end-to-end architecture consisting of the inception-v2 feature encoder and the described segmentation decoder is carried out first. Analogous to the previous illustrations, Figure 6.5 shows the corresponding overview of the integration of the segmentation decoder. It can be seen, that the features of the first block of the inception-v2 architecture are omitted for the use in the segmentation decoder, which makes sole use of the feature blocks

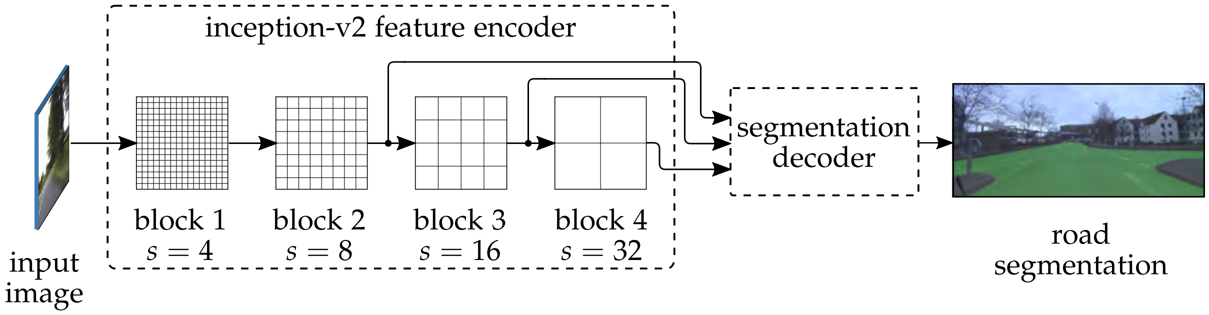


Figure 6.5: Schematic overview of the integration of the segmentation decoder with the inception-v2 feature encoder using intermediate feature maps of different spatial resolutions.

two to four. This design choice is again mainly due to computational reasons, as the features of the first block have a prohibitively high spatial resolution. To investigate the effect of the element-wise weights, it is useful to compare both variants of the model architecture. For this purpose, the segmentation decoder indicated in Figure 6.5 is interchangeable. An end-to-end model can thus be implemented either with the segmentation decoder without the Hadamard layer, as illustrated in Figure 6.2, or with element-wise Hadamard weights for he explicit encoding of spatial class priors, as illustrated in Figure 6.4. For the variant with existing Hadamard layer the term *Hadamard-FCN* is used in the following, while the model architecture without Hadamard layer is termed as *Plain-FCN*.

6.3 Experiments on drivable road area segmentation

With the discussed design of the segmentation decoder, the two model variants are evaluated and compared in the following. For further assessment, a comparison with other works from the relevant literature is also carried out. The evaluation is focused on the segmentation of the drivable road area since this is one of the main objectives of image segmentation in automotive environment models.

Benchmark datasets and evaluation protocol

The comparative assessment is carried out based on datasets annotated for the binary problem of road segmentation. Firstly, the evaluation is based on the KITTI road benchmark dataset [Fri+13], which has been well established in the relevant literature, so that numerous comparable results of alternative approaches are directly available. This dataset comprises 579 images ($N_{\text{train}} = 289$, $N_{\text{test}} = 290$) from a front facing camera recorded in urban traffic environments taken on 5 different days and with a minimum distance of 20 m to avoid overly correlated images.

Secondly, a corresponding dataset was created with the Nissan Leaf test vehicle presented in section 3.1, which is dubbed as InVerSiV dataset in the following. For this purpose, the test vehicle was used to acquire images near the city of Dortmund on two different days. To avoid correlated recordings, a time delay between successive images was ensured and a separate manual inspection of the obtained dataset was performed. Afterwards, as part of the present work, the images were supplemented by a manual annotation of the road area. Compared to the KITTI road benchmark, the annotation of the road area is however less precise since only a rough outline of the contours in the form of a polygonal chain was registered. The InVerSiV dataset contains $N_{\text{train}} = 240$ images of traffic scenes for training and $N_{\text{test}} = 60$ images for testing (300 total).

Following [Fri+13], pixel-based performance measures are evaluated. Note that, in contrast to road topology recognition, the binary performance measures can be evaluated directly for road area segmentation. This specifically applies to the calculation of the *pre*, *rec*, and $F1$ values analogous to equation 5.3.4. According to the evaluation protocol of [Fri+13], however, the pixel class prediction is not determined solely by the pixel confidence scores. Instead, a decision threshold τ^* is firstly chosen by maximizing $F1$ as given in the following equation.

$$\tau^* = \underset{\tau}{\operatorname{argmax}} F1 = \underset{\tau}{\operatorname{argmax}} \left(2 \frac{pre \cdot rec}{pre + rec} \right) \tag{6.3.1}$$

Subsequently, the binary performance measure $F1_{\max}$ and the corresponding *pre* and *rec* values are calculated based on the optimized threshold value τ^* and examined for the following evaluation.
Also, the intersection-over-union (IoU, sometimes also termed as Jaccard index) measure is taken into account, which provides a direct measure of the overlap between the generated road segmentation and the annotated ground truth road area. It is calculated as follows.

$$IoU = \frac{TP}{TP + FP + FN} \tag{6.3.2}$$

For a conclusive evaluation, the mean IoU is determined over the images of the test partitions of the datasets. Furthermore, similar to the road topology recognition task, the mAP measure is also examined based on the *pre-rec* curve. For the calculation of the mAP measure, the interpolation according to equation 5.3.6 and the calculation based on 11 equidistantly distributed points as in equation 5.3.7 is again employed.
To evaluate the performance of the road area segmentation with respect to a practical application for automated driving functions, a transformation of the segmentation map into a BEV image according to the relationship elaborated in chapter 3.2 is carried out. Note that in the perspective view, areas near the camera in front of the ego vehicle occupy a proportionally larger area in the image. After performing the BEV transformation, a more balanced representation emerges, so that near and far areas have an identical weight when evaluating the segmentation performance measures. Since the close areas usually exhibit a more homogeneous appearance, it can therefore be assumed that the performance measures resulting in BEV-space are generally lower. Furthermore, note that in contrast to [Fri+13], no cropping of the BEV images near the ego vehicle is performed for the InVerSiV dataset.

Comparative study on road segmentation performance

For analyzing the effect of the element-wise weights, both the Hadamard-FCN as well as the Plain-FCN are trained on the two datasets respectively. The optimization setup is chosen identical to the one described in section 5.3, except for the batch size that is chosen to $N_{\text{batch}} = 1$ as well as the Hadamard weights that are initialized to the constant value of one. The choice of the smaller batch size stems from the added memory consumption of the segmentation model due to the added high-resolution feature maps. Again, no data augmentation was used to not affect the spatial priors in the dataset. The resulting performance measures are given in the following Table 6.1, additionally a more detailed breakdown of the KITTI road results can be found in the appendix under section A.5. For further assessment, the general relevance of spatial priors for the perception of the drivable road area is examined. For this purpose, the average distribution of the ground truth annotations was determined and accordingly

Table 6.1: Comparison of the pixel-based performance metrics of the Hadamard-FCN and Plain-FCN, evaluated on the test splits of the used datasets. The runtimes stem from the publications or, if no publication was specified, were measured with a desktop GPU (see Table A.2.d).

InVerSiV dataset (perspective view)						
	$F1_{\max}$	*pre*	*rec*	*mAP*	*IoU*	runtime
Hadamard-FCN	**92.33 %**	**90.45 %**	94.29 %	**91.26 %**	**85.76 %**	19.8 ± 1.1 ms
Plain-FCN	89.92 %	88.63 %	91.26 %	90.52 %	81.70 %	19.4 ± 1.7 ms
prior-baseline	84.48 %	76.49 %	**94.33 %**	86.13 %	73.12 %	n/a

InVerSiV dataset (BEV)						
	$F1_{\max}$	*pre*	*rec*	*mAP*	*IoU*	runtime
Hadamard-FCN	**90.70 %**	**90.22 %**	**91.18 %**	**90.58 %**	**82.98 %**	19.8 ± 1.1 ms
Plain-FCN	88.35 %	87.42 %	89.29 %	89.99 %	79.12 %	19.4 ± 1.7 ms
prior-baseline	69.63 %	60.30 %	82.37 %	74.12 %	53.40 %	n/a

KITTI road benchmark dataset (BEV)						
	$F1_{\max}$	*pre*	*rec*	*mAP*	*IoU*	runtime
Hadamard-FCN	**94.85 %**	**94.81 %**	**94.89 %**	91.48 %	n/a	19.8 ± 1.1 ms
Plain-FCN	92.26 %	92.80 %	91.72 %	91.83 %	n/a	19.4 ± 1.7 ms
prior-baseline[10]	73.63 %	69.98 %	77.69 %	78.84 %	58.27 %	n/a
[Han+17]	91.57 %	90.02 %	93.19 %	84.68 %	n/a	6000 ms
[Tei+18]	93.99 %	94.51 %	93.48 %	**93.24 %**	n/a	98.1 ms

evaluated as a constant prediction independent from the actual image contents. This analysis is used to estimate the importance of the spatial priors and will be referred to as prior-baseline in the following.

Additionally, results from the relevant literature are also included for the KITTI road dataset. Moreover, the respective runtimes are included where available. Note, that since the evaluation for the KITTI dataset was carried out with the help of the provided evaluation server, the *IoU* measure as well as an evaluation in perspective space was not possible here.

Notably, the Hadamard-FCN model outperforms the Plain-FCN model without the Hadamard layer consistently for both datasets. Furthermore, the comparison with the prior baseline shows, that even this simple estimation of the road area can solve a significant amount of the perception task, which is also confirmed by similar findings reported in [Fri+13; Bru+15]. Both results underline the significance of spatial priors for the segmentation of traffic scenes and demonstrate, how FCN networks can benefit from the proposed element-wise Hadamard weights.

Again, the obtained *pre* and *rec* measures demonstrate an overall balanced behavior,

[10] The baseline for the a-priori class distribution was measured on the training split of the KITTI road dataset.

however, in the case of the InVerSiV dataset, a slightly increased tendency towards *FP* errors is observed. Besides, the results show a generally lower performance for the InVerSiV dataset. It can therefore be assumed that the task of road segmentation is more difficult to learn with this dataset, presumably due to its smaller size and more coarse annotations.

In general, it can also be stated that the achieved model performance is on par with other current methods from the literature. However, when comparing the average runtimes, it becomes apparent that the present work achieves considerable fast runtimes and is thus better suited to enable real-time processing in automotive applications. Furthermore, as expected from the previous discussions, the resulting measures in BEV-space are slightly lower than in perspective space.

Since the segmentation of the drivable road area comprises a binary problem, the *pre* and *rec* values can directly be weighted against each other. Again, the corresponding *pre*-*rec* curves provide an assessment of the model's ability to balance the respective *FP* and *FN* cases. Due to its superior performance, only the Hadamard-FCN is considered in the following, such that Figure 6.6 provides the respective curves only for this model. First of all, it is noticeable that the resulting plots show a clear tendency to follow the ideal, step-like curve. Due to the similar shape of the curves, Figure 6.6 additionally shows a zoomed in view of the particularly relevant area. Here, it is again notable that a slightly superior curve is obtained for the evaluation in the perspective space than in BEV. Furthermore, a slightly superior curve is again obtained for the KITTI dataset, which also turns out generally smoother. Here, a significant difference between the two datasets becomes apparent, as the KITTI dataset has considerably more test images.

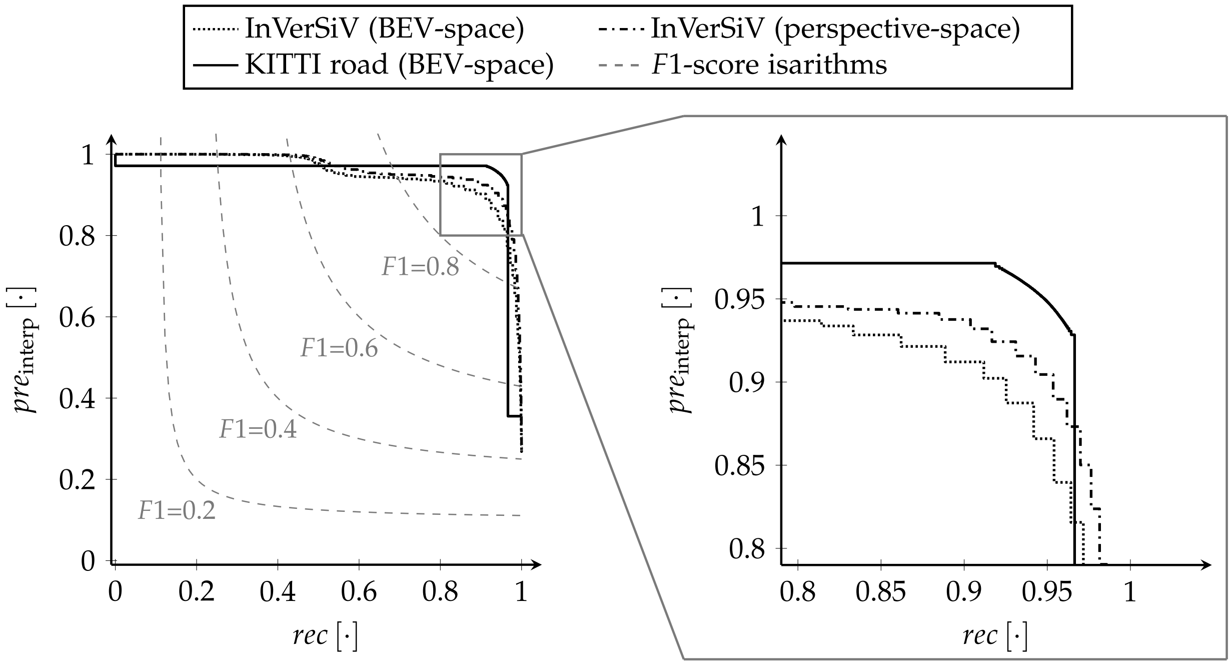


Figure 6.6: *pre*-*rec* curves for the Hadamard-FCN obtained from the test images of the KITTI road and InVerSiV datasets. The results for the KITTI road dataset were determined on the official evaluation server.

🟩 *TP* 🟥 *FP* 🟦 *FN*

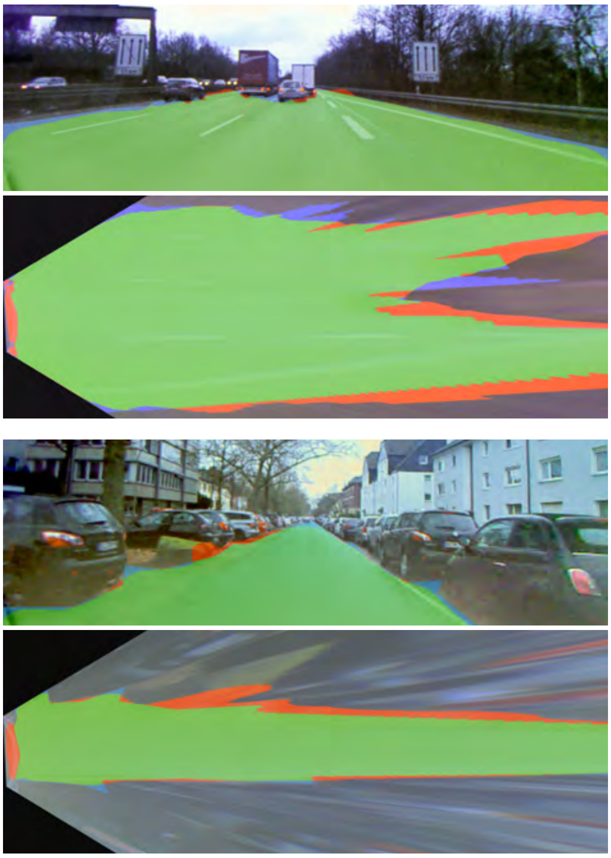


Figure 6.7.a: Example segmentation maps of the Hadamard-FCN on the InVerSiV test dataset.

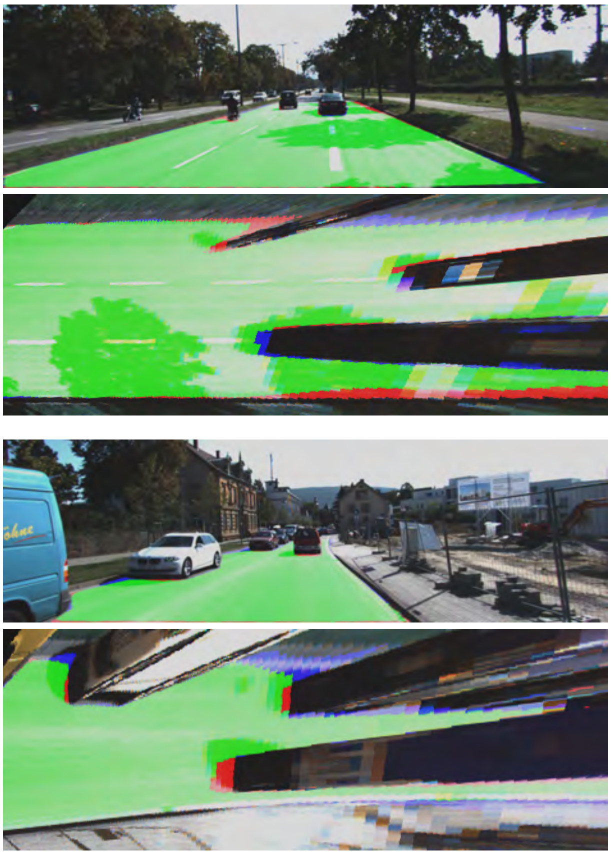


Figure 6.7.b: Example segmentation maps of the Hadamard-FCN on the KITTI road test dataset.

For an additional qualitative assessment of the road segmentation, example segmentation images of the Hadamard-FCN from the respective test datasets are considered. A corresponding representation is given in Figure 6.7.b and Figure 6.7.a. Herein, the road segmentation is shown both for the original images in perspective space and after the transformation into BEV-space. To distinguish the segmented pixels into the cases TP, FP, and FN, they have been highlighted in different colors. From the illustrations, it can be seen that the drivable road area is segmented correctly for the most part. In this way, misclassifications occur mostly at the contours of the road area.

A comparison of the images in perspective and BEV-space further shows that the existing misclassifications take up a larger area in the BEV images. This effect of the BEV transformation again illustrates the somewhat lower values of the performance measures for the BEV in Table 6.1. Furthermore, it can be seen from Figure 6.7.b and Figure 6.7.a, that the most significant misclassifications are in those image regions that represent far away areas of the scene. This can be explained by the fact that the traffic elements located at a higher distance only take up a few pixels in the original image and therefore comparatively less information is available for their respective segmentation.

For a further qualitative assessment, a visualization of the spatial distribution of the pixels belonging to the road area is considered. For this purpose, Figure 6.8 shows the average ground truth annotations for the two datasets respectively, as well as a representation of the element-wise weights of the Hadamard layer of the trained models. At first sight the relatively noisy Hadamard weights are noticeable in direct comparison. Besides the generally lower resolution, this can be attributed mainly to stochastic influences during model optimization, such as the stochastic batch optimization or the random initialization of some of the weight parameters. In general, however, it can be clearly seen that the average spatial distribution of the road area is reflected in the optimized Hadamard weights.

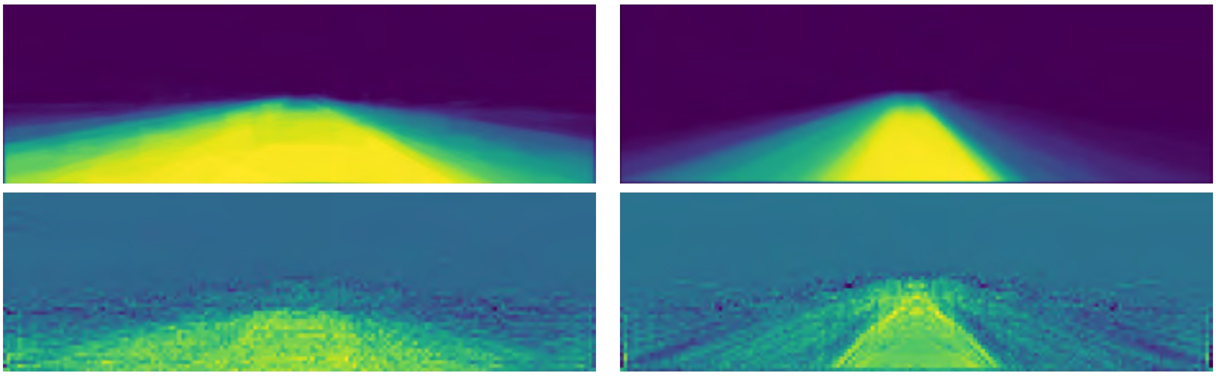


Figure 6.8: Heatmaps of the average drivable road area. Top left: InVerSiV dataset, top right: KITTI road dataset, bottom: Hadamard weights $\mathbf{W}_l$ learned on the respective datasets.

7 Road Users from Bounding Box Detection

Object detection aims to locate objects in an image by means of bounding boxes. Generally, this type of representation is suitable for those traffic elements where individual object instances can be clearly distinguished from each other. This is especially the case for other road users present in a given scene so that in the context of automotive applications an object detection can be used for example to detect other vehicles. For this, a challenge arises from the fact that subsequent processing steps require a reconstruction of the 3D parameters that describe the object's spatial properties. Thus, it is not sufficient to determine enclosing bounding boxes of objects in the image space.
In the spatial reconstruction, the object's position is determined by the observer's point of view, which is defined by the observation angle and the distance. To realize object detection with a CNN, the feature encoder is extended by a detector stage. The design of this detection stage is the subject of this chapter. For this purpose, some general considerations regarding the basic approach to locate objects in 2D image coordinates are given first. Following this, a procedure for the reconstruction of viewpoints and spatial descriptions of the detected objects is elaborated. Furthermore, the chapter concludes with a comparative experimental evaluation of the obtained results. Parts of the following explanations have been published in the papers [Oel+18a; Oel+19b] and [Oel+19a].

7.1 Classification and localization of 2D bounding boxes

For the realization of a bounding box decoder, it is necessary to define a numerical description of the bounding boxes suitable for CNNs. A description of the 2D bounding box requires at least four parameters corresponding to the four boundaries in the image space. A naive implementation would directly perform a regression on these boundaries $u_{1,\mathrm{min}}, u_{1,\mathrm{max}}, u_{2,\mathrm{min}}, u_{2,\mathrm{max}}$. However, a direct regression is not advisable, if discontinuities in the depicted scene contents are to be expected. For this reason, mixed approaches consisting of a discrete classification and a continuous regression represent a superior alternative for object detection. Herein, a discretization of the object position is combined with a regression of the remaining discretization error, as illustrated in Figure 7.1. Thus, the determination of 2D bounding boxes requires two output paths of the detection decoder.

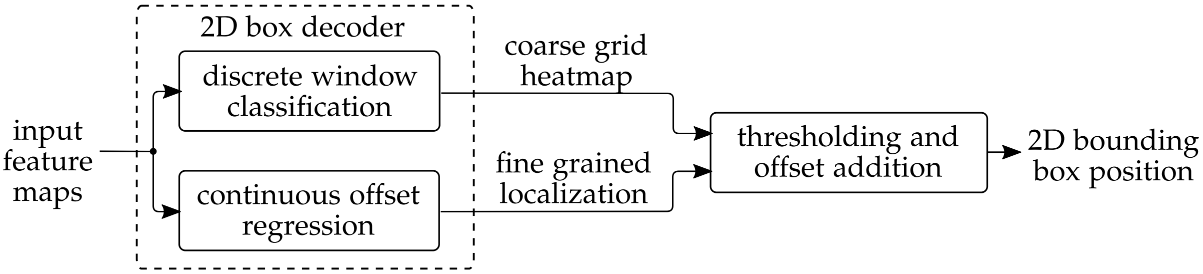

2D box decoder
input feature maps
discrete window classification
continuous offset regression
coarse grid heatmap
fine grained localization
thresholding and offset addition
2D bounding box position

Figure 7.1: Multimodal regression for object detection. The space of possible bounding boxes is divided into several discrete modes, which are mapped into a coarse grid heatmap classification. In another output path, the regression of the continuous discretization offset is performed, which is applied to each cell of the heat map to determine the final bounding boxes.

The classification of the discrete object position is done by mapping it to a feature map which, similar to a heatmap, assigns higher activation values to positions with existing objects and lower values to positions without objects. This procedure is similar to a sliding window, where the image contents are scanned with prototypical window boxes. Usually, a fixed size is assumed for the windows boxes, which are also referred to as *anchors*, see Figure 7.2 for an illustration. This general scheme has proven remarkably effective in numerous recent research results [Gir+14; Liu+16a; Ren+17]. As a concise explanation for these observations, [Mou+17] argue that for the position of objects with clear contours, there is a hard transition between object and background at the boundaries, rendering object localization into a multi-modal regression problem. For a practical implementation, it is essential to incorporate multiple different parameterizations of the bounding box anchors to detect objects of different sizes and shapes. This is evident for example from the works of [Liu+16a], which found that features with multiple scales are better suited to ensure the detection of large and small objects. This property is of particular importance in the context of automotive environment perception, since the size of the objects in the image can vary greatly in typical traffic scenes due to the large distance variations. Thus, different aspect ratios and dimensions of bounding box anchors can be taken into account by providing a separate heatmap for each parameterization of the anchors.

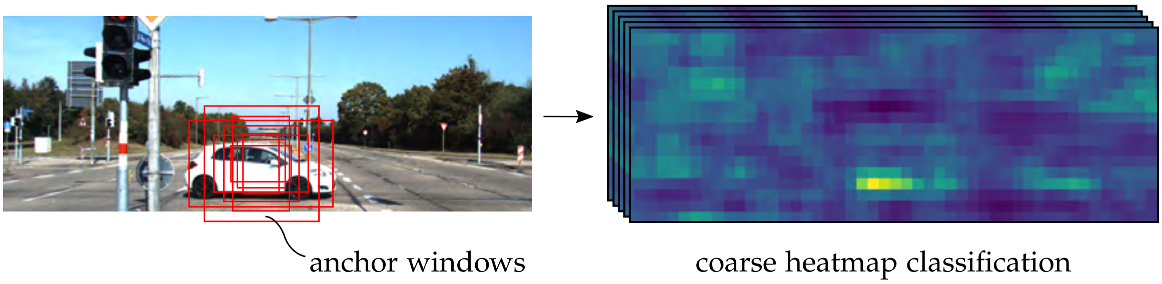


Figure 7.2: Visualization of an example street scene and the corresponding classification heatmap for the detection of vehicle bounding boxes. Each position in the heatmap represents the midpoint of one corresponding 2D bounding box.

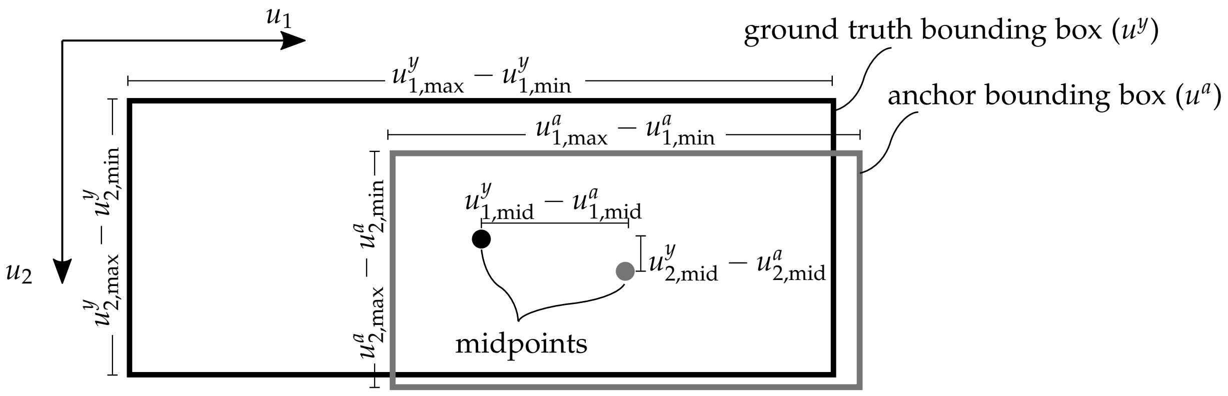

u_1
u_2
ground truth bounding box (u^y)
anchor bounding box (u^a)
$u^y_{1,\max} - u^y_{1,\min}$
$u^a_{1,\max} - u^a_{1,\min}$
$u^y_{2,\max} - u^y_{2,\min}$
$u^a_{2,\max} - u^a_{2,\min}$
$u^y_{1,\mathrm{mid}} - u^a_{1,\mathrm{mid}}$
$u^y_{2,\mathrm{mid}} - u^a_{2,\mathrm{mid}}$
midpoints

Figure 7.3: Illustration of the regression target variables. The deviations of the actual bounding box from the anchor bounding box are given by the midpoint displacement and the width and height offsets.

The computation of the heatmaps in a fixed grid inevitably leads to a discretization offset, the influence of which is even greater if a reduced resolution is used for the corresponding feature map. For reasons of computational efficiency, however, this is regularly the case in any practical implementation. To compensate for this remaining discretization offset, the relative displacement of the current object is additionally estimated for all coordinates of the heatmap, see also Figure 7.3 for an illustration. For this purpose, the mapping according to [Ren+17] is used, in which the encoding of the displacement of the midpoint of the bounding box as well as of the deviations in height $u_{1,\max} - u_{1,\min}$ and width $u_{2,\max} - u_{2,\min}$ takes place. It is furthermore advantageous to normalize these regression targets with respect to the size of the bounding box, since this allows the universal application of the bounding box decoder without adjustments when processing different image resolutions. Moreover, the regression of the size of the bounding boxes on a logarithmic scale is numerically advantageous according to [Ren+17], due to the wide range of bounding box dimensions. Thus, the target variables for the bounding box offset regression are given as follows.

$$\begin{aligned} y_{\mathrm{mid},u_1} &= \frac{u^y_{1,\mathrm{mid}} - u^a_{1,\mathrm{mid}}}{u^a_{1,\max} - u^a_{1,\min}} \; , \; y_{\mathrm{mid},u_2} = \frac{u^y_{2,\mathrm{mid}} - u^a_{2,\mathrm{mid}}}{u^a_{2,\max} - u^a_{2,\min}} \; , \\ y_{\mathrm{w}} &= \log\left(\frac{u^y_{1,\max} - u^y_{1,\min}}{u^a_{1,\max} - u^a_{1,\min}}\right) \; , \; y_{\mathrm{h}} = \log\left(\frac{u^y_{2,\max} - u^y_{2,\min}}{u^a_{2,\max} - u^a_{2,\min}}\right) \end{aligned} \tag{7.1.1}$$

Here, y_{mid,u_1} and y_{mid,u_2} account for the displacement of the 2D bounding box midpoint, and y_{w} and y_{h} for the respective deviations in width and height.

Discussion and preliminary selection of the detection method

To implement the general approach outlined above, a concrete decoder architecture needs to be defined. Two different approaches are considered for this:

- A single-stage decoder architecture for the direct derivation of object detections from the feature maps

- A two-stage architecture for the indirect determination of bounding boxes via intermediate object hypotheses

A characteristic feature of the first approach is that an end-to-end architecture is formed from the input images to the target variables defined in the previous section, which exclusively uses the basic network components specified in section 2.3. Furthermore, in the direct approach, no explicit intermediate representations are inherently enforced by the architecture, but instead, the composition of all feature maps is determined solely during the optimization process.

In the case of two-stage detectors, a rough determination of object hypotheses takes place first, which are only refined to an exact object detection in a second step. This is done by explicitly specifying object hypotheses as representations of intermediate feature maps through an additional term in the overall loss function. Based on the individual object hypotheses, the feature maps are sampled through a pooling operation, and from this the actual bounding boxes are determined.

In general, the two-stage approaches achieve higher detection accuracies. This is mainly the result of the more sophisticated data processing, as the consideration of object hypotheses allows gradual filtering of the bounding boxes. Thus, a rough preselection is established first, which is subsequently refined. At the same time, however, the computational effort is also correspondingly increased, see for example the investigation in [Hua+17]. For this reason, a single-stage approach is used for the architecture of the detection decoder.

Examples that follow this scheme include the works in [Liu+16a], [RF18], and [Wu+17]. The respective architectures differ in the choice of the anchors, the decoder layers (e.g. convolutional or fully connected), and the exact choice of multi-scale features. The SSD approach of [Liu+16a] uses a particularly wide variety of anchor boxes and more comprehensive multi-scale features and can be considered as a fairly universal method. Therefore a decoder architecture based on this approach is used. The following Figure 7.4 contains an illustration of the employed implementation. The generation of the output paths for the classification step as well as for the regression step is apparent through 1×1 convolutions. Besides, the multi-scale features are generated by subsampling, except for the highest resolution features, which are directly formed from an intermediate layer of the feature encoder. Note, that the feature map of the 2D box encodings provides a channel dimension of $u_3 = 4$, corresponding to the four target variables $y_{\mathrm{mid},u_1}, y_{\mathrm{mid},u_2}, y_{\mathrm{w}}$ and y_{h}.

Due to their importance for the driving task, but also for the availability of datasets and comparative performance analyses, the object detector is further implemented for the detection of other vehicles as typical road users. In principle, however, the method is also applicable to other types of objects, such as pedestrians or bicyclists. The channel dimension of the heatmap features for object classification is $u_3 = 2$ to accordingly reflect the vehicle and background class. The loss function used for object detection results as a superposition of the normalized smooth $L1$ loss and the negative log likelihood. Following [Liu+16a], both loss components are weighted equally. If $\mathbf{Y}_{\mathrm{Cls}}, \mathbf{H}_{\mathrm{Cls}}$ denote the targets and output features for the classification heatmap and $\mathbf{Y}_{\mathrm{Loc}}, \mathbf{H}_{\mathrm{Loc}}$

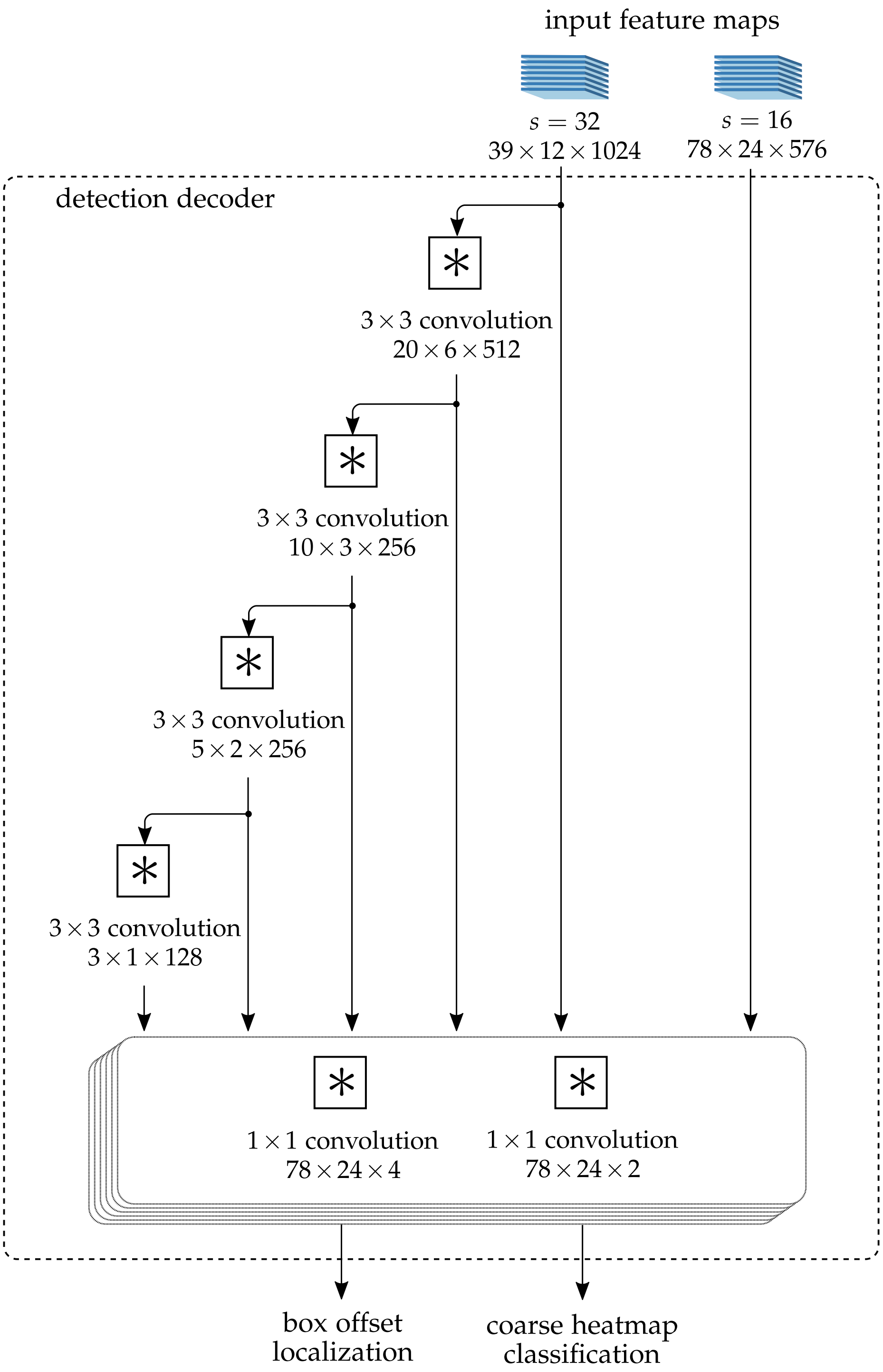


Figure 7.4: General architecture of the detection decoder based on the single shot method according to [Liu+16a]. Heatmap classifications as well as offset regression outputs are generated through 1×1 convolutions.

the respective tensors for the offset localization output path, then the combined loss function is given as:

$$L_{\mathrm{2Dbox}}\left(\mathbf{Y}_{\mathrm{Cls},j},\mathbf{H}_{\mathrm{Cls},j},\mathbf{Y}_{\mathrm{Loc},j},\mathbf{H}_{\mathrm{Loc},j}\right)=-\frac{1}{N_{\mathrm{Det}}}\sum_{v=1}^{N_{\mathrm{Det}}}\ln\left(y_{\mathrm{Cls},j}\left(\mathbf{u}_v\right)\cdot\varphi_{\mathrm{s}}\left(h_{\mathrm{Cls},j}\left(\mathbf{u}_v\right)\right)\right)$$
$$+\frac{1}{N_{\mathrm{Det}}}\sum_{v=1}^{N_{\mathrm{Det}}}\begin{cases}0.5\cdot\left(y_{\mathrm{Loc},j}\left(\mathbf{u}_v\right)-h_{\mathrm{Loc},j}\left(\mathbf{u}_v\right)\right)^2, & \text{if } \left|y_{\mathrm{Loc},j}\left(\mathbf{u}_v\right)-h_{\mathrm{Loc},j}\left(\mathbf{u}_v\right)\right|<1\\ \left|y_{\mathrm{Loc},j}\left(\mathbf{u}_v\right)-h_{\mathrm{Loc},j}\left(\mathbf{u}_v\right)\right|-0.5, & \text{else}\end{cases}, \tag{7.1.2}$$

where N_{Det} is the number of anchor boxes that overlap with annotated ground truth bounding boxes, $\mathbf{u}_v$ denotes their respective feature map coordinates and φ_{s} is again the softmax function. In the discussions so far, only the determination of detections in 2D image space is considered. The further generation of a spatial object description is examined in the following.

7.2 Auxiliary regressands and decoder architecture for spatial reconstruction

The determination of the spatial parameters for the detected vehicles forms an essential processing step in the context of automotive applications since any subsequent driving functions necessarily require a spatial scene description. For the investigation of possible solutions towards this, some fundamental ideas are discussed and compared below. These are given as:

- Representation transformation to derive low level 3D features
- Matching features with offline created shape models
- Utilization of constraints based on a back projection of the scene
- Direct prediction of 3D parameters through a learned model

With representation transformation, the reconstruction of the scene is performed at the feature level. The idea is to transform some or all of the feature maps into the BEV, or to perform an intermediate estimation of depth images. With this approach, however, leveraging the high level of maturity of 2D detection methods is rendered difficult.

Matching 2D detections with offline generated and stored shape models provides another alternative approach, which however yields a complex extension of the processing chain. If, as in [Mot+15], a grid search across all 3D parameters is performed, this approach also requires considerable additional computation effort. Furthermore, variations in the shape of different vehicle classes, such as small cars, vans, or coupés have to be taken into account in the implementation. This either requires additional annotation effort if the variations are incorporated in the learned model, or it causes a further increased search space if an exhaustive search across all vehicle classes is performed. Thus, substantial practical disadvantages arise from this approach, which contradicts the intended application.

When making use of constraints based on a back projection of the scene, the fact that many 3D parameters enter linearly into the camera model is exploited. Thus, known corresponding points in 2D image space and 3D world coordinates can be used to determine the 3D parameters using a linear least squares method. The computational costs of this method are small enough to be essentially negligible. However, 2D bounding boxes alone do not generate a sufficient number of measurements. Therefore, backprojection constraints can only be used with further assumptions about the scene or by combining them with other methods into a hybrid approach.

The direct estimation of the 3D parameters of the detected objects by the learned model is possible, too. Here, general statements about the applicability of this strategy are hardly possible. This is because the prerequisites for this approach strongly depend on the exact choice of the predicted 3D parameters. In this respect, especially those 3D parameters are suitable for a direct prediction that have a decisive and unambiguous effect on the visual appearance of the object and thus the image features. Furthermore, the availability of annotated training data influences the choice of the predicted 3D parameters in practice. The direct prediction of the 3D parameters can often be integrated into a CNN analogous to the 2D bounding box parameters, which enables an implementation with little additional computational demands.

Based on these considerations, an approach is pursued which combines backprojection constraints and a direct prediction of 3D parameters. The division of the full 3D reconstruction across these two prediction schemes pays special attention to the nonlinear parts of the camera model, the influence of individual 3D parameters on the visual appearance, and any introduced manual annotation effort.

Constraint and appearance-based viewpoint and 3D reconstruction

For the following consideration of the 3D detection method, some aspects of the reconstruction task are firstly reviewed in more detail. Analogous to the 2D bounding box, which is determined in image coordinates, the 3D bounding box defines an enclosing cuboid in the spatial scene description. The following Figure 7.5 shows an illustrative representation of the described relationships. In total nine parameters are required for the complete description of the 3D bounding box. They define the coordinates of the 3D object centroid $\mathbf{n}_{\mathrm{W}}^{\mathrm{C}} = \left(n_{x_1}^{\mathrm{C}}, n_{x_2}^{\mathrm{C}}, n_{x_3}^{\mathrm{C}}\right)^{\top}$, the orientation angles ϕ, ϑ, ψ (roll, pitch, yaw), and the object's length, width and height $\mathbf{d} = \left(d_{x_1}, d_{x_2}, d_{x_3}\right)^{\top}$ (measured in a right-handed and vehicle aligned coordinate system). For the mathematical representation of the orientation angles, a 3 × 3 rotation matrix $\mathbf{R}\left(\phi, \vartheta, \psi\right)$ is used in the following. With these definitions, a description of the 3D coordinates of the bounding box nodes can be derived.

The 3D world coordinates can be converted into 2D image coordinates with the known camera model and extrinsic calibration. The result is a vector that yields the image position in homogeneous coordinates. For example, consider a node $\mathbf{n}^{\mathrm{FR}}$ positioned at the front right (FR) bottom of the 3D bounding box enclosing a detected vehicle.

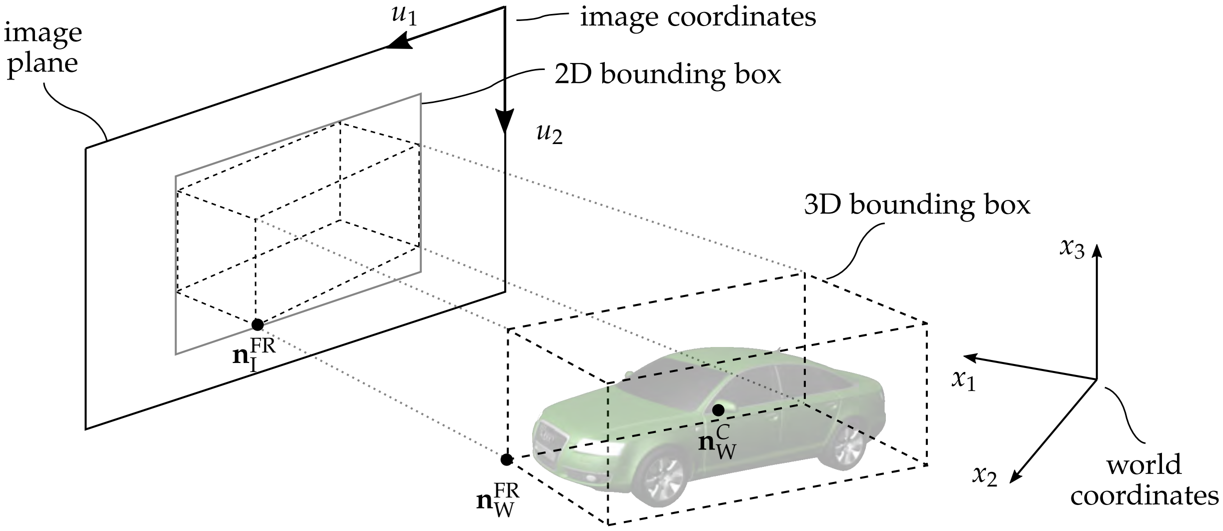


image plane
u_1
image coordinates
2D bounding box
u_2
3D bounding box
x_3
x_1
x_2
world coordinates
$\mathbf{n}_{\mathrm{I}}^{\mathrm{FR}}$
$\mathbf{n}_{\mathrm{W}}^{C}$
$\mathbf{n}_{\mathrm{W}}^{\mathrm{FR}}$

Figure 7.5: Schematic illustration of the tight-fit constraint of backprojected vehicles. Image coordinates of 3D bounding box nodes correspond to 2D box boundaries.

Then, the following equation indicates the relationship between its spatial and image coordinates.

$$\tilde{\mathbf{n}}_{\mathrm{I}}^{\mathrm{FR}} = \mathbf{P} \cdot \tilde{\mathbf{n}}_{\mathrm{W}}^{\mathrm{FR}} = \mathbf{P} \cdot \begin{pmatrix} \mathbf{R} & \mathbf{n}_{\mathrm{W}}^{C} \\ \mathbf{0} & 1 \end{pmatrix} \cdot \left(\tfrac{d_{x_1}}{2}, -\tfrac{d_{x_2}}{2}, -\tfrac{d_{x_3}}{2}, 1 \right)^{\top} \tag{7.2.1}$$

Herein, **P** again denotes the camera projection matrix, see section 3.2. Assuming that the node's backprojection corresponds with the bottom border of the 2D bounding box, then the resulting constraint is given by:

$$\begin{aligned} & \frac{\mathbf{p}_{2}^{\text{▬}} \cdot \tilde{\mathbf{n}}_{\mathrm{W}}^{\mathrm{FR}}}{\mathbf{p}_{3}^{\text{▬}} \cdot \tilde{\mathbf{n}}_{\mathrm{W}}^{\mathrm{FR}}} = u_{2,\max} \\ & \quad \Leftrightarrow \; 0 = \left(u_{2,\max} \cdot \mathbf{p}_{3}^{\text{▬}} - \mathbf{p}_{2}^{\text{▬}} \right) \tilde{\mathbf{n}}_{\mathrm{W}}^{\mathrm{FR}} \\ & \quad \Leftrightarrow \; 0 = \left(u_{2,\max} \cdot \mathbf{p}_{3}^{\text{▬}} - \mathbf{p}_{2}^{\text{▬}} \right) \cdot \begin{pmatrix} \mathbf{R} & \mathbf{n}_{\mathrm{W}}^{C} \\ \mathbf{0} & 1 \end{pmatrix} \cdot \left(\tfrac{d_{x_1}}{2}, -\tfrac{d_{x_2}}{2}, -\tfrac{d_{x_3}}{2}, 1 \right)^{\top} \; . \end{aligned} \tag{7.2.2}$$

Herein, $\mathbf{p}_{\square}^{\text{▬}}$ are the respective row vectors of **P**. From this mathematical relationship it can be seen, that equation 7.2.2 depends linearly on the 3D parameters given by $\mathbf{n}_{\mathrm{W}}^{C}$, **d**, and all elements of **R**. Therefore, several constraints can be formulated under the basic assumption that the backprojection of the 3D bounding box into the image plane fits tightly into the 2D bounding box determined by the detection model. Provided that **P** is known from camera calibration, it is possible to directly solve for the 3D bounding box parameters. However, it is not possible to determine all 3D parameters in this way. This is firstly because the system of equations is under-determined and secondly only the elements of the rotation matrix **R** but not the actual orientation angles enter equation 7.2.2 linearly. Therefore, the following paragraph will examine additional assumptions that allow for a simplified 3D reconstruction to ensure that the system of equations is indeed solvable.

Towards this, note that some 3D parameters are of subordinate relevance in any practical automotive application whereas other parameters can directly be derived from the learned model and do not require a constraint-based reconstruction. For instance, the observed vehicles ϕ and ϑ angles can be approximated to $\phi = \vartheta \approx 0$ without many practical implications, since in typical traffic scenes vehicles are usually oriented in an upright pose. This simple assumption reduces the number of unknowns by two. Similarly, an assumption can be made that all existing vehicles are on a horizontal, level road surface. Therefore, the vertical vehicle position is aligned with the road surface, so that $n^C_{x_3} \approx {}^{d_{x_3}}/_2$ follows for the 3D bounding box centroid, which again reduces the number of unknowns by one. Furthermore, the observed vehicle's height d_{x_3} adds little information for subsequent driving functions like path planning or collision avoidance. Thus, an average value can be assumed for this parameter $d_{x_3} \approx \bar{d}_{x_3}$ without affecting the viability of the obtained reconstruction. It should be noted, however, that especially the assumption of coplanar vehicle positions on a common road surface is sometimes violated in real traffic scenes, which becomes more relevant the further away a detected vehicle is located [Ans+18].

Five unknowns still remain, which means that the resulting system of equations is yet under-determined. Therefore further analyses are necessary to find a solvable system of equations. Also, the considerations have so far neglected that the required world-image correspondences depend on the vehicle pose. The following explanations therefore first of all supplement how the estimation of additional auxiliary variables can serve to obtain a fully determined system of equations. This is followed by a section dealing in detail with a case discrimination that determines point correspondences based on the viewing direction of the camera.

Definition and prediction of auxiliary regressands

For the above-mentioned estimation of additional auxiliary regressands, it is appropriate to investigate, which additional information about detected vehicles can be determined directly through a learned CNN. More precisely, it must be considered which of the sought variables have particularly significant effects on the visual appearance of an object and therefore on the CNN feature maps. Based on [Mas+16], it can be established that the visual appearance of vehicles varies greatly depending on their orientation. However, the determining parameters for this are not the previously considered orientation angles ϕ, ϑ, and ψ, but instead the so-called observation angles. Since ψ remains as the only unknown orientation angle, the relevant relationship is illustrated in Figure 7.6 for the BEV. Here, the observation angle corresponding with ψ is denoted as α and β is termed as the position angle.

Note, that β is naturally estimated from the 2D bounding box with high accuracy. As introduced by [LM11], the 2D midpoint of the bounding box is projected into a horizontal plane 1 m above the ground using inverse perspective mapping (see section 3.2), β is then estimated through trigonometry from the obtained coordinates, measured relative to the camera, as follows.

$$\beta = \arctan\left(\frac{x_1}{x_2}\right) \tag{7.2.3}$$

Thus, the integration of the observation angle α as an auxiliary regressand and the

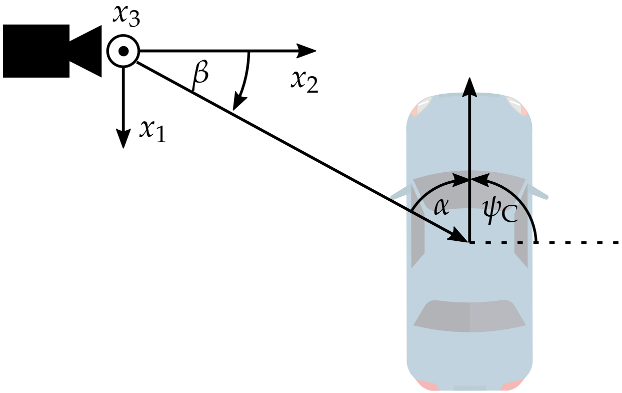


Figure 7.6: Definition of the yaw angle ψ_C, observation angle α, and position angle β in the BEV (top view), measured relative to the camera coordinate system. The left symbol depicts the test platform with the camera system, the right vehicle indicates the detected object.

estimation of the position angle β using the aforementioned procedure also allows for direct reconstruction of the objects yaw orientation ψ.

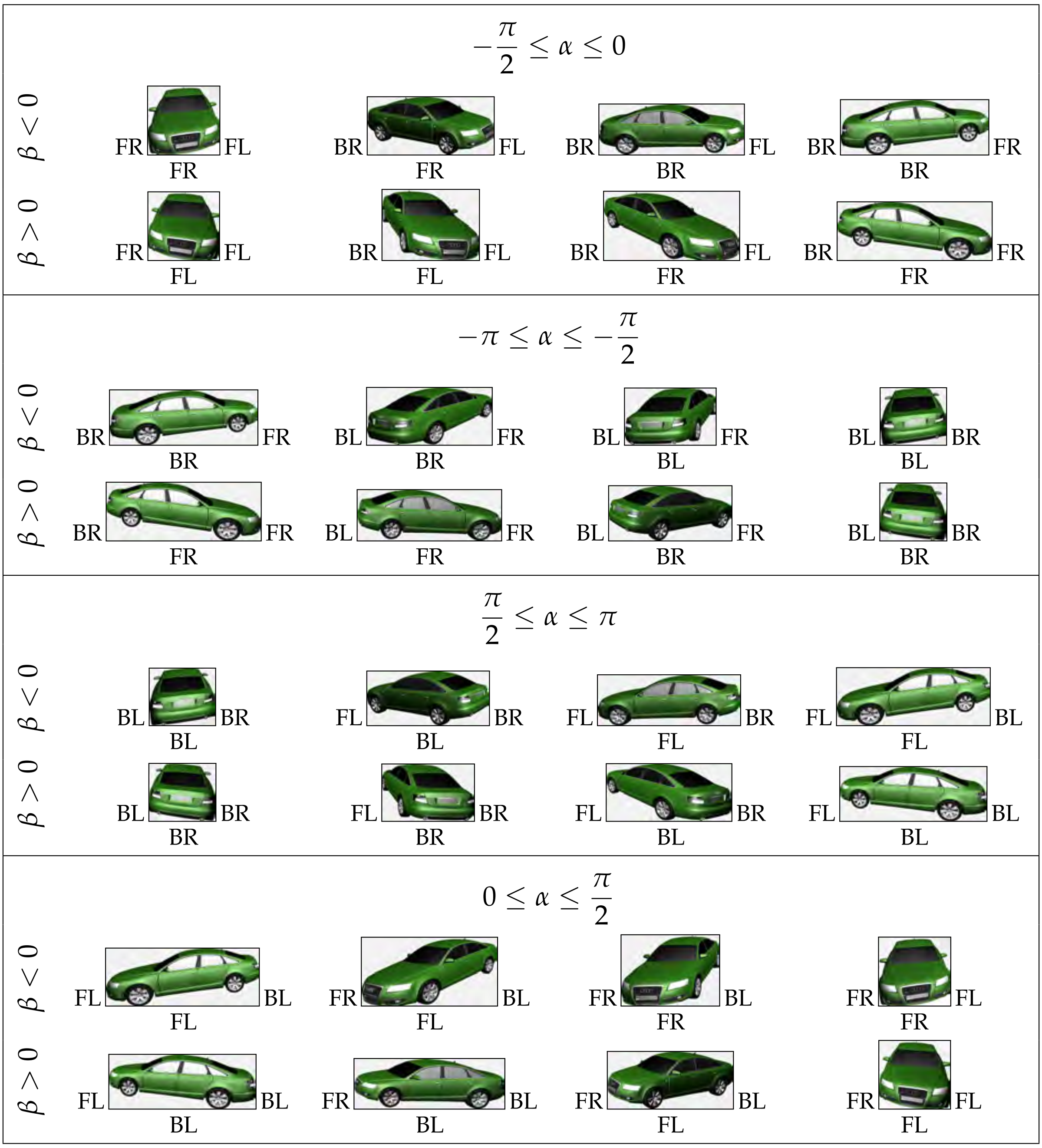
$$\psi_C = \pi - \alpha - \beta \tag{7.2.4}$$

In addition to the vehicle orientation, [Mou+17] shows that especially the vehicle dimensions can be derived from the visual appearance. This follows from the fact that within certain vehicle classes, such as station wagons, small cars, or vans the exterior dimensions vary only slightly between different models. Instead of a direct prediction of the vehicle dimensions, the corresponding aspect ratios are considered in the following. This design choice is based on the idea that the aspect ratios are more characteristic of certain vehicle classes and additionally a remaining degree of freedom to distinguish between large and small vehicles of a given class can be maintained.
Furthermore, the dimension ratios can be directly integrated as additional regressands instead of an explicit classification of vehicle classes. This way, a definition of vehicle classes as well as the manual annotation of additional data labels supplementing the 3D bounding boxes can be omitted, which simplifies the overall approach. Since the vehicle's height d_{x_3} was already determined, the ratio is formed of the remaining dimensions $\varrho = {}^{d_{x_2}}/_{d_{x_1}}$ and this value is included as an auxiliary regressand.

Correspondences

The considered correspondence problem consists of assigning the nodes of the back-projected 3D bounding box to the edges of the 2D bounding box. If no further restrictions are formulated, the number of possible combinations is 8^4. However, the previous assumptions made to simplify the linear equation system also allow to narrow down the possible node-edge correspondences. From the restriction of the 3D orientation to an upright position aligned to the road surface, it can be concluded that the lower edge of the 2D bounding box can only correspond to the bottom nodes of the 3D

Table 7.1: Case discrimination to establish the node-edge correspondences required for the constraint-based reconstruction. α denotes the observation angle and β denotes the position angle. Note, that this table assumes an upright camera and a definition of the angles as in Figure 7.6, e.g. measured relative to the camera coordinate system.



bounding box. An analogous restriction can also be formulated for the upper edge of the 2D bounding box. However, the previous simplifications for the vertically oriented 3D parameters, among other things, neglect the reconstruction of the vehicle height. Due to this, the assumption of a tight fit for the upper edge of the 2D bounding box no longer seems appropriate. Therefore, only the remaining edges of the 2D bounding box should be assigned to the bottom nodes of the 3D bounding box so that now only 4^3 possible configurations have to be distinguished.

For the remaining node correspondences, further restrictions can be made by examining which sides of the vehicle can face the camera at the same time. As an example, consider the configurations from Figure 7.5 where $u_{2,\max} \rightarrow$ FR applies. Then, the FL node can be excluded from the possible correspondences of $u_{1,\min}$. Generally, the correspondences depend on the vehicle's observation angle α and the position angle β. A respective estimation of these angles can readily be obtained as outlined in the previous section. However, the correspondences also depend on the vehicles dimensions $\mathbf{d}$, which is evident for example from equation 7.2.2. These are yet unknown, which is the reason why the correspondences can be obtained only with some remaining ambiguity. However, based on the known position angle β and observation angle α of the detected vehicle, a meaningful case discrimination can be established. For this, eight different cases can be distinguished in which the vehicle rotates between two configurations that are aligned with the viewing direction. The relationships are shown in Table 7.1. The proposed approach then evaluates all possible correspondences according to this case distinction in analogy to equation 7.2.2. Following [Mou+17], the resulting four sets of correspondences are evaluated for each detected object and the backprojection error is calculated accordingly. The underlying assumption is, that wrong assignments result in a 3D bounding box whose backprojection violates the tight fit constraint. The final selection of the reconstructed 3D parameters is therefore determined on this basis. In comparison to the rest of the processing chain, this evaluation of all possible correspondences based on the linear least squares method does not introduce significant additional computational effort. This applies all the more since the solutions for the different resulting systems of equations can be predetermined offline.

Network architecture integration

For further clarification, the integration of the overall processing chain for spatial object detection is considered. A corresponding illustration is shown in Figure 7.7. Here,

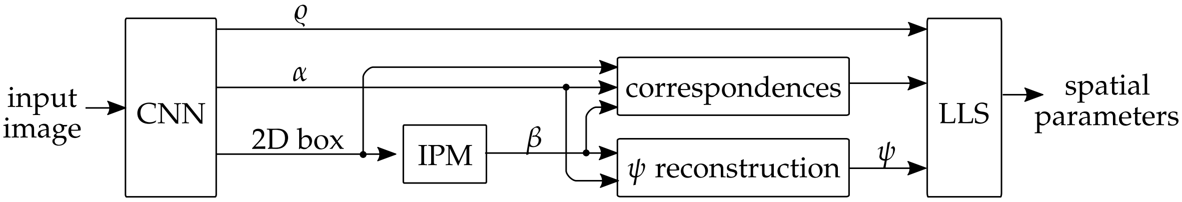


Figure 7.7: Illustration of the spatial object detection pipeline. Besides the CNN, the inverse perspective mapping (IPM) module, the node-edge correspondences, the reconstruction of the yaw angle ψ, and the final linear least squares (LLS) step are depicted.

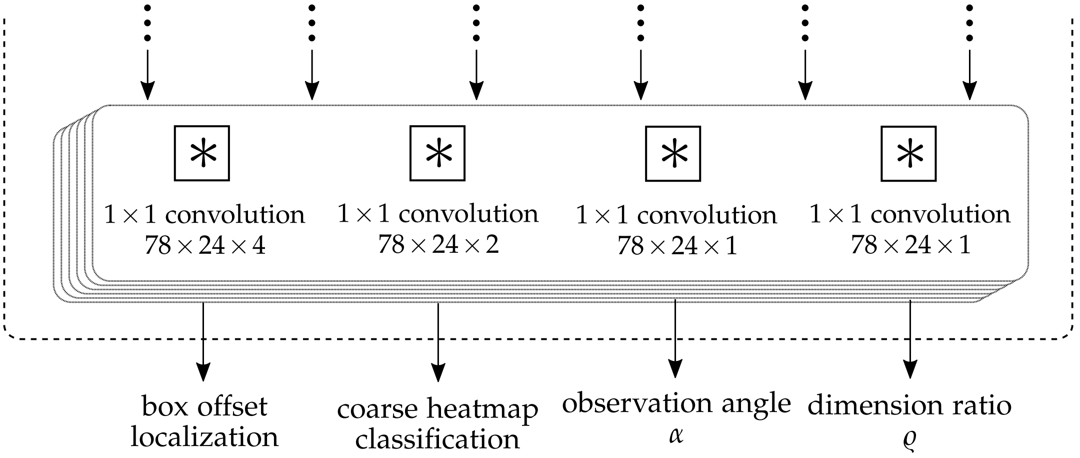


Figure 7.8: Excerpt from the modified detection decoder (the full illustration can be found in the appendix in Figure A.6). Here, the additional convolutional layers for the direct prediction of the auxiliary regressands α and ϱ are shown. These are determined through direct regression based on the feature maps and thus based on the visual appearance of the vehicle.

the interaction of the individual steps of the previously outlined method for spatial parameter reconstruction is illustrated once again. In addition to the 2D bounding boxes, the CNN also generates the estimates of the auxiliary regressands α and ϱ. The 2D position of the bounding box is used to estimate the position angle β through inverse perspective mapping (IPM, see section 3.2). The yaw angle ψ is reconstructed from α and β. Furthermore, the necessary correspondences to establish the linear system of equations are determined based on Table 7.1. Subsequently, this system of equations is solved using the linear least squares (LLS) method and taking the dimension aspect ratio ϱ into account.

For the combination of the outlined object detection and reconstruction with the feature encoder, the integration into a fully end-to-end trainable network architecture is again considered. For this purpose, the detection decoder according to Figure 7.4 is first extended by a prediction of the previously discussed auxiliary regressands. This adapted bounding box decoder architecture is illustrated in Figure 7.8. In the SSD detection method, simple 1×1 convolutions perform classifications and anchor offset localizations of 2D bounding boxes. Building upon that, another 1×1 convolution is added whose outputs are fitted to the ground truth observation angle α. Thus, instead of performing a discrete classification followed by a continuous offset regression as in the case of bounding box locations, a direct regression infers the observation angles. This is motivated by the works of [Mou+17], which argues that the viewpoint distribution in traffic scenes is generally less diverse in comparison to standard applications, which limits the benefits of a discrete-continuous approach.

Similarly, the ratio of the object dimensions can be determined by another 1×1 convolutional layer. Since, as shown in the previous section, an explicit classification of vehicle categories would cause additional annotation effort and also only a very limited range of values results for ϱ, an estimation through direct regression is used again. For the auxiliary variables, the regression loss according to equation 2.3.9 is used.

Since the definition of $\alpha \in [-\pi/2, \pi/2[$ involves a discontinuity, the regression loss can however not be applied directly. Instead, it needs to be adapted to explicitly take the discontinuity of α into account. Accordingly, the loss function for the detection decoder is defined as follows:

$$\begin{aligned} &L_{\text{Det}}\left(\mathbf{Y}_{\text{Cls},j}, \mathbf{H}_{\text{Cls},j}, \mathbf{Y}_{\text{Loc},j}, \mathbf{H}_{\text{Loc},j}, \mathbf{Y}_{\varrho,j}, \mathbf{H}_{\varrho,j}, \mathbf{Y}_{\alpha,j}, \mathbf{H}_{\alpha,j}\right) = \\ &L_{\text{2Dbox}}\left(\mathbf{Y}_{\text{Cls},j}, \mathbf{H}_{\text{Cls},j}, \mathbf{Y}_{\text{Loc},j}, \mathbf{H}_{\text{Loc},j}\right) + L_{\varrho}\left(\mathbf{Y}_{\varrho,j}, \mathbf{H}_{\varrho,j}\right) + L_{\alpha}\left(\mathbf{Y}_{\alpha,j}, \mathbf{H}_{\alpha,j}\right) \quad , \end{aligned} \tag{7.2.5}$$

where L_{2Dbox} is defined as in equation 7.1.2 and the added loss components are given as:

$$L_{\varrho}\left(\mathbf{Y}_{\varrho,j}, \mathbf{H}_{\varrho,j}\right) = \frac{1}{N_{\text{Det}}} \sum_{v=1}^{N_{\text{Det}}} \begin{cases} 0.5 \cdot \left(y_{\varrho,j}(\mathbf{u}_v) - h_{\varrho,j}(\mathbf{u}_v)\right)^2, & \text{if } \left|y_{\varrho,j}(\mathbf{u}_v) - h_{\varrho,j}(\mathbf{u}_v)\right| < 1 \\ \left|y_{\varrho,j}(\mathbf{u}_v) - h_{\varrho,j}(\mathbf{u}_v)\right| - 0.5, & \text{else} \end{cases}$$

$$\begin{aligned} &L_{\alpha}\left(\mathbf{Y}_{\alpha,j}, \mathbf{H}_{\alpha,j}\right) = \\ &\frac{1}{N_{\text{Det}}} \sum_{v=1}^{N_{\text{Det}}} \begin{cases} 0.5 \cdot \min\limits_{q} \left|y_{\alpha,j}(\mathbf{u}_v) - h_{\alpha,j}(\mathbf{u}_v) + q\right|^2, & \text{if } \min\limits_{q} \left|y_{\alpha,j}(\mathbf{u}_v) - h_{\alpha,j}(\mathbf{u}_v) + q\right| < 1 \\ \min\limits_{q} \left|y_{\alpha,j}(\mathbf{u}_v) - h_{\alpha,j}(\mathbf{u}_v) + q\right| - 0.5, & \text{else} \end{cases} . \end{aligned} \tag{7.2.6}$$

Herein, $\mathbf{Y}_{\varrho}, \mathbf{H}_{\varrho}$ denote the targets and output features for the prediction of the dimension ratio ϱ and $\mathbf{Y}_{\alpha}, \mathbf{H}_{\alpha}$ the respective targets and outputs for the observation angle α. Furthermore, $q \in 2\pi \cdot \mathbb{Z}$ in conjunction with the $\min(\square)$ function accounts for the discontinuity of α in the corresponding part of the loss function, N_{Det} is again the number of anchor boxes that overlap with ground truth bounding boxes, and $\mathbf{u}_v$ denotes their respective feature map coordinates.

To solve for the actual 3D bounding box parameters, the remaining unknowns must first be specified. Since the yaw orientation ψ was already reconstructed through other means, and since solving for ψ is disadvantageous anyway due to the nonlinear expressions in the rotation matrix $\mathbf{R}$, the existing prediction of ψ is maintained. The remaining parameters are $d_{x_1}, d_{x_2}, n^C_{x_1}$ and $n^C_{x_2}$. Since $d_{x_2} = \varrho \cdot d_{x_1}$ applies, the system of equations is set up based on the described constraints and solved for the unknowns $d_{x_1}, n^C_{x_1}, n^C_{x_2}$. For the sake of convenience, the symbolic computer algebra system published in [Meu+17] is used in the practical implementation.

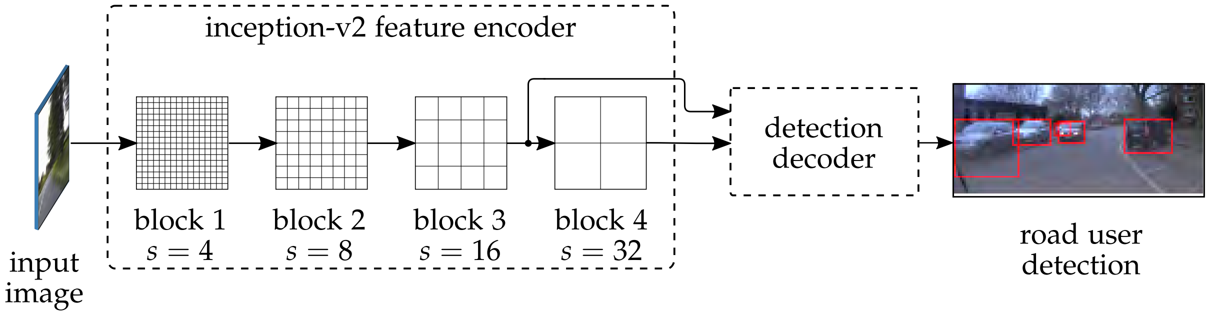


Figure 7.9: Overview of the end-to-end architecture for the detection of other road users. Two feature maps of the encoder are used to determine the 2D boxes and auxiliary regressands.

An overview of the integrated end-to-end network architecture is shown in Figure 7.9. It can be seen, that two of the intermediate feature maps of the inception-v2 encoder are evaluated as input for the decoder stage. Further lower-resolution features are generated internally within the decoder stage through sub-sampling. The resulting object representation is then derived for coarse-resolution object classification, fine-resolution offset regression, and the corresponding auxiliary regressands.

7.3 Object detection and reconstruction experiments

In this section, the presented approach towards vehicle detection and their respective 3D reconstruction is evaluated. Furthermore, the results will be related to the comparative results of alternative approaches known from the literature. Due to the step-by-step processing chain, the individual detection steps are evaluated separately. Furthermore, the same optimization setup as in section 5.3 is used for the experiments.

Benchmark datasets and evaluation protocol

The analysis requires respective datasets of traffic scenes for which annotations of vehicles with bounding boxes are available. This affects both the analysis of 2D object detection and the analysis of spatial 3D reconstruction. The evaluation of the 2D detection in image space is carried out both, based on a dataset recorded with the test platform Nissan Leaf (see section 3.1) and annotated as part of the present work, and based on the public KITTI object dataset [Gei+12] for better comparability. In the case of the Nissan Leaf, $N_{\text{train}} = 2789$ training and $N_{\text{test}} = 700$ testing images were recorded in traffic scenes near the city of Dortmund and annotated with 2D bounding boxes. The corresponding dataset is again dubbed as InVerSiV dataset in the following. However, the annotation of ground truths for the 3D parameters requires a corresponding reference sensor system, which is not part of the employed test platforms.

Therefore, the evaluation of the spatial reconstruction is exclusively based on the KITTI object dataset, since for this dataset the corresponding annotations of the 3D object parameters are available. For a conclusive analysis and to be consistent with the literature, the evaluation is carried out analogously to [Gei+12; Sim+19]. Towards this, a threshold for the overlap of the predicted 2D bounding box and the ground truth 2D bounding box is first defined, which determines the detections that are counted as TP, FN and FP respectively. This threshold value is based on the IoU measure, which is evaluated for a pair of bounding boxes as follows.

$$IoU_{2\text{D}} = \tag{7.3.1}$$

$$\frac{\left(\min\left(u^y_{1,\max}, u^h_{1,\max}\right) - \max\left(u^y_{1,\min}, u^h_{1,\min}\right)\right) \cdot \left(\min\left(u^y_{2,\max}, u^h_{2,\max}\right) - \max\left(u^y_{2,\min}, u^h_{2,\min}\right)\right)}{\left(\max\left(u^y_{1,\max}, u^h_{1,\max}\right) - \min\left(u^y_{1,\min}, u^h_{1,\min}\right)\right) \cdot \left(\max\left(u^y_{2,\max}, u^h_{2,\max}\right) - \min\left(u^y_{2,\min}, u^h_{2,\min}\right)\right)}$$

Herein, u^y and u^h denote the respective coordinates of the ground truth and predicted output 2D bounding boxes. Following [Gei+12], an IoU score of 70 % is required for a correct detection, from this the corresponding TP, FP, and FN cases can be derived directly. Furthermore, a distinction is made between three difficulty levels for the evaluation. For the easy difficulty level, truncated or occluded vehicles are ignored and only those vehicles are evaluated whose bounding boxes have a minimum height of 40 px. For the medium difficulty level, bounding boxes with a minimum height of 25 px are included and for the hard difficulty level, truncated or occluded vehicles are also counted.

The 2D detection performance can then be assessed by examining the resulting mAP based on the *pre-rec* curve. In order to obtain values that are again consistent with the literature [Sim+19], note that the increment that is used to compute the mAP measure is defined differently than in the previous chapters.

$$mAP = \frac{1}{40} \sum_{rec \in \frac{1}{40}, \frac{2}{40}, \ldots, 1} pre_{\text{interp}}\left(rec\right) \tag{7.3.2}$$

Furthermore, the mean IoU_{2D} value of all ground truth bounding boxes and maximized with respect to the confidence score threshold is given to estimate the 2D localization accuracy.

For the analysis of the spatial reconstruction, firstly the reconstruction of the viewpoint and vehicle orientation is considered. For this, [Gei+12; Mou+17] define the AOS (average orientation similarity) measure, which is the product of mAP and the cosine similarity (CS). Therefore:

$$CS_{\text{interp}}\left(rec\right) = \max_{rec^{\diamond} \geq rec} \left(\frac{1}{N_{\text{Det}}} \sum_{j=1}^{N_{\text{Det}}} \frac{1 + \cos\left(\Delta\psi_j\left(rec^{\diamond}\right)\right)}{2} \right) \tag{7.3.3}$$

$$AOS = mAP \cdot \frac{1}{40} \sum_{rec \in \frac{1}{40}, \frac{2}{40}, \ldots, 1} CS_{\text{interp}}\left(rec\right) \quad , \tag{7.3.4}$$

where N_{Det} denotes the number of detected objects and $\Delta\psi_j\left(rec^{\diamond}\right)$ denotes the yaw orientation error for a given detection j. Additionally, the reconstruction of the dimension ratio ϱ is assessed for all ground truth bounding boxes using the normalized mean squared error (NMSE) as follows.

$$\varrho_{\text{NMSE}} = \frac{\sum_{j=1}^{N_{\text{train}}} \left(\varrho_j^y - \varrho_j^h\right)^2}{\sum_{j=1}^{N_{\text{train}}} \left(\varrho_j^y - \overline{\varrho^y}\right)^2} \tag{7.3.5}$$

Herein, ϱ^y is the aspect ratio ground truth, $\overline{\varrho^y}$ denotes the arithmetic mean of the ground truth, and ϱ^h is the aspect ratio predicted by the detection decoder. For the further analysis of the reconstruction of the 3D parameters, it is necessary to modify the evaluation protocol specified in [Gei+12]. The reason is that the criteria set out in [Gei+12] are geared towards approaches combining cameras and other environment sensors such as LIDAR. In comparison, purely monocular camera-based methods do not include any active depth measurement. Thus, they operate under fundamentally

different conditions, which must be taken into account when defining the evaluation criteria. In this regard, the evaluation method of counting only those detections with $IoU > 70\,\%$ towards the TP cases does not allow a comprehensive assessment of the reconstruction performance. Moreover, a restriction of the KITTI object dataset is given by the fact, that only a fixed extrinsic calibration measured once offline is included, so that dynamic pitch ϑ and roll ϕ movements of the ego vehicle cannot be compensated. For this reason, most comparable works on monocular reconstruction of 3D bounding boxes exclude a true assessment of the actual reconstruction performance and instead focus on the AOS measure only, see for example [Gäh+18; Gui+18]. However, for the sake of completeness, the mean overlaps IoU_{BEV} and IoU_{3D} are also evaluated in the following. For this, IoU_{BEV} and IoU_{3D} are determined from the intersected and combined areas and volumes respectively, analogous to the perspective image space.
Since the ground truth annotations are not made public for the test partition of the KITTI object dataset and the official evaluation server does not provide all discussed performance measures, the described evaluation cannot be performed on this basis. Therefore, following [Xia+15], the training partition of the KITTI dataset is further divided into reduced training and validation sets, with $N_{\mathrm{train}} = 3682$ and $N_{\mathrm{val}} = 3799$ images respectively. All reported values for the KITTI dataset are therefore evaluated on this validation partition of the dataset, however, the evaluation on the test set in [Oel+18a] confirms similar results.

Performance evaluation of 2D detection and localization

When evaluating the 2D detection performance, it is first examined whether the auxiliary regressands as additional output paths of the model have significant effects on the 2D detection performance. For this purpose, a model without auxiliary regressands, corresponding to the architecture shown in Figure 7.4, is evaluated as a comparative basis. This model will be referred to as *Plain-SSD* hereafter. Furthermore, a model is evaluated, which implements the architecture for a complete 3D reconstruction according to Figure 7.8, which is termed as *SSD+AUX*. The respective results for both the KITTI object dataset and the InVerSiV dataset are presented in Table 7.2. Please note that for the InVerSiV dataset, no systematic registration of truncated or occluded vehicles was performed, so that only the difficulty levels easy and moderate are evaluated. Furthermore, only the Plain-SSD variant is evaluated on this dataset due to the available annotations.
The results show, that the additional auxiliary regressands seem to have no significant influence on the 2D detection performance. Moreover, the expected performance increase with decreasing degree of difficulty is evident as well as a generally high IoU_{2D} overlap which on average is well above the specified threshold.
For further assessment of the performance, results from other works in the literature are cited. From this analysis, the comparatively fast computation times of the proposed approach are evident. Additionally, the results indicate a generally comparable performance on par with other recent approaches. However, the proposed approach seemingly does not benefit from relaxing the test criteria in the easier difficulty levels as much as other approaches.

Table 7.2: Comparison of the 2D bounding box detection performance metrics of the SSD+AUX and Plain-SSD models, evaluated on the test/validation splits of the used datasets. The runtimes stem from the publications or, if no publication was specified, were measured with a desktop GPU (see Table A.2.d).

	InVerSiV dataset				
	mAP (hard)	mAP (moderate)	mAP (easy)	IoU_{2D} (hard)	runtime
Plain-SSD	n/a	67.48 %	85.07 %	75.16 %	25.2 ± 0.9 ms

	KITTI object dataset				
	mAP (hard)	mAP (moderate)	mAP (easy)	IoU_{2D} (hard)	runtime
SSD+AUX	71.94 %	83.55 %	86.38 %	**74.73 %**	24.8 ± 1.2 ms
Plain-SSD	**72.14 %**	**83.62 %**	86.48 %	74.68 %	25.2 ± 0.9 ms
[Tei+18]	67.59 %	83.35 %	92.80 %	n/a	98.1 ms
[Gui+18]	68.79 %	82.00 %	**92.91 %**	n/a	90.0 ms

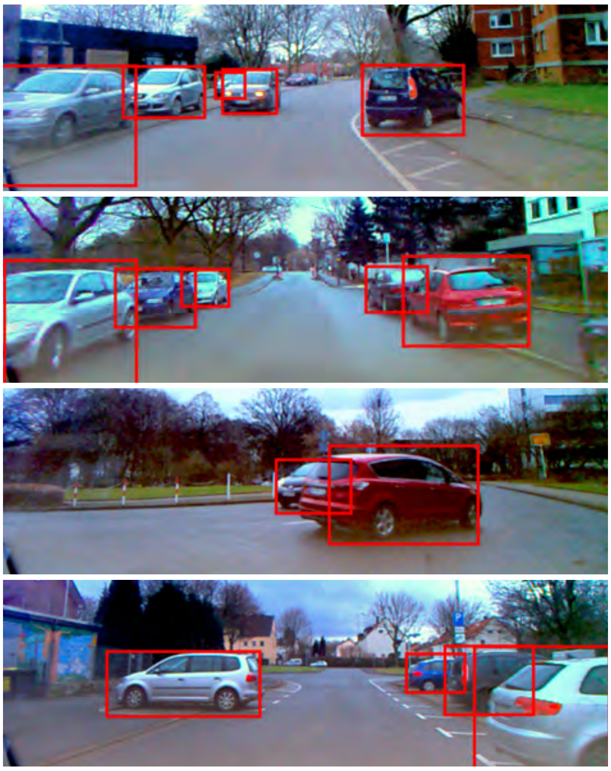


Figure 7.10.a: Example 2D detections of the Plain-SSD model on the InVerSiV test dataset.

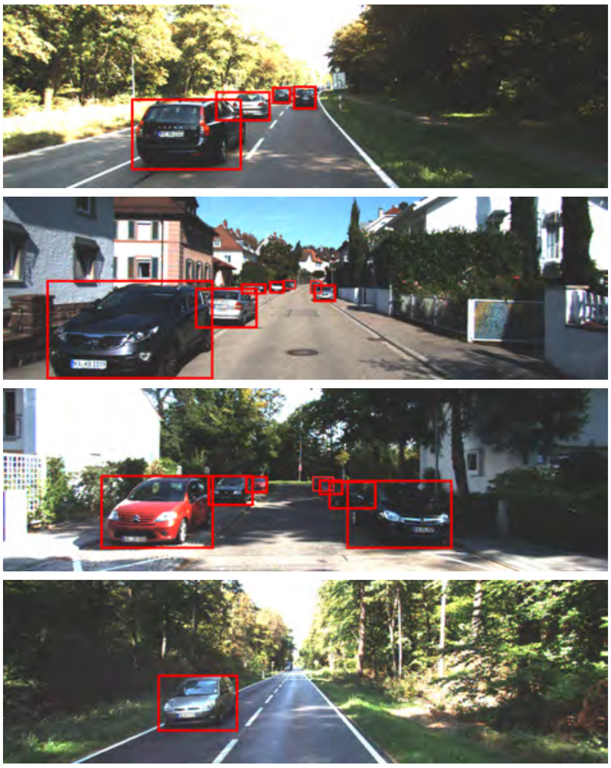


Figure 7.10.b: Example 2D detections of the SSD+AUX model on the KITTI object validation dataset.

Example detections from samples not utilized during training are illustrated in Figure 7.10.a and Figure 7.10.b. From this, it is evident, that the reliability of detections decreases with distance, which might originate from the exclusion of overly small bounding boxes in the datasets. However, a generally high performance is observable, which indicates a subjectively accurate detection of the other vehicles within the traffic scene.

Analysis and comparison of the spatial reconstruction

As discussed previously, the evaluation of the spatial reconstruction is based on different measures using the KITTI object dataset. The resulting values are given in Table 7.3. Most notable are the seemingly modest scores for the *IoU* criterias. In addition to the aforementioned shortcomings of the dataset regarding dynamic ego movements, it should also be noted that in the presented method the 3D estimation is made at the end of a multistage processing chain. Therefore, the spatial reconstruction is subject to error propagation. For example, for partially occluded vehicles or for vehicles that are only partially within the camera's area of coverage, truncated 2D bounding boxes are produced. In these cases, the tight fit assumption is violated, resulting in a corresponding deviation of the reconstructed 3D parameters. For comparison, results from the literature are again listed. For the given reasons, the evaluation of these works is however limited to the *AOS* measure. Again, the similar performance levels are on par with other state of the art approaches, however, the general conflict of objectives between computation time and performance is also evident. To this end, the approach of [Gui+18] achieves a significantly better reconstruction performance for the relaxed difficulties at the expense of a higher runtime.

In addition to the evaluation of the performance measures, an examination of individual images from the validation dataset is also carried out to enable a qualitative assessment of the results and a closer examination of individual effects and shortcomings. For this purpose, selected detections with superimposed 3D bounding boxes are shown in Figure 7.11. From this, it can first of all be seen that a predominantly plausible and apparently accurate spatial reconstruction of the detected vehicles can be deter-

Table 7.3: 3D bounding box detection performance metrics of the SSD+AUX model. The provided runtimes stem from the publications or, if no publication was specified, were measured with a desktop GPU (see Table A.2.d).

	KITTI object dataset						
	AOS (hard)	AOS (moderate)	AOS (easy)	IoU_{BEV} (hard)	IoU_{3D} (hard)	ϱ_{NMSE} (hard)	runtime
SSD+AUX	**67.83 %**	79.07 %	82.29 %	18.37 %	16.25 %	23.80 %	24.8 ms
[Gäh+18]	59.84 %	76.12 %	85.38 %	n/a	n/a	n/a	22.5 ms
[Gui+18]	67.49 %	**80.57 %**	**91.50 %**	n/a	n/a	n/a	90.0 ms

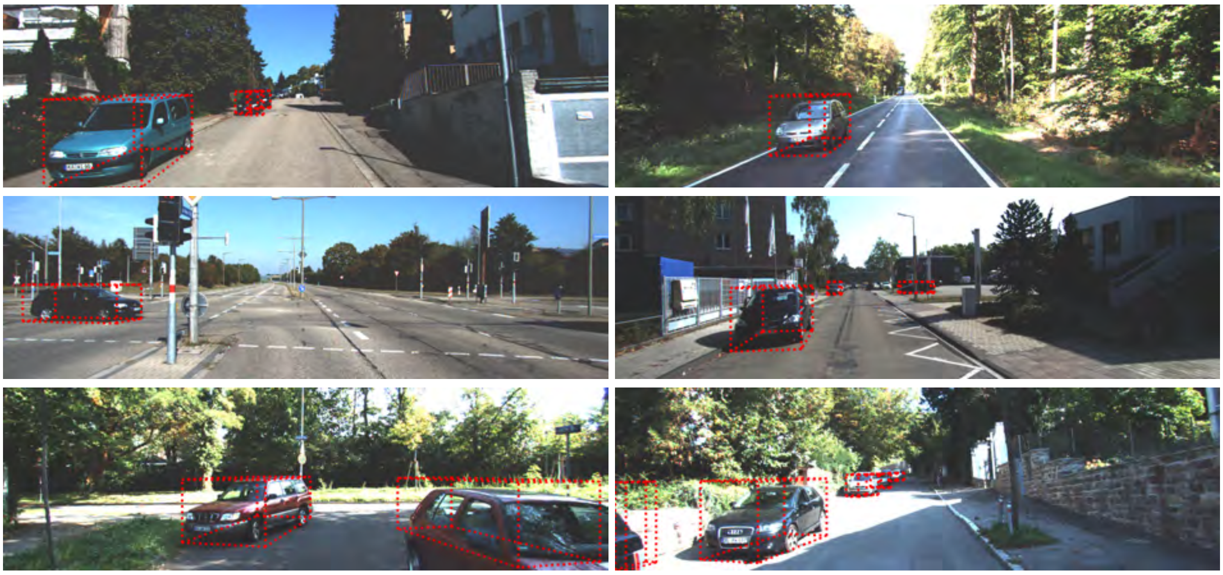


Figure 7.11: Example 3D detection boxes of the SSD-AUX model on the KITTI object validation dataset.

mined. Furthermore, it is shown that the orientation can be reconstructed with high accuracy and that no major outliers are obvious from the qualitative assessment. This positive qualitative impression is caused in particular by the enforced compliance with the backprojection constraints. However, the relatively good visual impression seems to contradict the remaining significant deviations of the reconstructed 3D bounding boxes that are evident from the IoU scores in Table 7.3. From this, it can be concluded that these deviations are primarily due to the inaccurate camera calibration with no correction of dynamic pitch ϑ and roll ϕ movements, and presumably also to the violation of the assumption of a flat road surface.

Additionally, in the bottom left example, a further fundamental effect of the investigated method is apparent. This refers to the error propagation in the case of only partially visible vehicles. In this case, truncated 2D bounding boxes are generated for which a corresponding 3D bounding box is fitted. This mechanism, due to in this case inappropriate assumptions about the tight fit constraints, potentially leads to displaced and incorrect 3D bounding boxes, as is clearly evident from the mentioned example.

8 Multi-task Integration and Conclusive Experimental Analysis

The perception tasks considered in the previous chapters already form a sufficient framework for a basic environment model. However, the CNN models discussed so far do not yet utilize the previously discussed concept of shared feature maps. Yet, due to the advantages in terms of computational efficiency, this is indispensable for the practical applicability in the used test platforms.

For this, the integration into a combined multi-task CNN is essential, as it allows better utilization of the available resources due to the elimination of repeated computations. Therefore, the subject of the present chapter is the design of an integrated CNN architecture, the discussion of an appropriate training strategy as well as a comprehensive evaluation of the effects resulting from simultaneous incorporation of the considered perception tasks.

8.1 Multi-task decoder and architecture integration

After having dealt with the methods for the individual perception tasks, the combined decoder architecture for the simultaneous generation of all environment representations of the individual tasks is discussed below. For this, the following considerations are based exclusively on the task decoders including all previously discussed adaptations. This concerns in particular the decoder for the segmentation of the drivable road area. Thus, the segmentation decoder is integrated using the variant with explicit consideration of the a-priori spatial class distribution according to Figure 6.4. Furthermore, this affects the decoder for vehicle detection, which is integrated including the auxiliary regressands that enable the spatial parameter determination, as given in Figure 7.8.

As part of the final evaluation, the investigation of possible interactions and reciprocal effects through the integration of different task combinations will be examined. Therefore, it is desirable to implement the architecture of the multi-task decoder with a modular design, which allows to activate and deactivate single tasks in a flexible way in order to investigate the different model variants. For evaluation in practical experiments, it is necessary to integrate the multi-task decoder with the inception-v2 feature

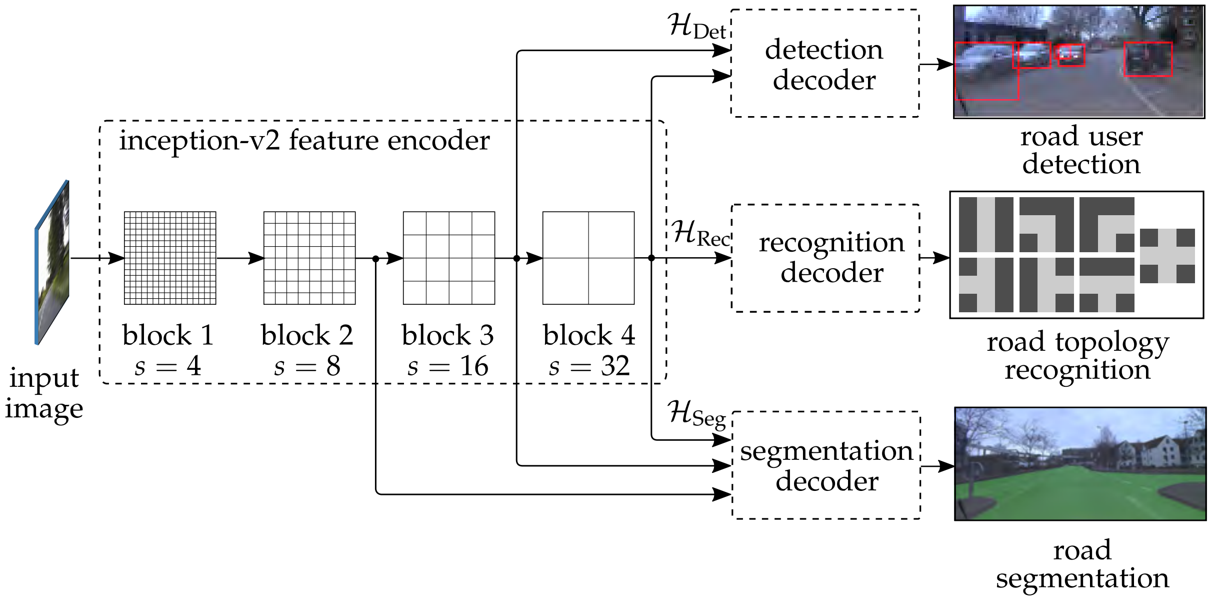


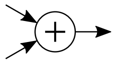
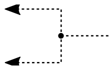
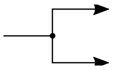
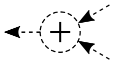
Figure 8.1: Overview of the integrated multi-task CNN architecture. For the task decoders, all previously discussed adaptations are taken into account.

encoder. Due to the choice of the hard parameter sharing approach, this integration is in close analogy to the single-task models discussed in the previous chapters. For this purpose, a single instance of the encoder is used to determine the feature maps once for all tasks. The inputs of the decoder are given as the set of all feature maps $\{\mathcal{H}_{\text{Det}}, \mathcal{H}_{\text{Rec}}, \mathcal{H}_{\text{Seg}}\}$ used for the individual tasks. This results in the same computational constraints as with the single-task models, so that ultimately the feature maps of the blocks two to four are used due to their spatial resolution. Altogether, this yields the end-to-end architecture according to Figure 8.1. Note, that the modular deactivation of individual tasks mentioned above can result in individual decoder inputs remaining unused. This also means, that not all inputs are active and receive learning signals in all conducted experiments.

8.2 Practical strategy for the joint training of all perceptual tasks

For the implementation of a multi-task architecture using the hard feature sharing approach, one has to consider the duality of the forward propagation of activation values and the backward propagation of gradient values. The basic relationship is apparent from Table 8.1. It shows, that on the one hand, a summation node in the network architecture leads to a branching of the gradient propagation. On the other hand, a branching of the network architecture corresponds to a summation of the gradients during the training phase. With this basic relationship, the learning of several tasks according to the hard feature sharing approach can be performed, provided a loss signal is available for all perception task outputs. Mathematically, the problem of model training turns into an optimization with respect to multiple objectives. In

Table 8.1: The duality of the forward inference of output predictions (solid lines) and the backward propagation of gradient values (dashed lines), see also [Ran13]. A summation of feature maps results in a branch of the gradient flow (copy) during backward propagation and vice versa.

forward inference of output predictions	backward propagation of gradient values
 sum	 copy
 copy	 sum

reference to equation 2.3.8, the total loss function thus results from a linear combination of the loss functions of the individual tasks in a weighted sum.

$$L_{\text{total}} = w_{\text{Rec}} \cdot L_{\text{Rec}} + w_{\text{Seg}} \cdot L_{\text{Seg}} + w_{\text{Det}} \cdot L_{\text{Det}} \tag{8.2.1}$$

With these considerations, the effects of a multi-task approach on the network architecture are already defined. For practical implementation, however, additional factors should be taken into account. For a start, there is the inevitable matter of the choice of weighting factors.

This significantly influences the contributions of the loss functions of the individual tasks on the gradient flow and thus on the change of the model parameters. Ultimately, the choice of the weight factors should ensure, that no task takes priority over another task. For the choice of the weights, two suitable strategies will be examined in more detail:

- Definition of constant weight factors using general heuristics
- Training the loss weight factors through optimization as (regularized) model parameters

Empirical results known from the literature indicate that often the use of general, simple heuristics already yields promising results, leaving little room for improvement to more sophisticated techniques. Examples of corresponding studies can be found in [Ser+14; EF15; Lia+16; Uhr+16; Kok17; CC17; Tei+18], and [Wan+19b].

The investigation in [Ken+18] on the explicit integration of the weighting factors as part of the model optimization provides further guidance on the applicability of both strategies. From this, it can be stated that the value ranges of the individual loss terms are decisive. To this end, [Ken+18] describes an increased model performance when using learned weight factors for specific problems where individual tasks involve the regression of unbounded quantities. Such an example is the prediction of geometric quantities, such as those that occur when estimating a depth image. As described in [Ken+18], this task even proves to be sensitive to the choice of the measurement scale (e.g. m, cm, mm, ...). However, a regression of unbounded target variables is not included in the examined perception tasks. For this reason, and also based on the known empirical findings, uniform constant factors are used to weight the individual perceptual tasks in the loss function.

Another aspect concerns the availability of annotated training data, for which some special considerations must be taken into account in the multi-task case. This is due to the fact, that combined multi-task problems have been investigated comparatively less often so that the majority of the available datasets do not devote special attention to them. Possible approaches to compensate for this fact are limited. Essentially, they are given by either the tedious process of creating the missing annotations by hand or by switching the individual tasks in each iteration and thereby training the CNN in an alternating manner. The biggest advantage of alternated training is the reduced manual annotation effort. A further advantage is the ease of a task-dependant use of data augmentation techniques [SK19]. Based on these practical considerations, an approach based on task alternated training is pursued in the following. Formally, task alternated training can be described as follows.

$$w_{\text{Rec},i} = \begin{cases} 1, & \text{if } i \bmod 3 = 0 \\ 0, & \text{else} \end{cases}, \quad w_{\text{Seg},i} = \begin{cases} 1, & \text{if } i \bmod 3 = 1 \\ 0, & \text{else} \end{cases},$$

$$w_{\text{Det},i} = \begin{cases} 1, & \text{if } i \bmod 3 = 2 \\ 0, & \text{else} \end{cases} \tag{8.2.2}$$

Herein, i denotes again the current iteration count. With this strategy, alternated training allows combining datasets for multi-task models from several single-task datasets. However, following the above formalism, an identical number of training examples is required to consistently switch between the tasks. This can be achieved by oversampling or undersampling the individual single-task datasets, with oversampling being preferable in the sense that it allows the best possible utilization of the available data.

8.3 Experimental results and comparison

The subject of this section is the final evaluation of the combined multi-task architecture according to Figure 8.1. For this purpose, the focus will be on the analysis of effects and influences, which emerge as a result of the multi-task approach in particular. To investigate potential reciprocal effects, the results of a comparative analysis of different task combinations will be evaluated. Therefore, leaning on the method of ablation studies [Mey+19], variants in which individual tasks are omitted are evaluated in addition to the full multi-task CNN.

For this, a possible positive inductive bias effect due to the additional features of different tasks will be considered. Furthermore, the opposite effects are also plausible, for instance through the effect of capacity exhaustion. The precise assessment is made in the ensuing analysis, in which the considered perceptual tasks are examined successively. Therefore, the objectives and measures for evaluating the model performance, as discussed in the previous chapters for the individual tasks, are largely retained in the following.

Compared to the procedure for the single-task experiments, a slightly different training setup is used for technical reasons. To this end, the integration of the segmentation decoder again requires the batch size to be set to $N_{\text{batch}} = 1$, so that this value is also used for other tasks. Furthermore, preliminary experiments revealed a pattern in which the vehicle detection performance for the multi-task architecture was significantly lower than in the single-task case. Presumably, this can be attributed to the task switching strategy which leads to an overall less smooth loss function that might negatively impact the optimization convergence. In this context, however, it has proven useful to compensate for this effect by initializing the CNN parameters from the trained single-task detection model (see chapter 7), contrasting with the previously used training setup. Nevertheless, the remaining configuration of the optimization solver is unchanged with respect to the single-task experiments.

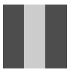
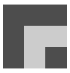
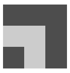
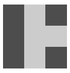
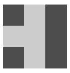
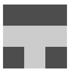
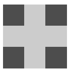
Road topology recognition

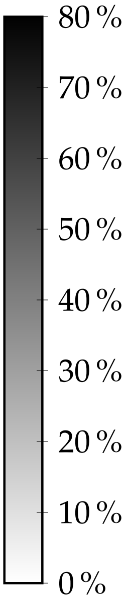
The following evaluation of the road topology recognition is carried out analogously to section 5.3. Thus, the dataset based on a subset of Cityscapes [Cor+16] and supplemented by an annotation of the road topology is retained as the basis. Likewise, the criteria and performance measures remain identical, which consist of the $F1$ and mAP measure averaged over all samples as well as the pre, rec, and $F1$ measure averaged over all classes according to equations 5.3.5, 5.3.7, and 5.3.4. Hence, two dual-task models are compared, which in addition to the topology recognition integrate the other tasks once each, as well as the complete multi-task model, introduced in Figure 8.1. An overview of the respective results is given in the following Table 8.2. These results show that the achieved performance of all compared models is generally on a similar level. The same assessment is also found when comparing the corresponding single-task results (see Table 5.1). Thus, the results indicate that the model capacity of the inception-v2 feature encoder is not a limiting factor with respect to the investigated road topology recognition. Furthermore, in comparison to the multiple execution of the single-task models, significantly fewer calculations have to be performed for roughly the same performance. This suggests, regarding the existence of shareable features, that these do indeed contribute to the intended increase in computational efficiency. At the

Table 8.2: Road topology recognition results, measured as mAP, $F1$, pre, and rec using per sample micro-averaging and per class macro-averaging. The tested model variants reflect the different task combinations.

model	micro-averaged		macro-averaged		
	mAP_μ	$F1_\mu$	pre_{M}	rec_{M}	$F1_{\text{M}}$
Rec+Seg	74.66 %	70.86 %	69.07 %	67.11 %	67.68 %
Rec+Det	**74.79 %**	71.11 %	67.14 %	**68.24 %**	67.33 %
Rec+Seg+Det	74.68 %	**71.14 %**	**69.36 %**	67.83 %	**67.92 %**

Table 8.3: Per-class road topology recognition confusion matrix with the full multi-task CNN. The numbers indicate the percentage of the corresponding classifications with respect to the total number of samples of a class. Furthermore, the numbers in parentheses indicate the absolute number of classifications.

ground truth		prediction						
	[%]	a.	b.	c.	d.	e.	f.	g.
	a.	79.7 (59)	0.0 (0)	2.7 (2)	2.7 (2)	0.0 (0)	4.1 (3)	10.8 (8)
	b.	8.0 (2)	56.0 (14)	4.0 (1)	0.0 (0)	16.0 (4)	8.0 (2)	8.0 (2)
	c.	3.6 (1)	0.0 (0)	67.9 (19)	3.6 (1)	0.0 (0)	10.7 (3)	14.3 (4)
	d.	5.7 (3)	1.9 (1)	3.8 (2)	64.2 (34)	3.8 (2)	5.7 (3)	15.1 (8)
	e.	10.5 (4)	5.3 (2)	5.3 (2)	5.3 (2)	57.9 (22)	5.3 (2)	10.5 (4)
	f.	6.5 (2)	6.5 (2)	3.2 (1)	0.0 (0)	9.7 (3)	71.0 (22)	3.2 (1)
	g.	5.0 (5)	0.0 (0)	5.9 (6)	1.0 (1)	3.0 (3)	6.9 (7)	78.2 (79)



same time, however, it must be noted that increased generalizability of the multi-task model is not apparent from the results.

Alongside the aggregated performance measures, the breakdown of the results into the individual road topology categories is again considered. For a clearer presentation, only the most relevant model variant is used here, which is given by the full multi-task model. The corresponding confusion matrix is presented in Table 8.3, note that the class definitions are identical to those from chapter 5. A direct comparison with the results of the single-task model (see Table 5.2) again reveals a highly similar pattern. Thus, the general range of the results is again similar and the largest concentration of misclassifications is found in the *FP* cases of class g (intersection). In detail, however, the class imbalance of the dataset seems to have a somewhat stronger effect, since there is a slightly greater dispersion between the maximum and minimum *TP* values. For further assessment, the confusion matrices of the dual-task models and the complete set of *pre-rec* curves can be found in the appendix from section A.7 to A.10.

Drivable road area segmentation

The evaluation of the results for the task of drivable road area segmentation is carried out according to the procedure established in section 6.3. Thus, the evaluation is based on the InVerSiV dataset in image space and BEV-space and additionally on the KITTI road dataset [Fri+13] in BEV-space only. Consequently, the *F*1 measure maximized

Table 8.4: Comparison of the pixel-based performance metrics for the task of drivable road area segmentation. The results were measured for the different variants of the multi-task architecture using the test splits of the employed datasets.

InVerSiV dataset (perspective view)					
	$F1_{\text{max}}$	*pre*	*rec*	mAP	IoU
Seg+Rec	**92.26 %**	**91.78 %**	93.79 %	91.12 %	**85.63 %**
Seg+Det	91.88 %	89.31 %	**94.61 %**	91.02 %	84.98 %
Seg+Rec+Det	91.93 %	90.86 %	93.09 %	**91.14 %**	85.06 %

InVerSiV dataset (BEV)					
	$F1_{\text{max}}$	*pre*	*rec*	mAP	IoU
Seg+Rec	**90.69 %**	**89.41 %**	**92.01 %**	**90.78 %**	**82.97 %**
Seg+Det	90.29 %	88.78 %	91.85 %	90.51 %	82.30 %
Seg+Rec+Det	90.35 %	89.23 %	91.50 %	90.57 %	82.40 %

KITTI road benchmark dataset (BEV)					
	$F1_{\text{max}}$	*pre*	*rec*	mAP	IoU
Seg+Rec	**94.81 %**	94.76 %	**94.86 %**	91.97 %	n/a
Seg+Det	94.45 %	**94.92 %**	93.99 %	91.86 %	n/a
Seg+Rec+Det	94.41 %	94.63 %	94.19 %	**91.98 %**	n/a

with respect to the decision threshold, the corresponding values of *pre*, *rec*, IoU, and also the mAP of the *pre-rec* curve are evaluated. Once again, the comparison focuses on the evaluation of the different task combinations. The corresponding results are listed in Table 8.4, a more detailed breakdown of the KITTI road results can again be found in the appendix under section A.11. Here too, the measured performance of the models is generally similar and there appear to be no prominent outliers.

On closer inspection, the Seg+Rec model achieves slightly higher performance on most measures compared to the other models. A further review of Table 6.1 reveals, that this statement can be maintained even when comparing the results of the single-task model. However, the measured differences are only marginal, so that this result is unlikely to be of systematic importance, but is rather the outcome of random processes during the model training. Altogether, no performance limitation due to an exhausted model capacity is observable for the task of drivable road area segmentation. By achieving consistent performance in the multi-task case, the increased efficiency of this approach is also again evident. Moreover, the earlier conclusions are reaffirmed, in that the increased efficiency yields a decrease in computational demands, but no improvement of the generalization capability is apparent. The results on the test partition of the InVerSiV dataset are slightly worse than for the KITTI road dataset, which is again attributed to its smaller size. Additionally, the results indicate that in the case of the InVerSiV dataset the values for *pre* and *rec* are less balanced. Thus, the multi-task models have the same tendency to-

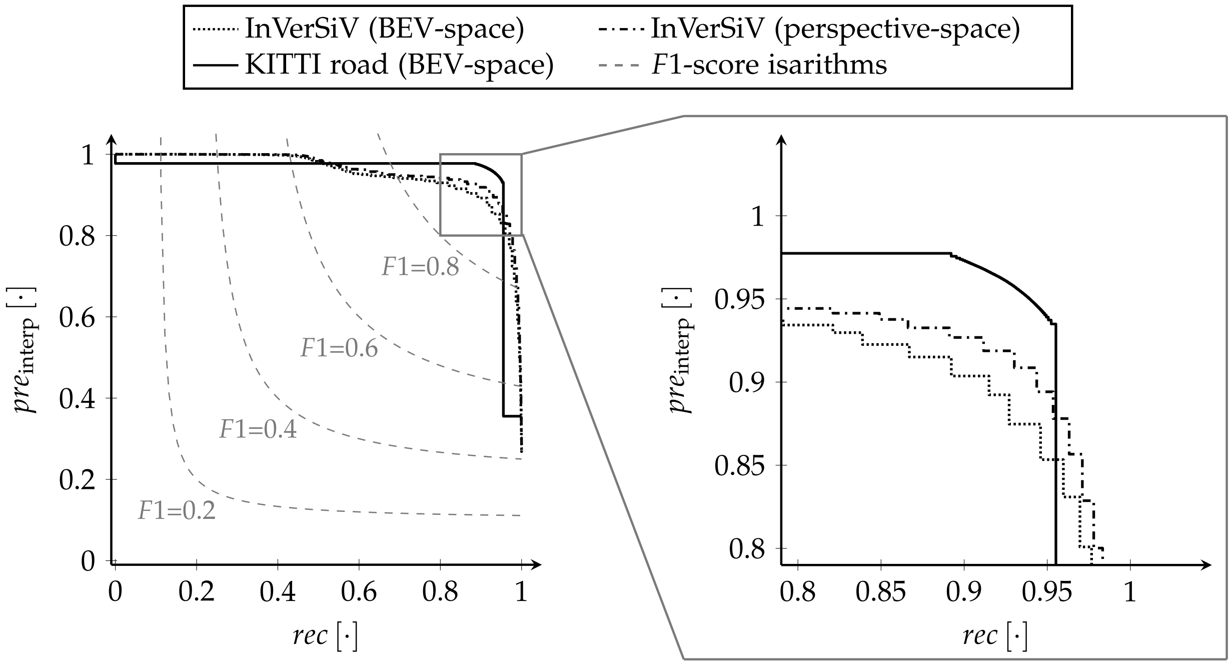


Figure 8.2: *pre-rec* curves for the Seg+Rec+Det multi-task CNN obtained from the test partitions of the KITTI road and InVerSiV datasets. The results for the KITTI road dataset were determined on the official evaluation server.

wards FP errors as the single-task variants when evaluated on the InVerSiV test data. For further analysis, Figure 8.2 shows the *pre-rec* curves of the most relevant Seg+Rec+Det model as well as the zoomed in view of the most relevant part of the curves. Similar to the single-task case, the curves are again close to the optimal, step-shaped curve. The results determined in the perspective space show a slightly superior curve in comparison. The described tendency of the model to FP errors is reflected in the curves by a slightly off-centered trend due to the early decrease of the *pre* values. As in Table 8.4, no comparable effect is observed for the results on the KITTI road dataset.

Vehicle detection and reconstruction

For the analysis of the detection of traffic objects with multi-task models the procedure according to section 7.3 serves as a reference. Therefore, the detection of vehicles through 2D bounding boxes and their respective viewpoint and full spatial reconstruction are considered. Note, that the following experiments are based on the SSD+AUX detection decoder architecture that requires 3D annotations. Therefore, the evaluation and comparison of the mAP, AOS, and IoU performance measures is based on the corresponding validation data of the KITTI object dataset [Gei+12]. The results for 2D vehicle detection for all different task combinations are shown in Table 8.5. As with the other scene representations, the performance is again comparable to that of the single-task models. In three of the four considered performance measures, the Det+Seg model

Table 8.5: 2D vehicle detection performance measures on the KITTI object validation dataset for all task combinations. All presented models use the SSD+AUX decoder stage to implement vehicle detection and reconstruction.

	mAP (hard)	mAP (moderate)	mAP (easy)	IoU_{2D} (hard)
Det+Rec	71.99 %	83.50 %	**86.42 %**	74.73 %
Det+Seg	**72.11 %**	**83.58 %**	86.05 %	**74.78 %**
Det+Rec+Seg	71.95 %	83.41 %	86.34 %	74.68 %

is slightly superior. However, the difference is within the range of general performance variations due to the non-deterministic factors in the training process. The average IoU_{2D} overlap of the predicted bounding boxes with the ground truths is slightly above the specified threshold of 70 % when considering the hard difficulty level with the largest number of samples. The consistent performance compared to the use of the single-task models confirms once again the ability to increase the computational efficiency of the simultaneous prediction of multiple environment representations due to shared features. However, the hypothesis of a further increased generalization ability through a multi-task inductive bias effect, as observed for example in [Tei+18; Kok17], can again not be affirmed.

For further analysis, the performance measures for the evaluation of the spatial reconstruction will be examined. The reconstruction of the viewpoint through the observation angle α is thereby assessed with the AOS measure. The actual reconstruction is again evaluated by determining the IoU_{BEV} values in the BEV-space and the volumetric IoU_{3D} of the full 3D bounding boxes. It should be noted in this context, that the same limitations set out in section 7.3 regarding ϑ and ϕ correction must again be taken into account when assessing the results. Table 8.6 contains the resulting values of the reconstruction performance measures. As can be anticipated from the previous analysis of the 2D performance measures, again no significant deviation from the results of the single-task model is found in the evaluation of the spatial reconstruction.

Table 8.6: 3D vehicle reconstruction performance measures on the KITTI object validation dataset for all task combinations. The SSD+AUX decoder stage is used in all models to implement the reconstruction of the vehicle orientations and 3D bounding boxes.

	AOS	AOS	AOS	IoU_{BEV}	IoU_{3D}
	(hard)	(moderate)	(easy)	(hard)	(hard)
Det+Rec	67.89 %	79.11 %	**82.37 %**	18.08 %	15.91 %
Det+Seg	**68.09 %**	**79.33 %**	82.08 %	17.76 %	15.74 %
Det+Rec+Seg	67.90 %	79.09 %	82.33 %	**18.17 %**	**15.97 %**

Noteworthy, when assessing the spatial *IoU* measures an opposite situation arises, where in contrast to the 2D performance measures the Det+Seg model now performs slightly worse than the other task combinations. However, the differences are again marginal, hence a systematic tendency cannot be concluded. In summary, after considering the vehicle detection and reconstruction task, primarily the striking gains in terms of computational efficiency due to the multi-task approach can be postulated. Yet, no further positive effect on generalizability towards unseen test samples can be observed.

Qualitative evaluation of selected test samples

In addition to the quantitative analysis of the individual perceptual tasks, a qualitative analysis through example images is conducted in the following. Since the training annotations of two of the three perceptual tasks are available for the KITTI dataset, corresponding examples from the KITTI object validation set will be examined to maintain consistent image characteristics. The obtained output representations based on the simultaneous perception by the Det+Seg+Rec model are illustrated in Figure 8.3. For better comparability, the selected example images correspond to those previously used for Figure 7.11. The visualization firstly confirms the general conclusion of an accurate perception of the depicted traffic scenes. In detail, from the middle left and bottom left examples it is noticeable, that the results of the drivable road area segmentation show remaining *FN* errors. This is obvious in the middle left example especially in the left part of the image and between the signposts. For the bottom left example, it is apparent that a large area of the road section depicted on the left side of the image could not be captured correctly.

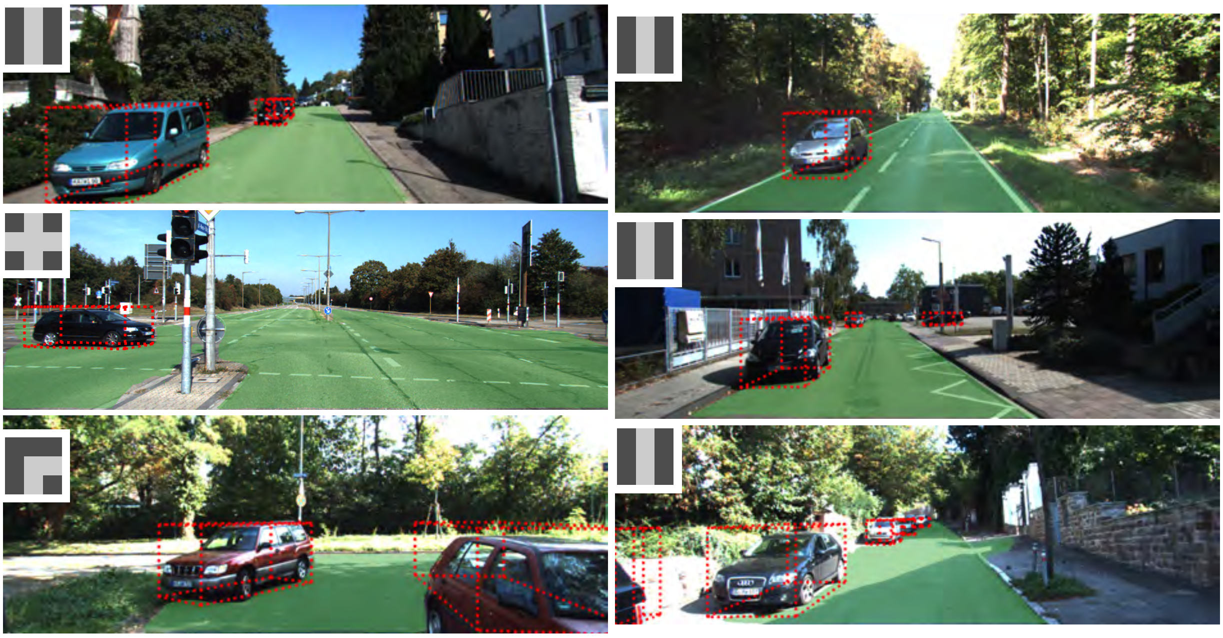


Figure 8.3: Visualization of the environment representations generated with the full multi-task CNN. The example images originate from the validation partition of the KITTI object dataset.

Concerning the detection results, there are no significant deviations from the visualizations of the single-task model (see Figure 7.11). Once again, mainly the general shortcomings of the approach based on backprojection constraints are evident for vehicles that are only partially depicted in the image. The results of the road topology recognition appear mostly plausible, except for the bottom left example. Here, the model has assigned the category of a right turn with the highest confidence, whereas for a human observer the category of a fork junction seems more appropriate. Interestingly, this example also shows the most striking errors in the drivable road area segmentation, which might hint at an underlying relation of these effects.

As another aspect of the qualitative evaluation, the transferability of the learned model towards altered operating conditions will be considered hereafter. For this purpose, example images will be considered in the following, which were recorded with the help of the second test platform road side unit in the context of the of the InVerSiV project. The altered operating conditions result from an observation of the traffic scene from a higher perspective so that the views of the various scene elements differ significantly from the views contained in the training dataset. Thus, this examination aims to investigate, whether the changed perspective already exceeds the systems generalization capabilities, or whether the perception tasks were learned with sufficient robustness so that a generally feasible scene description can still be maintained. For this purpose, corresponding images are given in Figure 8.4. Note, however, that due to the changed perspective and the fact that the environment of the road side unit test platform is given by one large paved area, no definite road topology can reasonably be determined according to the definition from section 5.3.

These results show, that despite the different perspective a mostly accurate scene description can be captured. This is especially noticeable for the segmentation of the drivable road area, which is captured correctly despite the changed spatial priors of

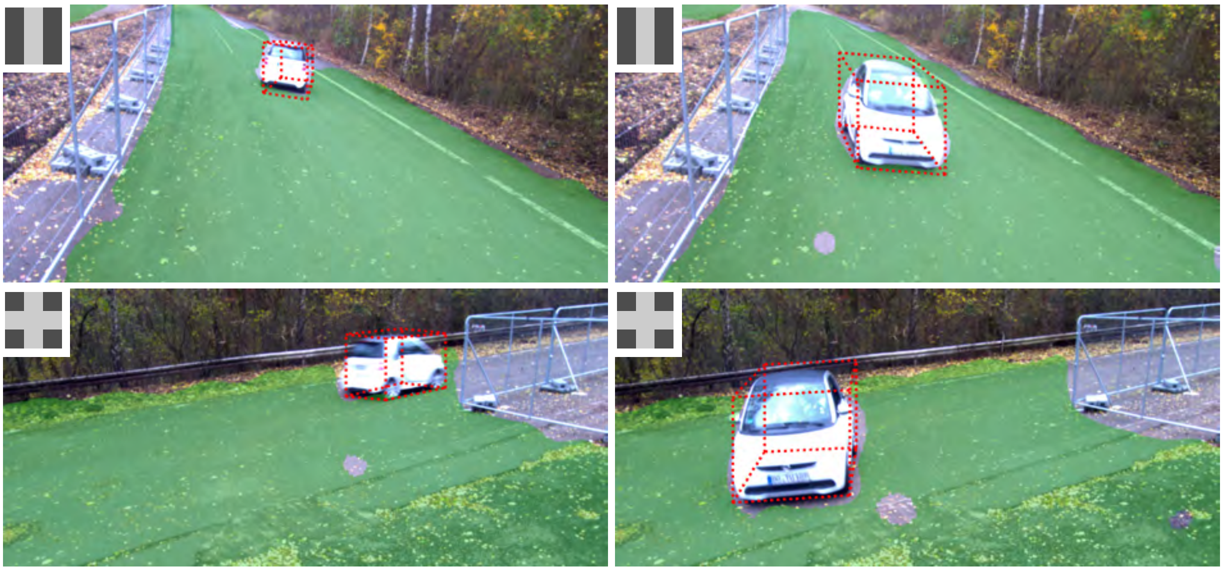


Figure 8.4: Visualization of the environment representations generated with the full multi-task model in the InVerSiV road side unit setup. Due to the altered operating conditions, the input images differ significantly from those of the used training dataset.

the class distributions. From this, it can be concluded that the generated segmentation is still largely based on the local visual appearance despite the adapted decoder architecture. Furthermore, it is noticeable that the contiguous road areas are segmented in part with some remaining gaps and that vegetation areas above the fence are captured as roads in the upper example images. Thus, in detail, a slight degradation of the segmentation can be noted compared to the previous results, which is presumably due to the different operating conditions. Furthermore, the determined road topologies appear generally plausible, despite the limitations mentioned above. In particular, in the lower example images the determination of a road topology seems subjectively difficult. Nevertheless, these images show a wide road area extending to the edge of the image, which is generally characteristic for intersection scenes and may explain the obtained topology class.

Runtime analysis on dedicated and embedded hardware

Besides the performance of the individual perception tasks, the achieved runtime of the developed approach is also of crucial importance according to the application for ADAS and automated driving systems. Hence, in the following a comprehensive analysis of the runtimes of the different task combinations is carried out. For this purpose, Figure 8.5 presents the corresponding measurements for all task configurations, which were determined through the evaluation of 100 randomly selected images. Additionally, the raw data as well as details on the spread of the measurements can be found in the appendix in Table A.12. The runtimes were measured once for common PC

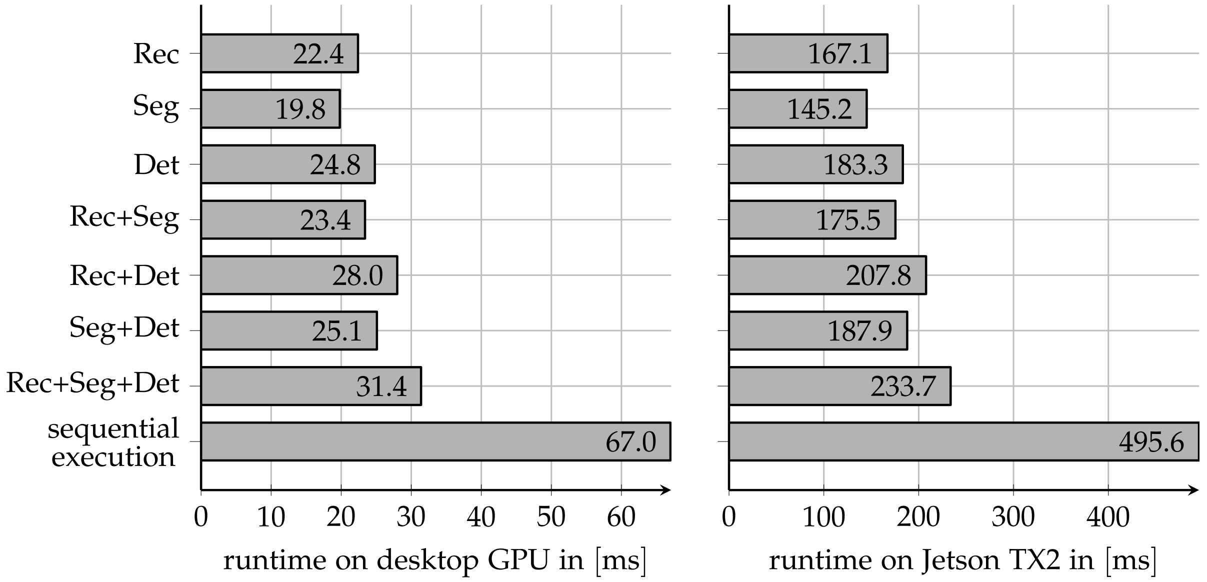


Figure 8.5: Runtimes of the single-task and multi-task CNNs compared to sequential execution, measured on a dedicated desktop GPU (see Table A.2.d) and on the embedded hardware platform Jetson TX2 (see Table A.2.c). The runtime evaluation is based on 100 randomly selected images.

hardware using a dedicated GPU as well as for the actual embedded target system of the camera platform, c.f. Figure 3.1. The comparison of the single-task results reveals, that the decoder for the task of road area segmentation is the fastest to compute, while the decoder for vehicle detection requires the most computation time. However, it should be noted that the single-task runtimes are generally similar due to the specific design of the bottleneck layers in the decoders. Moreover, the results indicate that the runtimes achieved on the PC hardware scale almost linearly to the embedded hardware, but are generally slower by a factor of about seven. The applicability of the approach is also generally evident, but it must be noted that the runtimes achieved on the embedded hardware are still slower than the commonly used control clock cycles in automotive applications. Strikingly, when considering the multi-task models, the significant increase in efficiency of the integrated approach becomes apparent. Thus, the computation of the full multi-task model requires about 53 % less runtime compared to the sequential execution of the single-task models. For this reason, the strategy of integrated perception of multiple environment representations can be regarded as an important cornerstone for the applicability of CNN-based methods on practical hardware systems. For the considered test platforms it can even be postulated, that the multi-task strategy is essential for the perception of a comprehensive environment model consisting of several complementary representations. This applies all the more, since the previous analyses did not reveal any significant deterioration in the achieved perception performance of the multi-task architecture.

9 Summary, Conclusion, and Outlook

This thesis investigates methods for traffic scene perception with monocular cameras as a foundation for a basic environment model in the context of automated vehicles. The developed perception system is designed with a special focus on the practical application in two experimental systems, which results in significant restrictions of computational resources. For this purpose, three distinct scene representations are investigated. These consist of the prevalent road topology as the global scene context and the perception of the drivable road area, which are both associated with the static environment. Furthermore, the detection and spatial reconstruction of vehicles present in a given scene is considered in order to take the dynamic aspects of the environment into account. In order to cope with the computational constraints, an approach is followed that allows the simultaneous perception of all environment representations with methods based on a multi-task CNN architecture. The implementation with a shared encoder stage and task specific decoders enables a systematic avoidance of repeated computations through shared features for all perception tasks to ease the overall computational burden.

Moreover, the approach allows to separately perform an initial examination of the individual perception tasks. For this purpose methods for the respective tasks are first developed independently and adapted to the special conditions of traffic scenes. Here, the recognition of the road topology is realized as general image recognition. In addition, the perception of the drivable road area is implemented as image segmentation. To this end, an approach based on the FCN architecture [She+17] is adapted to improve the incorporation of the a-priori class distribution present in traffic scenes. This is achieved through the inclusion of element-wise weight factors through the Hadamard product, which resulted in increased segmentation performance in the conducted experiments. Also, a task decoder for the perception of other road users is designed based on the compact SSD method for 2D detections according to [Liu+16a], which is extended by auxiliary regressands. These are used for the appearance-based estimation of the orientation and dimension ratio of detected objects. Together with a subsequent method for the reconstruction of spatial object parameters based on constraints derived from the back projection into the image plane, a scene description with all measurements for a basic environment model and subsequent ADAS and automated driving functions can be generated. From the examination of alternative multi-task approaches and considering the computational restrictions of the experimental systems, an integrated CNN architecture is implemented, which combines all perceptual tasks in a single end-to-end trainable model. In addition to the definition of the architecture, a strategy is discussed in which alternated training of the perception tasks, changing with each iteration, enables simultaneous learning from several single-task datasets in one optimization process. On this basis, a final experimental evaluation is performed in which a systematic analysis of different task combinations is conducted and the computational efficiency of the integrated multi-task architecture is demonstrated.

In conclusion, it can be stated that the incorporation of the specific properties of traffic scenes into the perception system can often offer potential for improvement or a simplification of the employed methods. Thus, the mere adoption of the general approaches for individual perception tasks is not advisable with regard to the goal of appropriate perception performance. Instead, a practical implementation requires a systematic consideration of application-specific adaptations. This can be seen, for example, from the evaluations in section 6.3 concerning the inclusion of the a-priori class distribution for the segmentation of the drivable road area through element-wise weights. Furthermore, the considerations from chapter 7.2 also confirm this conclusion, which discuss the simplifying assumptions possible in traffic scenes regarding the orientation and other spatial parameters of detected vehicles. In particular with respect to the spatial reconstruction of detected road users, however, the experimental assessment also reveals the remaining shortcomings of the implemented method. This refers in particular to the degradation of the estimated spatial parameters in case of a violation of the tight fit assumption of the backprojection into the image plane. In the course of the evaluations, it became obvious that this case is a common error in real traffic scenes. Furthermore, the strong sensitivity of the reconstruction based on the geometry of the camera model with respect to current and precise ego calibration data can be considered as a shortcoming. In light of this, the breakdown of the object reconstruction into appearance and constraint-based estimates could eventually be reconsidered.
Furthermore, the obtained results clearly show the importance of a combined approach to the perception tasks for automotive applications. Thus, the investigated multi-task CNN makes it possible to utilize existing synergies in order to control the computational complexity of the system. Within this context, the conducted experiments demonstrate that the integrated multi-task CNN for all relevant representations of the scene is indispensable for practical models on realistic embedded processing hardware. Regarding this, especially the existence of common, shareable image features for the perception of the individual scene representations, which are clearly evident from the results, is to be mentioned. At the same time, it should be noted that the investigations in this work cannot reproduce the effect of an additional inductive bias effect for increased perception performance solely due to the multi-task approach, which has been described previously in parts of the relevant literature [Kok17; Tei+18].

As an outlook on future work, first of all, the remaining processing steps for a fully operational environment model should be mentioned, which particularly refers to sensor data fusion. In line with the considered scene representations, this concerns the aggregation of measurements, for example as part of a tracking of the detected road users in the scene or a causal mapping of the drivable road area. Furthermore, methods for scene reconstruction based on an appearance-based depth estimation have made significant progress in the recent past, see for example [Jör+19; Din+20]. Thus, they offer a systematic way of compensating or circumventing the shortcomings of the constraint-based scene reconstruction in the case of incompletely depicted road users as well as with regard to the sensitivity to current and precise camera calibration. This could possibly further improve the performance of the perception system and a subsequent environment model in future work. Furthermore, the idea of systematically adapting the methods for the individual perception tasks to the specific conditions of traffic scenes offers further entry points for future research. This results mainly from the general machine learning guideline, that the highest possible correlation between the considered task and the loss function, which is used for the optimization of the model parameters, should be aspired. In this context, the current state of research often reveals a discrepancy between the 2D scene descriptions in the image plane, which have been investigated primarily, and the actually required spatial scene descriptions. An example could be the determination of the loss function for the segmentation of the drivable road area directly in BEV-space or a corresponding weighting of the image pixels in the image space loss function. Besides, recent advances have been made in hardware systems so that more and increasingly mature systems are now commercially available and accessible to the test platforms. For example, the costs for reference 3D sensors based on the LIDAR principle have decreased significantly, which expands the possibilities for a further focused development and evaluation of the spatial reconstruction of scene descriptions and thus offers approaches for future research. Likewise, significant further progress has been made in the meantime in processing hardware, and advances in the development of efficient CNN and encoder architectures are still emerging. Thus, for the computing hardware used in the online system of the test platforms, a successor product is now available on the market, which in some aspects has doubled computational resources, and with regard to efficient feature encoders and network architectures, for example, the works in [Yan+19; Tan+20] offer corresponding entry points for further research. In light of this, a re-evaluation of some of the implemented design decisions might become advisable for future work.

A

Appendix

A.1 Road topology dataset statistics

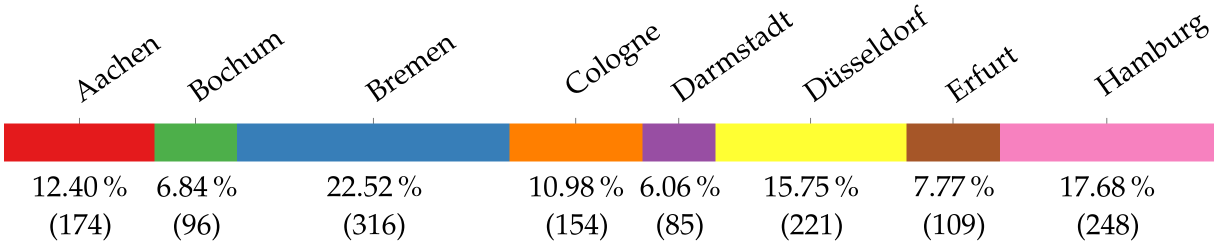


Figure A.1: Statistics of the road topology recognition dataset based on Cityscapes [Cor+16] with respect to the recording locations in eight different German cities.

A.2 Technical specifications of the camera system

Table A.2.a: Camera and sensor specifications

model	mvBlueFOX3
interface	USB3
sensor	OnSemi AR0331
type	CMOS
shutter	rolling
size	$\frac{1}{3}$ inch

Table A.2.b: Lens specifications

model	Lensagon B5M29740NDC
hor. FoV	82°
ver. FoV	61°
focal length	$f = 2.97\,\mathrm{mm}$
aperture	F4.0
optical distortion[11]	$\leq 1\,\%$

[11] The ratio between the difference of the ideal and distorted image diagonals to the ideal image diagonal is measured.

Table A.2.c: Online processing hardware specifications

model		Nvidia Jetson TX2
cores	CPU	2 × ARM A57
	GPU	256 × GP10B (Pascal)
RAM		8GB DDR4

Table A.2.d: Desktop GPU hardware specifications

model	GeForce GTX Titan X
manufacturer	Nvidia/Gainward
GPU	3072 × GM200 (Maxwell)
RAM	12 GB GDDR5

A.3 Single-task *pre*-*rec* curves for all road topologies

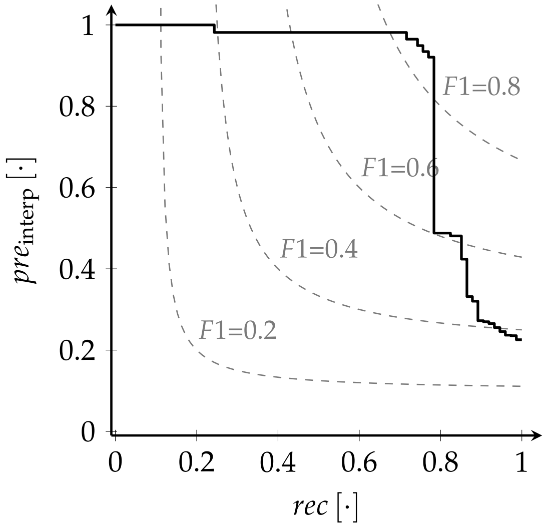


Figure A.3.a: Road topology a.: straight road

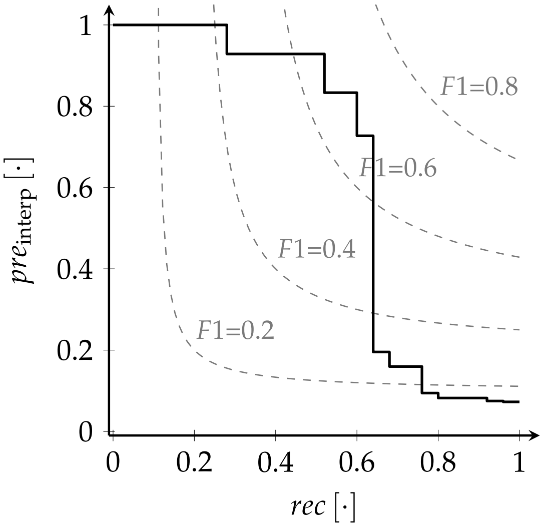


Figure A.3.b: Road topology b.: turn right

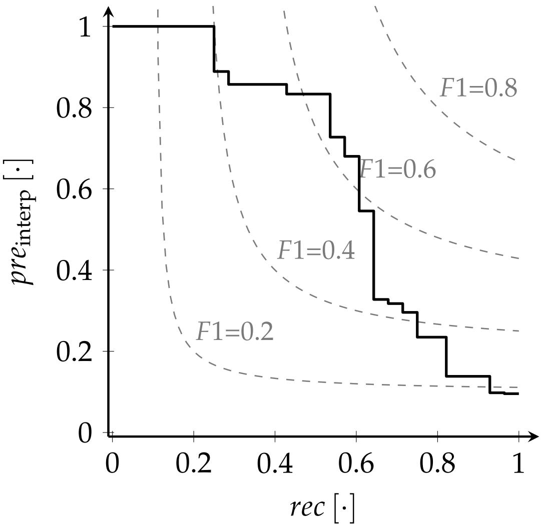


Figure A.3.c: Road topology c.: turn left

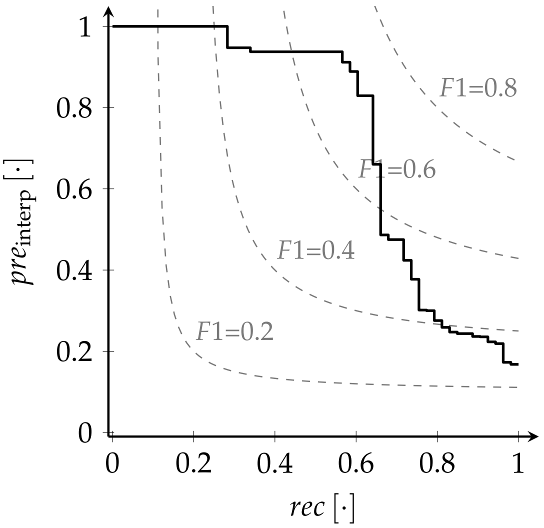


Figure A.3.d: Road topology d.: junction right

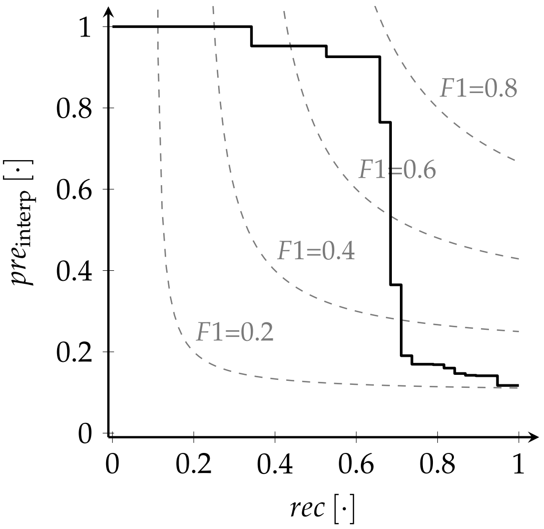


Figure A.3.e: Road topology e.: junction left

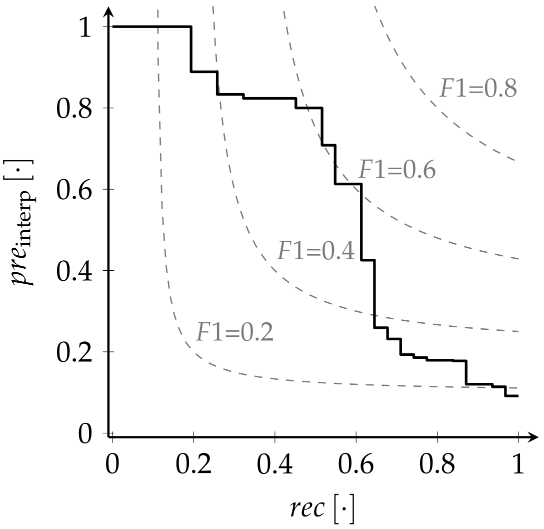


Figure A.3.f: Road topology f.: fork junction

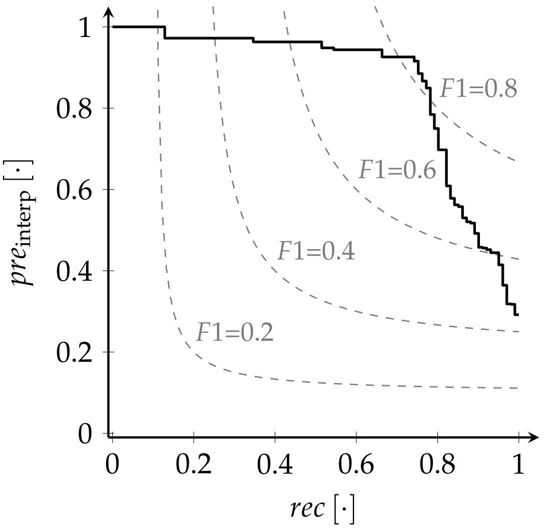


Figure A.3.g: Road topology g.: intersection

A.4 Overview of the segmentation decoder with Hadamard layer

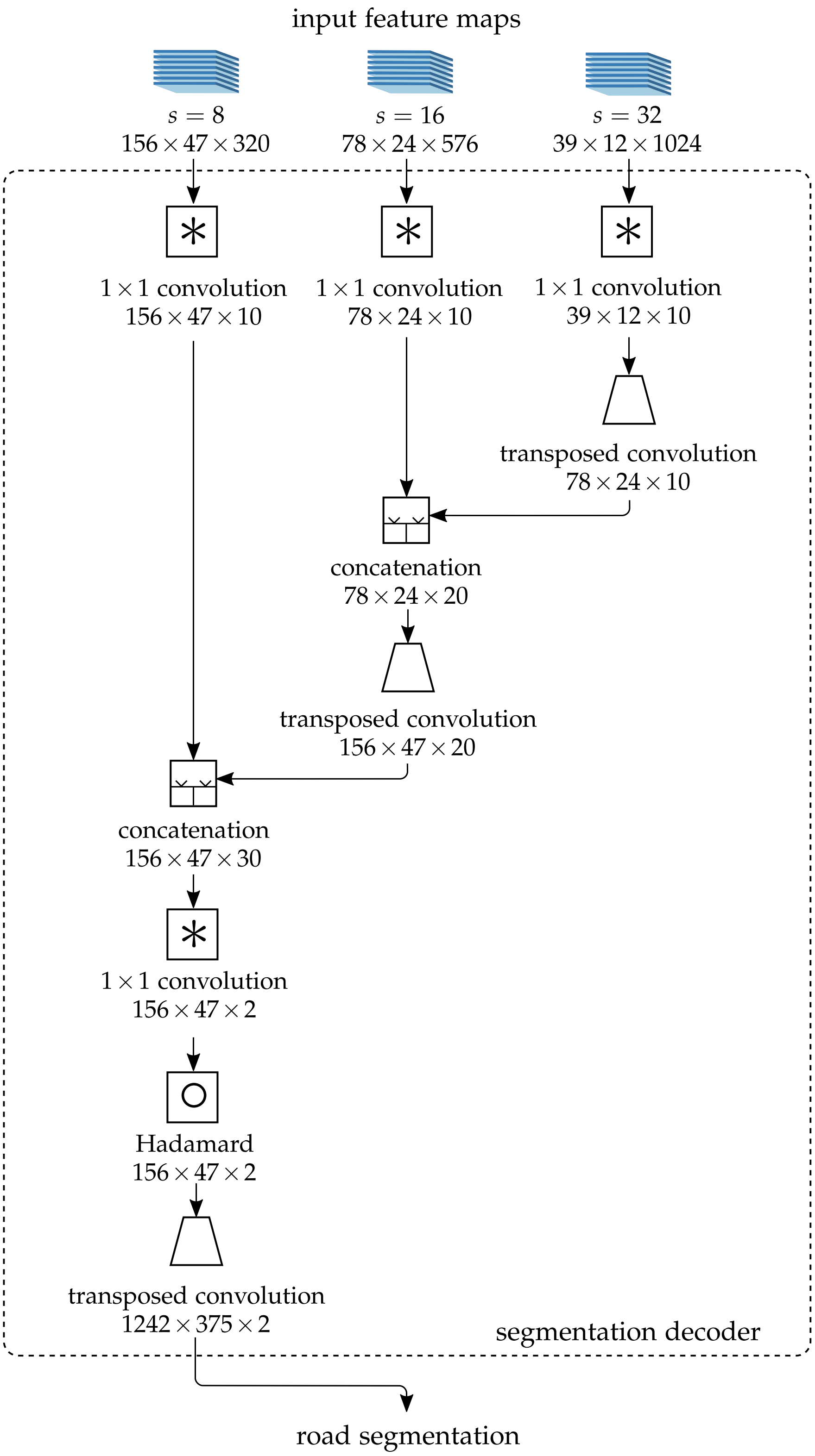


Figure A.4: Overview of the segmentation decoder with Hadamard layer.

A.5 Detailed breakdown of the single-task KITTI road segmentation results

Table A.5.a: Detailed breakdown of the single-task KITTI road segmentation results of the Hadamard-FCN into the road types defined in [Fri+13].

Benchmark	$F1_{\text{max}}$	mAP	pre	rec	FP-rate	FN-rate
UM_ROAD	94.06 %	90.89 %	94.62 %	93.50 %	2.42 %	6.50 %
UMM_ROAD	96.26 %	93.32 %	95.63 %	96.90 %	4.86 %	3.10 %
UU_ROAD	93.14 %	90.00 %	93.31 %	92.98 %	2.17 %	7.02 %
URBAN_ROAD	94.85 %	91.48 %	94.81 %	94.89 %	2.86 %	5.11 %

Table A.5.b: Detailed breakdown of the single-task KITTI road segmentation results of the Plain-FCN into the road types defined in [Fri+13].

Benchmark	$F1_{\text{max}}$	mAP	pre	rec	FP-rate	FN-rate
UM_ROAD	91.47 %	86.14 %	93.56 %	89.48 %	2.80 %	10.52 %
UMM_ROAD	95.38 %	94.26 %	95.03 %	95.73 %	5.51 %	4.27 %
UU_ROAD	87.15 %	82.36 %	88.61 %	85.74 %	3.59 %	14.26 %
URBAN_ROAD	92.26 %	91.83 %	92.80 %	91.72 %	3.92 %	8.28 %

A.6 Overview of the SSD decoder with auxiliary regressands

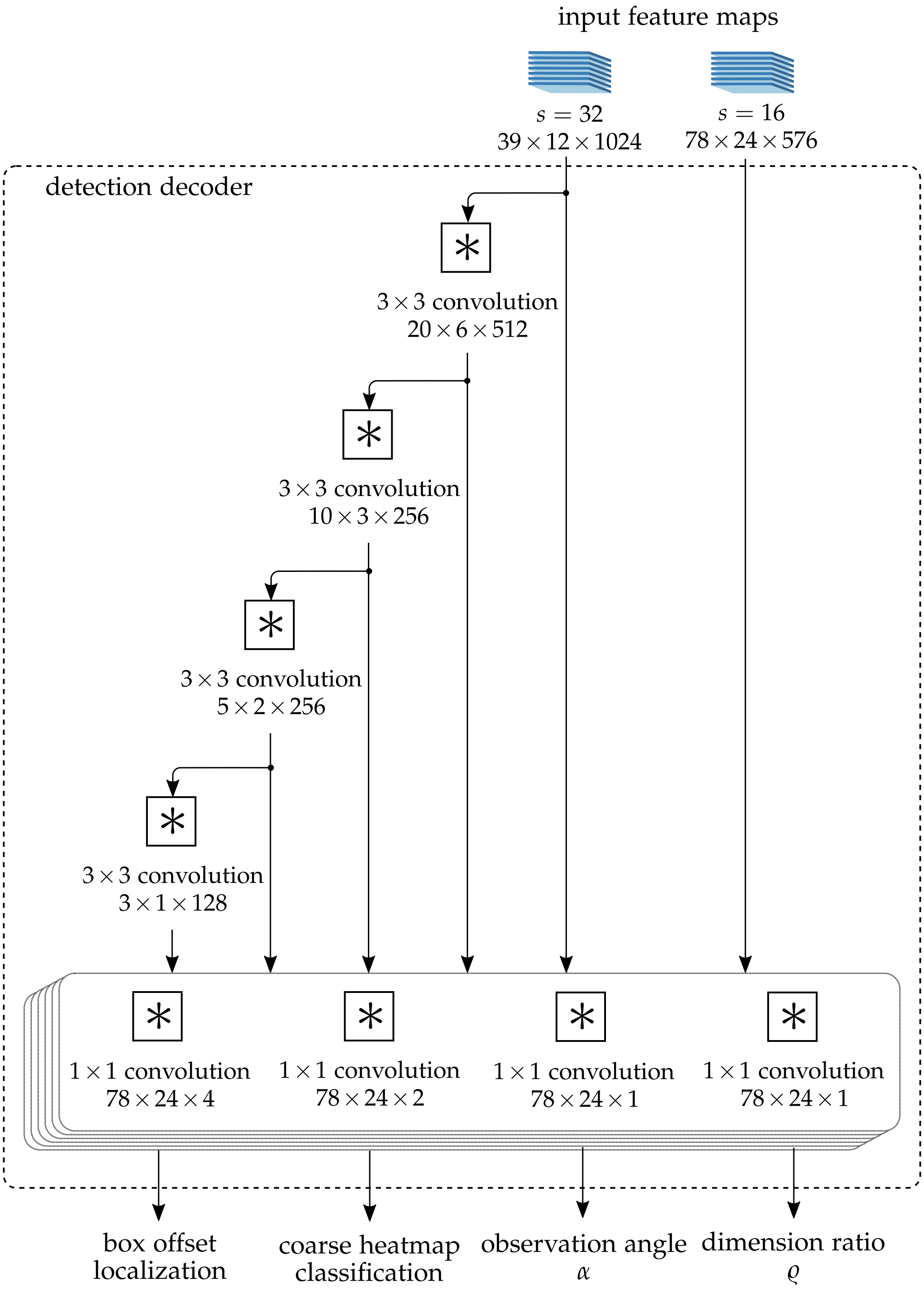


Figure A.6: Overview of the SSD decoder with auxiliary regressands (SSD+AUX).

A.7 Dual-task Rec+Seg *pre-rec* curves for road topology recognition

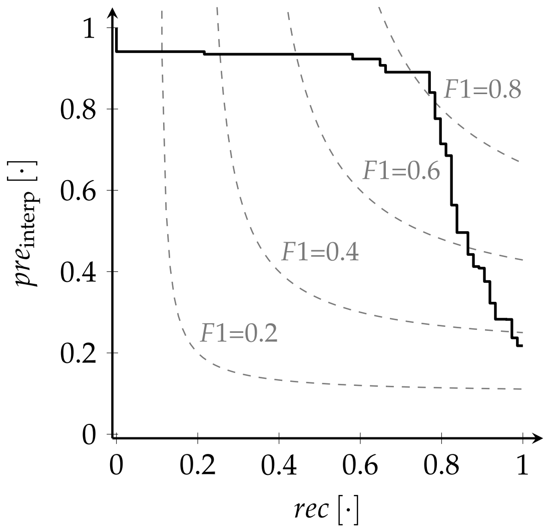


Figure A.7.a: Road topology a.: straight road

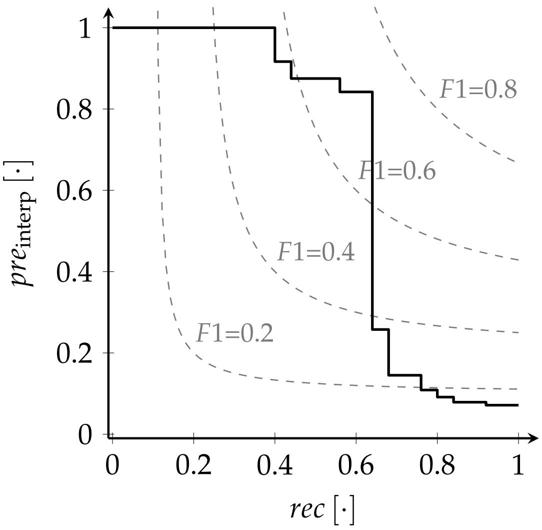


Figure A.7.b: Road topology b.: turn right

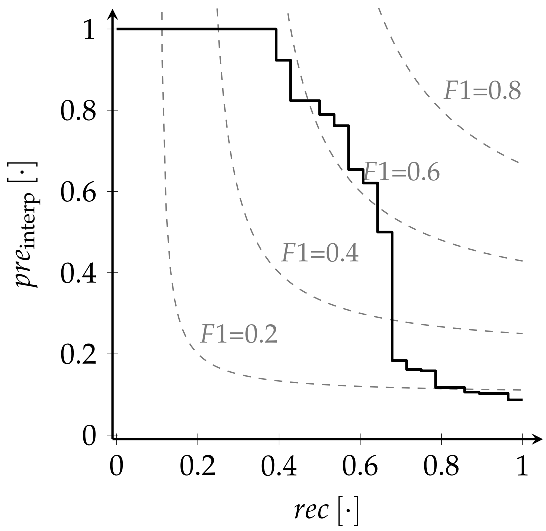


Figure A.7.c: Road topology c.: turn left

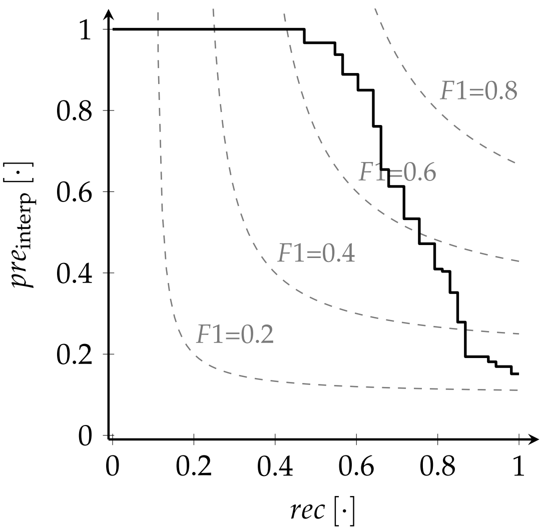


Figure A.7.d: Road topology d.: junction right

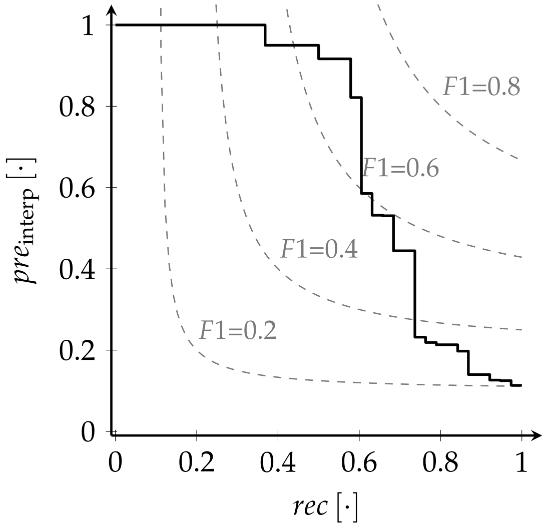


Figure A.7.e: Road topology e.: junction left

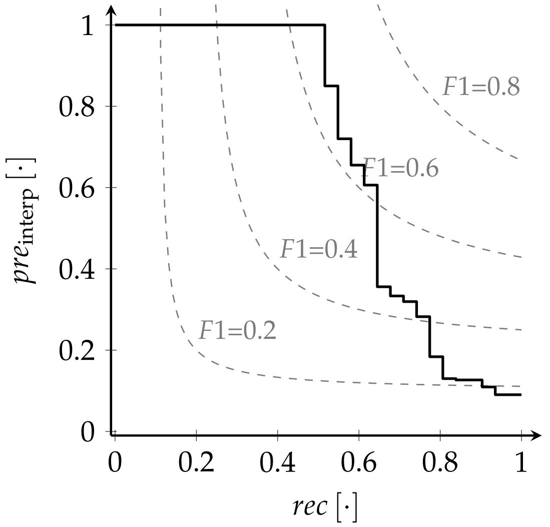


Figure A.7.f: Road topology f.: fork junction

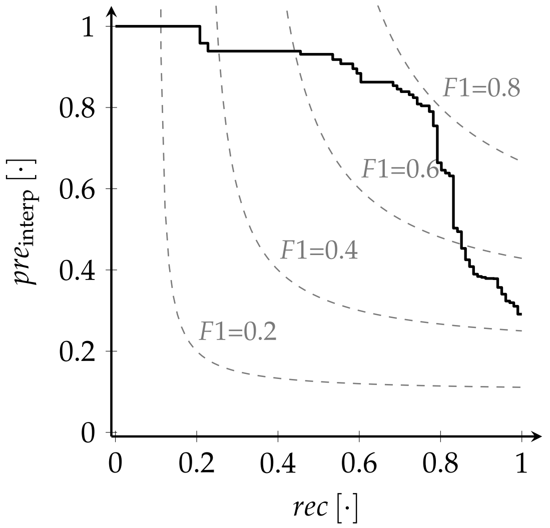


Figure A.7.g: Road topology g.: intersection

A.8 Dual-task Rec+Det *pre*-*rec* curves for road topology recognition

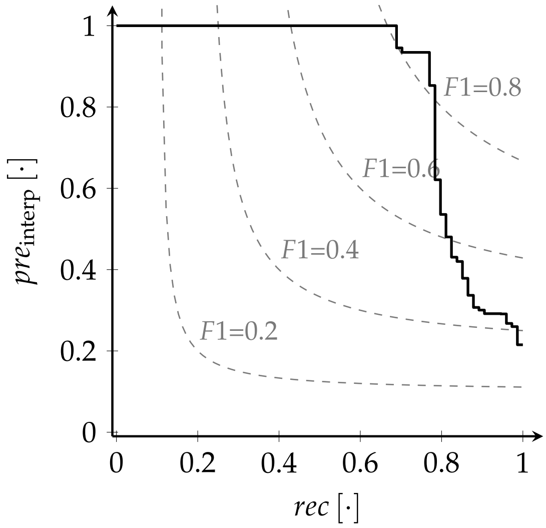


Figure A.8.a: Road topology a.: straight road

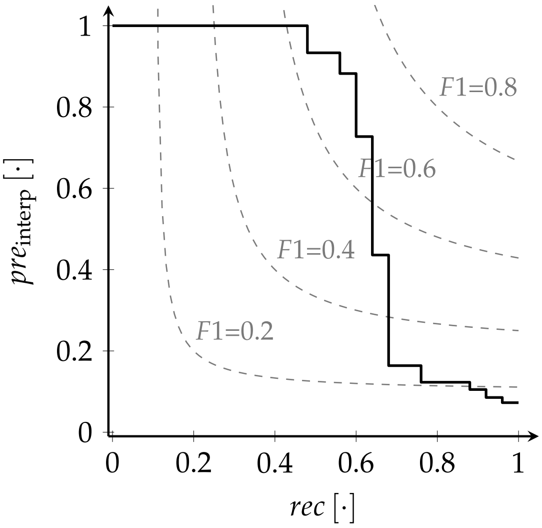


Figure A.8.b: Road topology b.: turn right

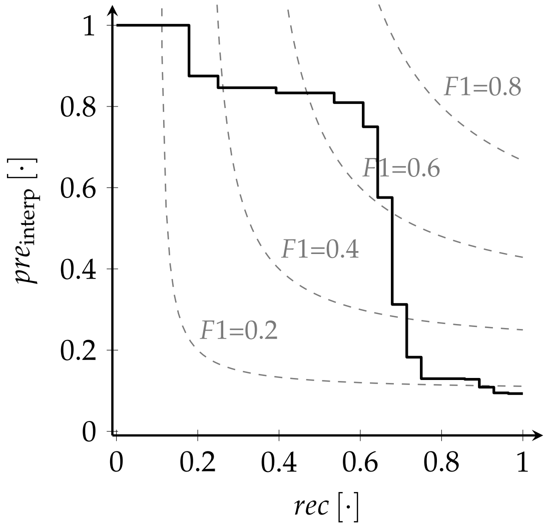


Figure A.8.c: Road topology c.: turn left

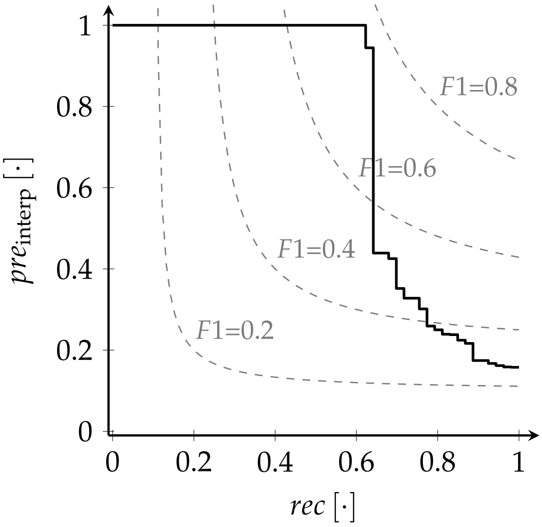


Figure A.8.d: Road topology d.: junction right

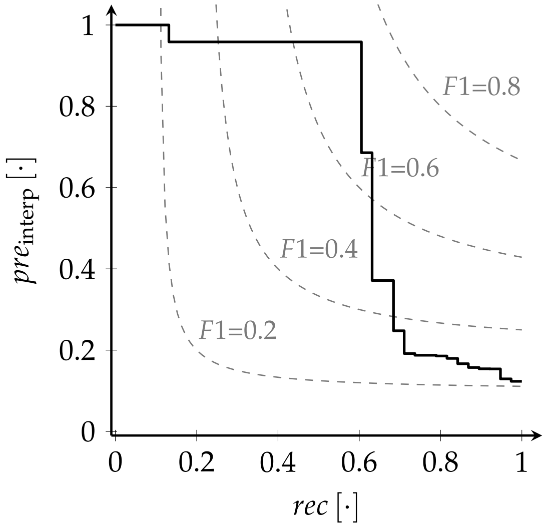


Figure A.8.e: Road topology e.: junction left

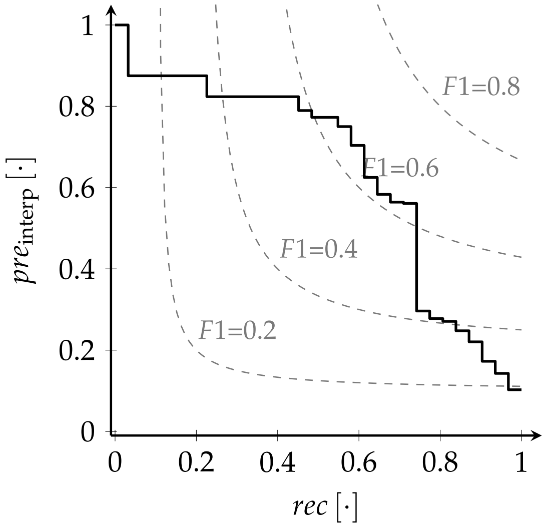


Figure A.8.f: Road topology f.: fork junction

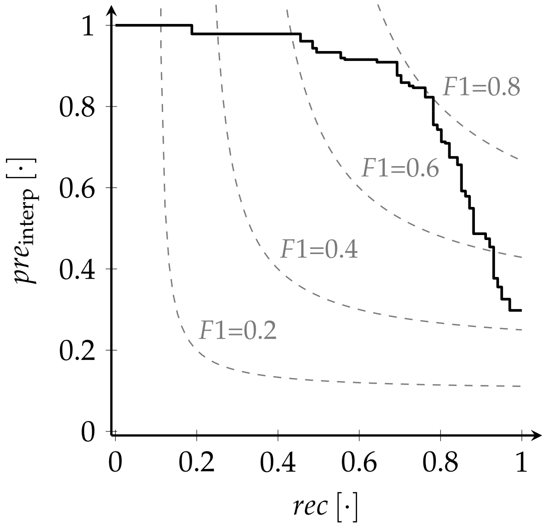


Figure A.8.g: Road topology g.: intersection

A.9 Multi-task *pre*-*rec* curves for road topology recognition

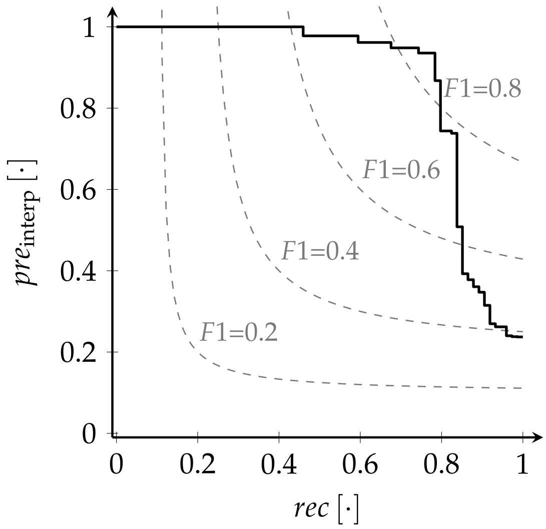


Figure A.9.a: Road topology a.: straight road

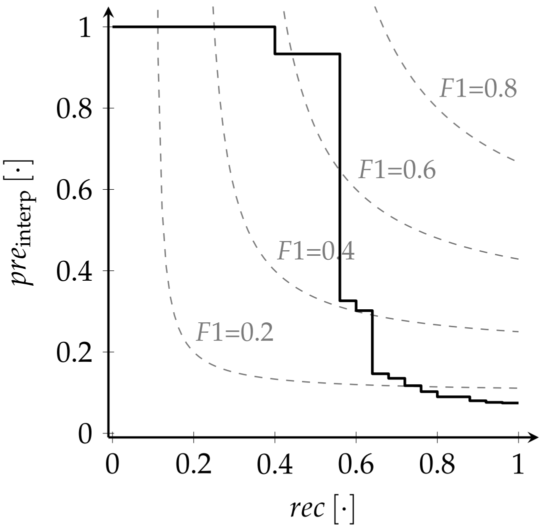


Figure A.9.b: Road topology b.: turn right

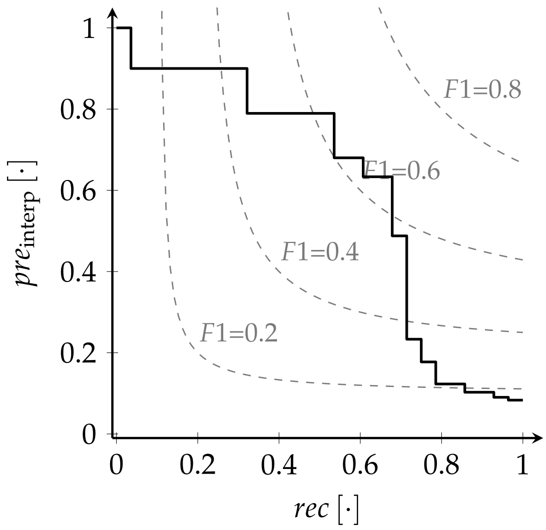


Figure A.9.c: Road topology c.: turn left

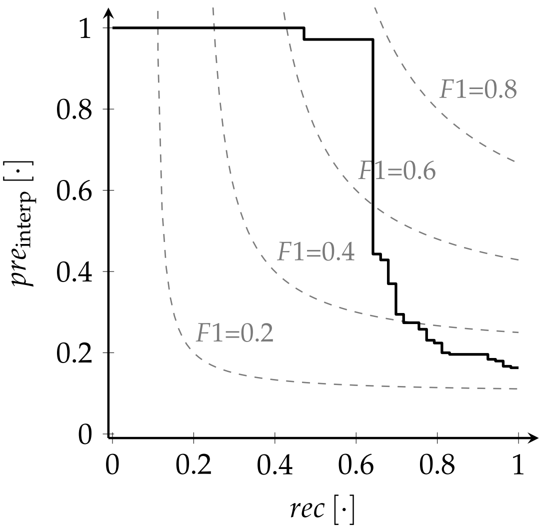


Figure A.9.d: Road topology d.: junction right

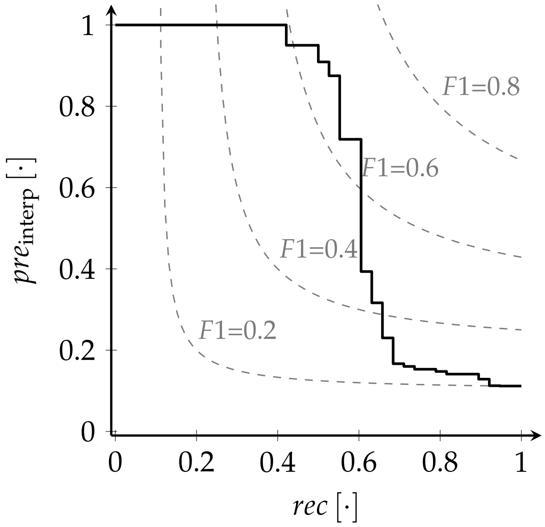


Figure A.9.e: Road topology e.: junction left

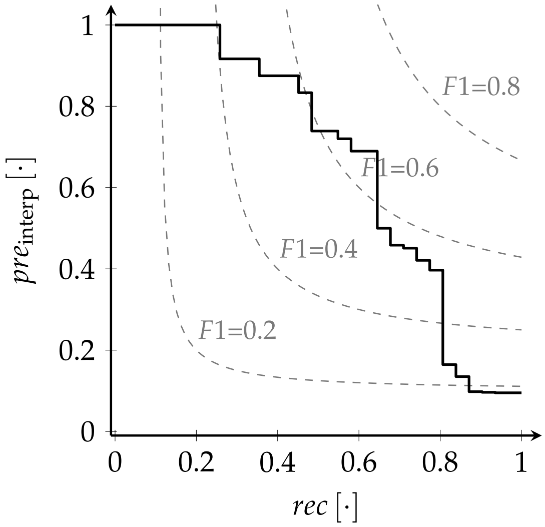


Figure A.9.f: Road topology f.: fork junction

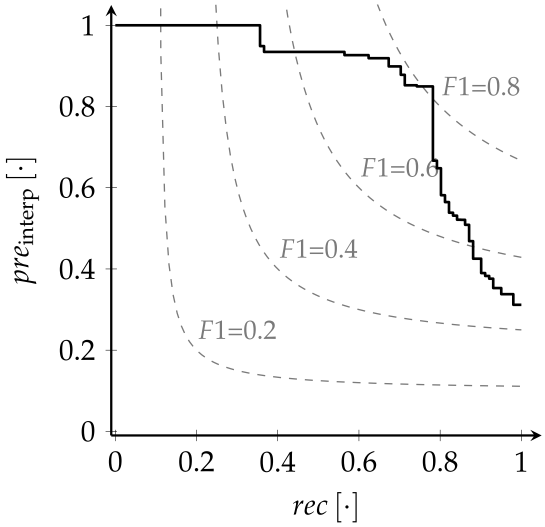
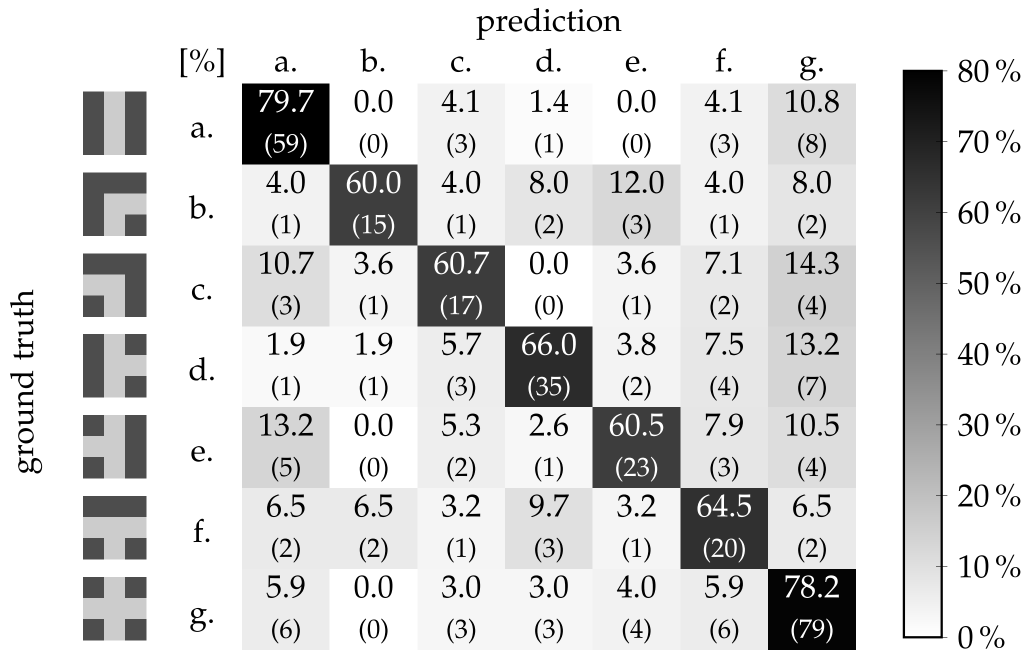


Figure A.9.g: Road topology g.: intersection

A.10 Dual-task road topology confusion matrices

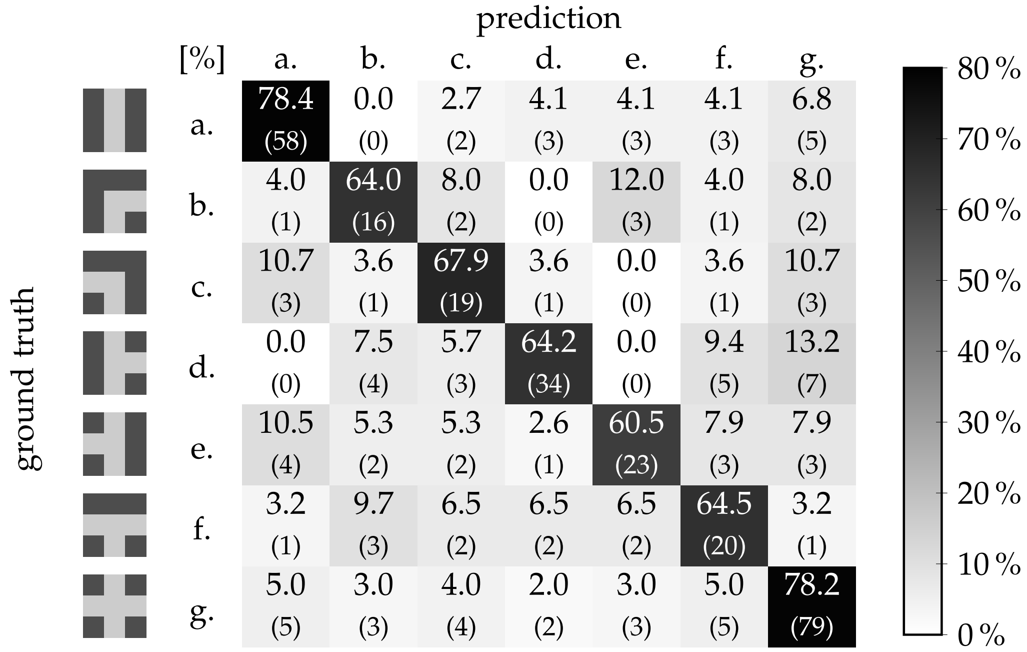
Table A.10.a: Full confusion matrix for the dual-task Rec+Seg model



ground truth \ prediction [%]	a.	b.	c.	d.	e.	f.	g.
a.	79.7 (59)	0.0 (0)	4.1 (3)	1.4 (1)	0.0 (0)	4.1 (3)	10.8 (8)
b.	4.0 (1)	60.0 (15)	4.0 (1)	8.0 (2)	12.0 (3)	4.0 (1)	8.0 (2)
c.	10.7 (3)	3.6 (1)	60.7 (17)	0.0 (0)	3.6 (1)	7.1 (2)	14.3 (4)
d.	1.9 (1)	1.9 (1)	5.7 (3)	66.0 (35)	3.8 (2)	7.5 (4)	13.2 (7)
e.	13.2 (5)	0.0 (0)	5.3 (2)	2.6 (1)	60.5 (23)	7.9 (3)	10.5 (4)
f.	6.5 (2)	6.5 (2)	3.2 (1)	9.7 (3)	3.2 (1)	64.5 (20)	6.5 (2)
g.	5.9 (6)	0.0 (0)	3.0 (3)	3.0 (3)	4.0 (4)	5.9 (6)	78.2 (79)

80 %
70 %
60 %
50 %
40 %
30 %
20 %
10 %
0 %

Table A.10.b: Full confusion matrix for the dual-task Rec+Det model



ground truth \ prediction [%]	a.	b.	c.	d.	e.	f.	g.
a.	78.4 (58)	0.0 (0)	2.7 (2)	4.1 (3)	4.1 (3)	4.1 (3)	6.8 (5)
b.	4.0 (1)	64.0 (16)	8.0 (2)	0.0 (0)	12.0 (3)	4.0 (1)	8.0 (2)
c.	10.7 (3)	3.6 (1)	67.9 (19)	3.6 (1)	0.0 (0)	3.6 (1)	10.7 (3)
d.	0.0 (0)	7.5 (4)	5.7 (3)	64.2 (34)	0.0 (0)	9.4 (5)	13.2 (7)
e.	10.5 (4)	5.3 (2)	5.3 (2)	2.6 (1)	60.5 (23)	7.9 (3)	7.9 (3)
f.	3.2 (1)	9.7 (3)	6.5 (2)	6.5 (2)	6.5 (2)	64.5 (20)	3.2 (1)
g.	5.0 (5)	3.0 (3)	4.0 (4)	2.0 (2)	3.0 (3)	5.0 (5)	78.2 (79)

80 %
70 %
60 %
50 %
40 %
30 %
20 %
10 %
0 %

A.11 Detailed breakdown of the multi-task KITTI road segmentation results

Table A.11.a: Detailed breakdown of the dual-task KITTI road segmentation results of the Seg+Rec model into the road types defined in [Fri+13].

Benchmark	$F1_{\mathrm{max}}$	mAP	pre	rec	FP-rate	FN-rate
UM_ROAD	94.10 %	91.58 %	94.52 %	93.67 %	2.47 %	6.33 %
UMM_ROAD	96.33 %	93.60 %	95.75 %	96.90 %	4.72 %	3.10 %
UU_ROAD	92.81 %	90.59 %	93.44 %	92.19 %	2.11 %	7.81 %
URBAN_ROAD	94.81 %	91.97 %	94.76 %	94.86 %	2.89 %	5.14 %

Table A.11.b: Detailed breakdown of the dual-task KITTI road segmentation results of the Seg+Det model into the road types defined in [Fri+13].

Benchmark	$F1_{\mathrm{max}}$	mAP	pre	rec	FP-rate	FN-rate
UM_ROAD	93.82 %	91.38 %	94.88 %	92.79 %	2.28 %	7.21 %
UMM_ROAD	96.07 %	93.57 %	95.78 %	96.36 %	4.67 %	3.64 %
UU_ROAD	92.23 %	90.40 %	92.77 %	91.70 %	2.33 %	8.30 %
URBAN_ROAD	94.45 %	91.86 %	94.92 %	93.99 %	2.77 %	6.01 %

Table A.11.c: Detailed breakdown of the multi-task KITTI road segmentation results of the Seg+Rec+Det model into the road types defined in [Fri+13].

Benchmark	$F1_{\mathrm{max}}$	mAP	pre	rec	FP-rate	FN-rate
UM_ROAD	93.71 %	91.30 %	94.26 %	93.17 %	2.59 %	6.83 %
UMM_ROAD	96.00 %	93.91 %	95.69 %	96.30 %	4.77 %	3.70 %
UU_ROAD	92.24 %	90.33 %	93.16 %	91.33 %	2.19 %	8.67 %
URBAN_ROAD	94.41 %	91.98 %	94.63 %	94.19 %	2.94 %	5.81 %

A.12 Full runtime measurement data

Table A.12: Runtimes of the single-task, dual-task, and full multi-task CNNs measured on a dedicated desktop GPU (see Table A.2.d) and on the embedded hardware platform Jetson TX2 (see Table A.2.c). The runtime evaluation is based on 100 randomly selected images.

	desktop GPU	Jetson TX2
Rec	22.4 ± 1.3 ms	167.1 ± 1.4 ms
Seg	19.8 ± 1.1 ms	145.2 ± 1.6 ms
Det	24.8 ± 1.2 ms	183.3 ± 1.0 ms
Rec+Seg	23.4 ± 1.2 ms	175.5 ± 1.5 ms
Rec+Det	28.0 ± 0.8 ms	207.8 ± 1.2 ms
Seg+Det	25.1 ± 0.7 ms	187.9 ± 1.4 ms
Rec+Seg+Det	31.4 ± 1.6 ms	233.7 ± 1.3 ms
sequential execution	67.0 ms	495.6 ms

Bibliography

[Aba+16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. "TensorFlow: A System for Large-scale Machine Learning". In: *Proceedings of the 12th Conference on Operating Systems Design and Implementation*. Savannah, USA: USENIX Association, 2016, pp. 265–283.

[Aeb19] M. Aeberhard. "Building the Universal Autonomous Driving System". In: *Tagungsband 14. DortmunderAutoTag*. Dortmund, Germany: TU Dortmund University, Institute of Control Theory and Systems Engineering, 2019, pp. 3–33.

[Ans+18] J. A. Ansari, S. Sharma, A. Majumdar, J. K. Murthy, and K. M. Krishna. "The Earth ain't Flat: Monocular Reconstruction of Vehicles on Steep and Graded Roads from a Moving Camera". In: *Proceedings of the International Conference on Intelligent Robots and Systems*. Madrid, Spain: IEEE, 2018, pp. 8404–8410.

[Arg+08] A. Argyriou, T. Evgeniou, and M. Pontil. "Convex multi-task feature learning". In: *Machine Learning* 73.3 (2008), pp. 243–272.

[Bar+20] I. Barabanau, A. Artemov, E. Burnaev, and V. Murashkin. "Monocular 3D Object Detection via Geometric Reasoning on Keypoints". In: *Proceedings of the 15th International Conference on Computer Vision Theory and Applications*. Scitepress. Valletta, Malta, 2020, pp. 652–659.

[Bax00] J. Baxter. "A Model of Inductive Bias Learning". In: *Journal of Artificial Intelligence Research* 12 (2000), pp. 149–198.

[Bin+09] A. Binder, M. Kawanabe, and U. Brefeld. "Efficient Classification of Images with Taxonomies". In: *Proceedings of the 9th Asian Conference on Computer Vision*. Springer. Xi'an, China, 2009, pp. 351–362.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.

[BL19] G. Brazil and X. Liu. "M3D-RPN: Monocular 3D Region Proposal Network for Object Detection". In: *Proceedings of the 17th International Conference on Computer Vision*. IEEE. Seoul, South Korea, 2019, pp. 9287–9296.

[Bou+10] Y.-L. Boureau, J. Ponce, and Y. LeCun. "A Theoretical Analysis of Feature Pooling in Visual Recognition". In: *Proceedings of the 27th International*

Conference on Machine Learning. Omnipress. Haifa, Israel, 2010, pp. 111–118.

[Bru+15] C.-A. Brust, S. Sickert, M. Simon, E. Rodner, and J. Denzler. "Convolutional Patch Networks with Spatial Prior for Road Detection and Urban Scene Understanding". In: *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*. Scitepress. Berlin, Germany, 2015, pp. 510–517.

[BS03] S. Ben-David and R. Schuller. "Exploiting Task Relatedness for Multiple Task Learning". In: *Proceedings of the 16th Annual Conference on Learning Theory*. Springer. Washington D.C., USA, 2003, pp. 567–580.

[BT12] A. Barriuso and A. Torralba. "Notes on image annotation". In: *arXiv preprint arXiv:1210.3448* (2012).

[Car93] R. Caruana. "Multitask Learning: A Knowledge-Based Source of Inductive Bias". In: *Proceedings of the 10th International Conference on Machine Learning*. Amherst, USA: Morgan Kaufmann, 1993, pp. 41–48.

[Car97] R. Caruana. "Multitask learning". In: *Machine Learning* 28.1 (1997), pp. 41–75.

[CC17] Z. Chen and Z. Chen. "RBNet: A Deep Neural Network for Unified Road and Road Boundary Detection". In: *Proceedings of the 24th International Conference on Neural Information Processing*. Springer. Guangzhou, China, 2017, pp. 677–687.

[Cha+17] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau. "Deep MANTA: A Coarse-to-fine Many-Task Network for joint 2D and 3D vehicle analysis from monocular image". In: *Proceedings of the 30th Conference on Computer Vision and Pattern Recognition*. IEEE. Honolulu, USA, 2017, pp. 2040–2049.

[Che+16a] T. Chen, B. Xu, C. Zhang, and C. Guestrin. "Training Deep Nets with Sublinear Memory Cost". In: *arXiv preprint arXiv:1604.06174* (2016).

[Che+16b] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. "Monocular 3D Object Detection for Autonomous Driving". In: *Proceedings of the 29th Conference on Computer Vision and Pattern Recognition*. IEEE. Las Vegas, USA, 2016, pp. 2147–2156.

[Che+17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2017), pp. 834–848.

[Cho17] F. Chollet. "Xception: Deep Learning With Depthwise Separable Convolutions". In: *Proceedings of the 30th Conference on Computer Vision and Pattern Recognition*. IEEE. Honolulu, USA, 2017, pp. 1251–1258.

[CK00] G. Chen and D. Kotz. *A Survey of Context-Aware Mobile Computing Research*. Computer Science Technical Report TR2000-381. Tech. rep. 2000.

[Cle+16] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". In: *Proceedings of the 4th International Conference on Learning Representations*. San Juan, Puerto Rico, 2016.

[Cor+16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the 29th Conference on Computer Vision and Pattern Recognition*. IEEE. Las Vegas, USA, 2016, pp. 3213–3223.

[Cor17] M. Cordts. "Understanding Cityscapes: Efficient Urban Semantic Scene Understanding". Doctoral dissertation. Technische Universität Darmstadt, 2017.

[Cyb89] G. Cybenko. "Approximation by Superpositions of a Sigmoidal Function". In: *Mathematics of Control, Signals and Systems* 2.4 (1989), pp. 303–314.

[Dem68] A. P. Dempster. "A Generalization of Bayesian inference". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 30.2 (1968), pp. 205–232.

[DG06] J. Davis and M. Goadrich. "The Relationship Between Precision-Recall and ROC Curves". In: *Proceedings of the 23rd International Conference on Machine Learning*. Association for Computing Machinery. Pittsburgh, USA, 2006, pp. 233–240.

[Di+16] S. Di, H. Zhang, X. Mei, D. Prokhorov, and H. Ling. "A Benchmark for Cross-Weather Traffic Scene Understanding". In: *Proceedings of the 19th International Conference on Intelligent Transportation Systems*. IEEE. Rio de Janeiro, Brazil, 2016, pp. 2150–2156.

[Di+17] S. Di, H. Zhang, C.-G. Li, X. Mei, D. Prokhorov, and H. Ling. "Cross-Domain Traffic Scene Understanding: A Dense Correspondence-Based Transfer Learning Approach". In: *Transactions on Intelligent Transportation Systems* 19.3 (2017), pp. 745–757.

[Die+05] K. Dietmayer, A. Kirchner, and N. Kämpchen. "Fusionsarchitekturen zur Umfeldwahrnehmung für zukünftige Fahrerassistenzsysteme". In: *Fahrerassistenzsysteme mit maschineller Wahrnehmung*. Springer, 2005, pp. 59–88.

[Din+20] M. Ding, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo. "Learning Depth-Guided Convolutions for Monocular 3D Object Detection". In: *Proceedings of the Computer Vision and Pattern Recognition Workshops*. IEEE. Virtual conference, 2020, pp. 4306–4315.

[Duc+11] J. Duchi, E. Hazan, and Y. Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *Journal of Machine Learning Research* 12.61 (2011), pp. 2121–2159.

[Duo+15] L. Duong, T. Cohn, S. Bird, and P. Cook. “Low Resource Dependency Parsing: Cross-lingual Parameter Sharing in a Neural Network Parser”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. Beijing, China, 2015, pp. 845–850.

[DV16] V. Dumoulin and F. Visin. “A guide to convolution arithmetic for deep learning”. In: *arXiv preprint arXiv:1603.07285* (2016).

[EF15] D. Eigen and R. Fergus. “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture”. In: *Proceedings of the 15th International Conference on Computer Vision*. IEEE. Santiago, Chile, 2015, pp. 2650–2658.

[Elf89] A. Elfes. “Using Occupancy Grids for Mobile Robot Perception and Navigation”. In: *Computer* 22.6 (1989), pp. 46–57.

[Eng+18] N. Engel, S. Hoermann, P. Henzler, and K. Dietmayer. “Deep Object Tracking on Dynamic Occupancy Grid Maps Using RNNs”. In: *Proceedings of the 21st International Conference on Intelligent Transportation Systems*. IEEE. Maui, USA, 2018, pp. 3852–3858.

[Ess+09] A. Ess, T. Müller, H. Grabner, and L. van Gool. “Segmentation-Based Urban Traffic Scene Understanding”. In: *Proceedings of the 20th British Machine Vision Conference*. BMVA Press. London, United Kingdom, 2009, pp. 1–11.

[Eve+10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. “The PASCAL Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338.

[Far+13] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. “Learning Hierarchical Features for Scene Labeling”. In: *Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013), pp. 1915–1929.

[Fas+95] W. Fastenmeier et al. *Autofahrer und Verkehrssituation. Neue Wege zur Bewertung von Sicherheit und Zuverlässigkeit moderner Strassenverkehrssysteme*. 33. Verlag TÜV Rheinland GmbH, 1995.

[Fay+16] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, F. Huang, and R. Klette. “STFCN: Spatio-Temporal Fully Convolutional Neural Network for Semantic Segmentation of Street Scenes”. In: *Proceedings of the 13th Asian Conference on Computer Vision Workshops*. Springer. Taipei, Taiwan, 2016, pp. 493–509.

[Fed20a] Federal Motor Transport Authority of Germany (Kraftfahrt-Bundesamt). *Pressemitteilung Nr. 6/2020*. 2020. URL: `https://de.statista.com/`

statistik/daten/studie/12131/umfrage/pkw-bestand-in-deutschland/. Accessed: September 7th, 2020.

[Fed20b] Federal Statistical Office of Germany (Statistisches Bundesamt). *Press release No. 061*. 2020. URL: https://www.destatis.de/EN/Press/2020/02/PE20_061_46241.html. Accessed: September 7th, 2020.

[Fle+03] F. O. Flemisch, C. A. Adams, S. R. Conway, K. H. Goodrich, M. T. Palmer, and P. C. Schutte. *The H-Metaphor as a Guideline for Vehicle Automation and Interaction*. Technical Memorandum (TM) 20040031835. Tech. rep. 2003.

[FP05] L. Fei-Fei and P. Perona. "A Bayesian Hierarchical Model for Learning Natural Scene Categories". In: *Proceedings of the 18th Conference on Computer Vision and Pattern Recognition*. IEEE. San Diego, USA, 2005, pp. 524–531.

[Fra+15] B. Franz, M. Kauer, S. Geyer, and S. Hakuli. "Conduct-by-Wire". In: *Handbuch Fahrerassistenzsysteme*. Springer, 2015, pp. 1111–1121.

[Fri+13] J. Fritsch, T. Kühnl, and A. Geiger. "A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms". In: *Proceedings of the 16th International Conference on Intelligent Transportation Systems*. IEEE. The Hague, Netherlands, 2013, pp. 1693–1700.

[Gäh+18] N. Gählert, M. Mayer, L. Schneider, U. Franke, and J. Denzler. "MB-Net: MergeBoxes for Real-Time 3D Vehicles Detection". In: *Proceedings of the 29th Intelligent Vehicles Symposium*. IEEE. Changshu, China, 2018, pp. 2117–2124.

[GB10] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the 13th International Conferenceon Artificial Intelligence and Statistics*. MLR press. Sardinia, Italy, 2010, pp. 249–256.

[Gei+12] A. Geiger, P. Lenz, and R. Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Proceedings of the 25th Conference on Computer Vision and Pattern Recognition*. IEEE. Rhode Island, USA, 2012, pp. 3354–3361.

[Gei+13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets robotics: The KITTI dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[Geo+18] J.-M. Georg, J. Feiler, F. Diermeyer, and M. Lienkamp. "Teleoperated Driving, a Key Technology for Automated Driving? Comparison of Actual Test Drives with a Head Mounted Display and Conventional Monitors". In: *Proceedings of the 21st International Conference on Intelligent Transportation Systems*. IEEE. Maui, USA, 2018, pp. 3403–3408.

[Gir+14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings* *of the 27th Conference on Computer Vision and Pattern Recognition*. IEEE. Columbus, USA, 2014, pp. 580–587.

[Gir15] R. Girshick. "Fast R-CNN". In: *Proceedings of the 15th International Conference on Computer Vision*. IEEE. Santiago, Chile, 2015, pp. 1440–1448.

[Glo+11] X. Glorot, A. Bordes, and Y. Bengio. "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the 14th International Conferenceon Artificial Intelligence and Statistics*. MLR press. Ft. Lauderdale, USA, 2011, pp. 315–323.

[Goo+14] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks". In: *Proceedings of the 2nd International Conference on Learning Representations*. Banff, Canada, 2014.

[Goo+16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT press, 2016.

[Goo19a] Google LLC. *Edge TPU*. 2019. URL: `https://cloud.google.com/edge-tpu`. Accessed: July 9th, 2020.

[Goo19b] Google LLC. *TensorFlow Lite*. 2019. URL: `https://www.tensorflow.org/lite`. Accessed: July 9th, 2020.

[Gra14] B. Graham. "Fractional Max-Pooling". In: *arXiv preprint arXiv:1412.6071* (2014).

[Gre+12] R. Grewe, A. Hohm, S. Hegemann, S. Lueke, and H. Winner. "Towards a Generic and Efficient Environment Model for ADAS". In: *Proceedings of the 23rd Intelligent Vehicles Symposium*. IEEE. Alcalá de Henares, Spain, 2012, pp. 316–321.

[Gre+14a] R. Grewe, A. Hohm, and S. Lueke. "An efficient environmental model for automated driving". In: *Tagungsband 14. Internationales Stuttgarter Symposium*. Springer. Stuttgart, Germany, 2014, pp. 267–280.

[Gre+14b] R. Grewe, A. Hohm, S. Lüke, and H. Winner. "Umfeldmodelle - standardisierte Schnittstellen für Assistenzsysteme". In: *Vernetztes Automobil*. Springer, 2014, pp. 207–213.

[Gui+18] C. Guindel, D. Martin, and J. M. Armingol. "Fast Joint Object Detection and Viewpoint Estimation for Traffic Scene Understanding". In: *Intelligent Transportation Systems Magazine* 10.4 (2018), pp. 74–86.

[Hah+00] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit". In: *Nature* 405.6789 (2000), pp. 947–951.

[Han+17] X. Han, H. Wang, J. Lu, and C. Zhao. "Road detection based on the fusion of Lidar and image data". In: *International Journal of Advanced Robotic Systems* 14.6 (2017), pp. 1–10.

[Har+15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. "Hypercolumns for Object Segmentation and Fine-grained Localization". In: *Proceedings of the 28th Conference on Computer Vision and Pattern Recognition*. IEEE. Boston, USA, 2015, pp. 447–456.

[Has+17] K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher. "A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. Copenhagen, Denmark, 2017, pp. 1923–1933.

[He+15a] K. He, X. Zhang, S. Ren, and J. Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the 15th International Conference on Computer Vision*. IEEE. Santiago, Chile, 2015, pp. 1026–1034.

[He+15b] K. He, X. Zhang, S. Ren, and J. Sun. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *Transactions on Pattern Analysis and Machine Intelligence* 37.9 (2015), pp. 1904–1916.

[He+16] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *Proceedings of the 29th Conference on Computer Vision and Pattern Recognition*. IEEE. Las Vegas, USA, 2016, pp. 770–778.

[He+17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. "Mask R-CNN". In: *Proceedings of the 16th International Conference on Computer Vision*. IEEE. Venice, Italy, 2017, pp. 2980–2988.

[He+19] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li. "Bag of Tricks for Image Classification with Convolutional Neural Networks". In: *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition*. IEEE. Long Beach, USA, 2019, pp. 558–567.

[Hoc91] S. Hochreiter. "Untersuchungen zu dynamischen neuronalen Netzen". Diploma thesis. Technische Universität München, 1991.

[Hoi+08] D. Hoiem, A. A. Efros, and M. Hebert. "Putting Objects in Perspective". In: *International Journal of Computer Vision* 80.1 (2008), pp. 3–15.

[Hon+15] S. Hong, H. Noh, and B. Han. "Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems*. MIT Press. Montréal, Canada, 2015, pp. 1495–1503.

[Hor+89] K. Hornik, M. Stinchcombe, H. White, et al. "Multilayer Feedforward Networks are Universal Approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366.

[How+17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. "MobileNets: Efficient Convolutional Neural

Networks for Mobile Vision Applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[HS19] T. He and S. Soatto. "Mono3D++: Monocular 3D Vehicle Detection with Two-Scale 3D Hypotheses and Task Priors". In: *Proceedings of the 33rd Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence. New York, USA, 2019, pp. 8409–8416.

[Hsu+17] L.-T. Hsu, Y. Gu, and S. Kamijo. "Intelligent Viaduct Recognition and Driving Altitude Determination using GPS Data". In: *Transactions on Intelligent Vehicles* 2.3 (2017), pp. 175–184.

[HT01] D. J. Hand and R. J. Till. "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems". In: *Machine Learning* 45.2 (2001), pp. 171–186.

[Hua+17] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. "Speed/accuracy trade-offs for modern convolutional object detectors". In: *Proceedings of the 30th Conference on Computer Vision and Pattern Recognition*. IEEE. Honolulu, USA, 2017, pp. 7310–7311.

[HW59] D. H. Hubel and T. N. Wiesel. "Receptive fields of single neurones in the cat's striate cortex". In: *The Journal of Physiology* 148.3 (1959), pp. 574–591.

[HW62] D. H. Hubel and T. N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of Physiology* 160.1 (1962), pp. 106–154.

[Hyv+05] A. Hyvärinen, M. Gutmann, and P. O. Hoyer. "Statistical model of natural stimuli predicts edge-like pooling of spatial frequency channels in V2". In: *BMC Neuroscience* 6.1 (2005), pp. 1–12.

[HZ03] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003.

[Ian+16] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size". In: *arXiv preprint arXiv:1602.07360* (2016).

[IK04] M. Ito and H. Komatsu. "Representation of Angles Embedded within Contour Stimuli in Area V2 of Macaque Monkeys". In: *Journal of Neuroscience* 24.13 (2004), pp. 3313–3324.

[IS15] S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. MLR press. Lille, France, 2015, pp. 448–456.

[Jör+19] E. Jörgensen, C. Zach, and F. Kahl. “Monocular 3D Object Detection and Box Fitting Trained End-to-End Using Intersection-over-Union Loss”. In: *arXiv preprint arXiv:1906.08070* (2019).

[Kas+09] R. Kastner, F. Schneider, T. Michalke, J. Fritsch, and C. Goerick. “Image-based classification of driving scenes by Hierarchical Principal Component Classification (HPCC)”. In: *Proceedings of the 20th Intelligent Vehicles Symposium*. IEEE. Xi’an, China, 2009, pp. 341–346.

[Kas+11] R. Kastner, T. Michalke, J. Adamy, J. Fritsch, and C. Goerick. “Task-Based Environment Interpretation and System Architecture for Next Generation ADAS”. In: *Intelligent Transportation Systems Magazine* 3.4 (2011), pp. 20–33.

[KB15] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *Proceedings of the 3rd International Conference on Learning Representations*. San Diego, USA, 2015.

[KB17] A. Kaehler and G. Bradski. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. first edition. O’Reilly Media, Inc., 2017.

[Ken+18] A. Kendall, Y. Gal, and R. Cipolla. “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics”. In: *Proceedings of the 31st Conference on Computer Vision and Pattern Recognition*. IEEE. Salt Lake City, USA, 2018, pp. 7482–7491.

[Kir+19] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. “Panoptic Segmentation”. In: *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition*. IEEE. Long Beach, USA, 2019, pp. 9404–9413.

[KK19] Y. Kim and D. Kum. “Deep Learning based Vehicle Position and Orientation Estimation via Inverse Perspective Mapping Image”. In: *Proceedings of the 30th Intelligent Vehicles Symposium*. IEEE. Paris, France, 2019, pp. 317–323.

[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 30th Conference on Computer Vision and Pattern Recognition*. IEEE. Honolulu, USA, 2017, pp. 6129–6138.

[Kri+12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems*. Curran Associates Inc. Lake Tahoe, USA, 2012, pp. 1097–1105.

[Ku+19] J. Ku, A. D. Pon, and S. L. Waslander. “Monocular 3D Object Detection Leveraging Accurate Proposalsand Shape Reconstruction”. In: *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition*. IEEE. Long Beach, USA, 2019, pp. 11867–11876.

[Kun+18] A. Kundu, Y. Li, and J. M. Rehg. “3D-RCNN: Instance-level 3D Object Reconstruction via Render-and-Compare”. In: *Proceedings of the 31st Conference on Computer Vision and Pattern Recognition*. IEEE. Salt Lake City, USA, 2018, pp. 3559–3568.

[Lee+11] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. “Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks”. In: *Communications of the ACM* 54.10 (2011), pp. 95–103.

[Lep+09] V. Lepetit, F. Moreno-Noguer, and P. Fua. “EPnP: An Accurate O(n) Solution to the PnP Problem”. In: *International Journal of Computer Vision* 81.2 (2009), p. 155.

[Li+19] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang. “GS3D: An Efficient 3D Object Detection Framework for Autonomous Driving”. In: *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition*. IEEE. Long Beach, USA, 2019, pp. 1019–1028.

[Lia+16] Y. Liao, S. Kodagoda, Y. Wang, L. Shi, and Y. Liu. “Understand Scene Categories by Objects: A Semantic Regularized Scene Classifier Using Convolutional Neural Networks”. In: *Proceedings of the 33rd International Conference on Robotics and Automation*. IEEE. Stockholm, Sweden, 2016, pp. 2318–2325.

[Lia+19] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun. “Multi-Task Multi-Sensor Fusion for 3D Object Detection”. In: *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition*. IEEE. Long Beach, USA, 2019, pp. 7345–7353.

[Lie+18] C. Lienke, M. Keller, K.-H. Glander, and T. Bertram. “Environment Modeling for the Application in Optimization-based Trajectory Planning”. In: *Proceedings of the 29th Intelligent Vehicles Symposium*. IEEE. Changshu, China, 2018, pp. 498–503.

[Lie+19] C. Lienke, M. Schmidt, C. Wissing, M. Keller, C. Manna, T. Nattermann, and T. Bertram. “Core components of automated driving – algorithms for situation analysis, decision-making, and trajectory planning”. In: *Tagungsband der 5. Internationalen ATZ-Fachtagung Automatisiertes Fahren*. Wiesbaden, Germany: Springer, 2019, pp. 195–215.

[Lin+14] M. Lin, Q. Chen, and S. Yan. “Network In Network”. In: *Proceedings of the 2nd International Conference on Learning Representations*. Banff, Canada, 2014.

[Lin+16] G. Lin, C. Shen, A. van den Hengel, and I. Reid. “Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation”. In: *Proceedings of the 29th Conference on Computer Vision and Pattern Recognition*. IEEE. Las Vegas, USA, 2016, pp. 3194–3203.

[Lin+20] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. “Focal Loss for Dense Object Detection”. In: *Transactions on Pattern Analysis and Machine Intelligence* 42.2 (2020), pp. 318–327.

[Liu+15] B. Liu, X. He, and S. Gould. “Multi-class Semantic Video Segmentation with Exemplar-based Object Reasoning”. In: *Proceedings of the 15th Winter Conference on Applications of Computer Vision*. IEEE. Hawaii, USA, 2015, pp. 1014–1021.

[Liu+16a] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. “SSD: Single Shot MultiBox Detector”. In: *Proceedings of the 14th European Conference on Computer Vision*. Amsterdam, Netherlands: Springer, 2016, pp. 21–37.

[Liu+16b] W. Liu, A. Rabinovich, and A. C. Berg. “ParseNet: Looking Wider to See Better”. In: *Workshop of the 4th International Conference on Learning Representations*. San Juan, Puerto Rico, 2016.

[Liu+19a] H. Liu, K. Simonyan, and Y. Yang. “DARTS: Differentiable Architecture Search”. In: *Proceedings of the 7th International Conference on Learning Representations*. New Orleans, USA, 2019.

[Liu+19b] L. Liu, J. Lu, C. Xu, Q. Tian, and J. Zhou. “Deep Fitting Degree Scoring Network for Monocular 3D Object Detection”. In: *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition*. IEEE. Long Beach, USA, 2019, pp. 1057–1066.

[LM11] M. J. Leotta and J. L. Mundy. “Vehicle Surveillance with a Generic, Adaptive, 3D Vehicle Model”. In: *Transactions on Pattern Analysis and Machine Intelligence* 33.7 (2011), pp. 1457–1469.

[Lu+17] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris. “Fully-adaptive Feature Sharing in Multi-Task Networks with Applications in Person Attribute Classification”. In: *Proceedings of the 30th Conference on Computer Vision and Pattern Recognition*. IEEE. Honolulu, USA, 2017, pp. 5334–5343.

[Luo+16] W. Luo, Y. Li, R. Urtasun, and R. Zemel. “Understanding the Effective Receptive Field in Deep Convolutional Neural Networks”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Curran Associates Inc. Barcelona, Spain, 2016, pp. 4898–4906.

[Lüt+18] M. Lütkemöller, M. Oeljeklaus, T. Bertram, K. Rink, U. Stählin, and R. Grewe. “Derivation and Application of an Observer Structure to Detect Inconsistencies Within a Static Environmental Model”. In: *Tagungsband der 4. Internationalen ATZ-Fachtagung Automatisiertes Fahren*. Wiesbaden, Germany: Springer, 2018, pp. 67–79.

[Maa+13] A. L. Maas, A. Y. Hannun, and A. Y. Ng. “Rectifier Nonlinearities Improve Neural Network Acoustic Models”. In: *Proceedings of the 30th International Conference on Machine Learning*. MLR press. Atlanta, USA, 2013.

[Man+19] F. Manhardt, W. Kehl, and A. Gaidon. “ROI-10D: Monocular Lifting of 2D Detection to 6D Pose and Metric Shape”. In: *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition*. IEEE. Long Beach, USA, 2019, pp. 2069–2078.

[Mas+16] F. Massa, R. Marlet, and M. Aubry. “Crafting a multi-task CNN for viewpoint estimation”. In: *Proceedings of the 27th British Machine Vision Conference*. BMVA Press. York, United Kingdom, 2016, pp. 91.1–91.12.

[Mat+15] R. Matthaei, A. Reschka, J. Rieken, F. Dierkes, S. Ulbrich, T. Winkle, and M. Maurer. “Autonomes Fahren”. In: *Handbuch Fahrerassistenzsysteme*. Springer, 2015, pp. 1139–1165.

[Meu+17] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz. “SymPy: symbolic computing in Python”. In: *Computer Science* 3.e103 (2017).

[Mey+19] R. Meyes, M. Lu, C. W. de Puiseau, and T. Meisen. “Ablation Studies in Artificial Neural Networks”. In: *arXiv preprint arXiv:1901.08644* (2019).

[Mil+02] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. “Network Motifs: Simple Building Blocks of Complex Networks”. In: *Science* 298.5594 (2002), pp. 824–827.

[Mos+15] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. “Feedforward semantic segmentation with zoom-out features”. In: *Proceedings of the 28th Conference on Computer Vision and Pattern Recognition*. IEEE. Boston, USA, 2015, pp. 3376–3385.

[Mot+15] R. Mottaghi, Y. Xiang, and S. Savarese. “A Coarse-to-Fine Model for 3D Pose Estimation and Sub-category Recognition”. In: *Proceedings of the 28th Conference on Computer Vision and Pattern Recognition*. IEEE. Boston, USA, 2015, pp. 418–426.

[Mou+17] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká. “3D Bounding Box Estimation Using Deep Learning and Geometry”. In: *Proceedings of the 30th Conference on Computer Vision and Pattern Recognition*. IEEE. Honolulu, USA, 2017, pp. 5632–5640.

[Nai+19] A. Naiden, V. Paunescu, G. Kim, B. Jeon, and M. Leordeanu. “Shift R-CNN: Deep Monocular 3D Object Detection With Closed-Form Geometric Constraints”. In: *Proceedings of the 26th International Conference on Image Processing*. IEEE. Taipei, Taiwan, 2019, pp. 61–65.

[Nic+08] J. Nickolls, I. Buck, M. Garland, and K. Skadron. “Scalable Parallel Programming with CUDA”. In: *Queue* 6.2 (2008), pp. 40–53.

[Nie14] D. Nienhüser. “Kontextsensitive Erkennung und Interpretation fahrrelevanter statischer Verkehrselemente”. Doctoral dissertation. Karlsruhe Institute of Technology, 2014.

[Nwa+18] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning”. In: *arXiv preprint arXiv:1811.03378* (2018).

[Oel+14] M. Oeljeklaus, F. Posada, F. Hoffmann, and T. Bertram. “Analyse globaler Bildmerkmale zur Klassifikation von Verkehrsszenen”. In: *Proceedings of the 24th Workshop Computational Intelligence*. VDI/VDE-GMA. Dortmund, Germany, 2014, pp. 299–314.

[Oel+15a] M. Oeljeklaus, F. Hoffmann, and T. Bertram. “Bildbasierte Detektion von Kontextinformationen über Verkehrssituationen”. In: *Tagungsband der 7. Fachtagung AUTOREG*. VDI/VDE-GMA. Baden-Baden, Germany, 2015, pp. 243–254.

[Oel+15b] M. Oeljeklaus, F. Hoffmann, and T. Bertram. “Convolutional Neural Networks für die semantische Segmentierung von Szenen”. In: *Proceedings of the 25th Workshop Computational Intelligence*. VDI/VDE-GMA. Dortmund, Germany, 2015, pp. 241–254.

[Oel+16a] M. Oeljeklaus, F. Hoffmann, and T. Bertram. “Kameragestützte Segmentierung von Verkehrsszenen für automatisierte Fahrzeugsysteme”. In: *Tagungsband der 2. Fachtagung IFToMM D-A-CH*. IFToMM. Innsbruck, Austria, 2016, pp. 11.1–11.8.

[Oel+16b] M. Oeljeklaus, F. Hoffmann, and T. Bertram. “Kontextmodellierung für Verkehrssituationen auf Grundlage von Kamerabildern”. In: *at-Automatisierungstechnik* 64.5 (2016), pp. 375–384.

[Oel+17] M. Oeljeklaus, F. Hoffmann, and T. Bertram. “A Combined Recognition and Segmentation Model for Urban Traffic Scene Understanding”. In: *Proceedings of the 20th International Conference on Intelligent Transportation Systems*. IEEE. Yokohama, Japan, 2017, pp. 2292–2297.

[Oel+18a] M. Oeljeklaus, F. Hoffmann, and T. Bertram. “A Fast Multi-Task CNN for Spatial Understanding of Traffic Scenes”. In: *Proceedings of the 21st International Conference on Intelligent Transportation Systems*. IEEE. Maui, USA, 2018, pp. 2825–2830.

[Oel+18b] M. Oeljeklaus, F. Hoffmann, and T. Bertram. “A Shared Encoder DNN for Integrated Recognition and Segmentation of Traffic Scenes”. In: *Studies in Computational Intelligence* 739 (2018), pp. 103–120.

[Oel+19a] M. Oeljeklaus, N. Stannartz, M. Schmidt, F. Hoffmann, and T. Bertram. “Fahrzeugdetektion mit stationären Kameras zur automatischen Verkehrsüberwachung”. In: *Tagungsband der 9. Fachtagung AUTOREG*. VDI/VDE-GMA. Mannheim, Germany, 2019, pp. 67–76.

[Oel+19b] M. Oeljeklaus, N. Stannartz, M. Schmidt, F. Hoffmann, and T. Bertram. "Fahrzeugdetektion mit stationären Kameras zur automatischen Verkehrsüberwachung". In: *Forschung im Ingenieurwesen* 83.2 (2019), pp. 163–171.

[Pin+15] P. O. Pinheiro, R. Collobert, and P. Dollár. "Learning to Segment Object Candidates". In: *Proceedings of the 29th International Conference on Neural Information Processing Systems*. MIT Press. Montréal, Canada, 2015, pp. 1990–1998.

[Pol64] B. T. Polyak. "Some methods of speeding up the convergence of iteration methods". In: *USSR Computational Mathematics and Mathematical Physics* 4.5 (1964), pp. 1–17.

[Pri12] S. J. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

[Ran13] M. Ranzato. "Large-Scale Visual Recognition With Deep Learning". In: *Tutorial Large-Scale Visual Recognition at the 26th Conference on Computer Vision and Pattern Recognition*. IEEE. Portland, USA, 2013.

[Rea+17] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin. "Large-Scale Evolution of Image Classifiers". In: *Proceedings of the 34th International Conference on Machine Learning*. MLR press. Sydney, Australia, 2017, pp. 2902–2911.

[Rea+19] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le. "Regularized evolution for image classifier architecture search". In: *Proceedings of the 33rd Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence. New York, USA, 2019, pp. 4780–4789.

[Red+16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the 29th Conference on Computer Vision and Pattern Recognition*. IEEE. Las Vegas, USA, 2016, pp. 779–788.

[Ren+17] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1137–1149.

[RF18] J. Redmon and A. Farhadi. "YOLOv3: An Incremental Improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[Rhu+16] M. Rhu, N. Gimelshein, J. Clemons, A. Zulfiqar, and S. W. Keckler. "vDNN: Virtualized Deep Neural Networks for Scalable, Memory-Efficient Neural Network Design". In: *Proceedings of the 49th Annual International Symposium on Microarchitecture*. IEEE. Taipei, Taiwan, 2016, pp. 1–13.

[Rod+19] T. Roddick, A. Kendall, and R. Cipolla. "Orthographic Feature Transform for Monocular 3D Object Detection". In: *Proceedings of the 30th British*

Machine Vision Conference. BMVA Press. Cardiff, United Kingdom, 2019, pp. 285.1–285.13.

[Rös+19] C. Rösener, J. Sauerbier, A. Zlocki, L. Eckstein, F. Hennecke, D. Kemper, and M. Oeser. "Potenzieller gesellschaftlicher Nutzen durch zunehmende Fahrzeugautomatisierung". In: *Berichte der Bundesanstalt für Straßenwesen. Unterreihe Fahrzeugtechnik* 128 (2019).

[Ros58] F. Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain". In: *Psychological Review* 65.6 (1958), pp. 386–408.

[Rud19] S. Ruder. "Neural Transfer Learning for Natural Language Processing". PhD thesis. National University of Ireland Galway, 2019.

[Rui+15] A. Ruiz, G. Juez, P. Schleiss, and G. Weiss. "A safe generic adaptation mechanism for smart cars". In: *Proceedings of the 26th International Symposium on Software Reliability Engineering*. IEEE. Gaithersburg, USA, 2015, pp. 161–171.

[Rus+15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252.

[RX15] J. S. Ren and L. Xu. "On Vectorization of Deep Convolutional Neural Networks for Vision Tasks". In: *Proceedings of the 29th Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence. Austin, USA, 2015, pp. 1840–1846.

[San+17] A. Sankar, X. Zhang, and K. C.-C. Chang. "Motif-based Convolutional Neural Network on Graphs". In: *arXiv preprint arXiv:1711.05697* (2017).

[Sch+10] D. Scherer, A. Müller, and S. Behnke. "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition". In: *Proceedings of the 20th International Conference on Artificial Neural Networks*. Springer. Thessaloniki, Greece, 2010, pp. 92–101.

[Sch+15] M. Schreier, V. Willert, and J. Adamy. "Compact Representation of Dynamic Driving Environments for ADAS by Parametric Free Space and Dynamic Object Maps". In: *Transactions on Intelligent Transportation Systems* 17.2 (2015), pp. 367–384.

[Sch+18] M. Schmidt, M. Oeljeklaus, C. Lienke, F. Hoffmann, M. Krüger, T. Nattermann, M. Mohamed, and T. Bertram. "Fahrspurerkennung mit Deep Learning für automatisierte Fahrfunktionen". In: *Proceedings of the 28th Workshop Computational Intelligence*. VDI/VDE-GMA. Dortmund, Germany, 2018, pp. 147–173.

[Sch12] M. Schreiber. "Konzeptionierung und Evaluierung eines Ansatzes zu einer manöverbasierten Fahrzeugführung im Nutzungskontext Autobahnfahrten". Doctoral dissertation. Technische Universität Darmstadt, 2012.

[Sch16] V. Schomerus. "Context-Supported Lane Estimation-Understanding the Scene by Learning Spatial Relations Between Semantic Features and Virtual Ground Truth". Doctoral dissertation. Technische Universität Braunschweig, 2016.

[See+16] C. Seeger, A. Müller, L. Schwarz, and M. Manz. "Towards Road Type Classification with Occupancy Grids". In: *Deep Driving-Learning Representations for Intelligent Vehicles Workshop of the 27th Intelligent Vehicles Symposium*. IEEE. Gothenburg, Sweden, 2016.

[Sen+12] S. Sengupta, P. Sturgess, L. Ladický, and P. H. Torr. "Automatic Dense Visual Semantic Mapping from Street-Level Imagery". In: *Proceedings of the International Conference on Intelligent Robots and Systems*. IEEE. Algarve, Portugal, 2012, pp. 857–862.

[Ser+14] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun. "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *Proceedings of the 2nd International Conference on Learning Representations*. Banff, Canada, 2014.

[Sha+17] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer". In: *Proceedings of the 5th International Conference on Learning Representations*. Toulon, France, 2017.

[Sha76] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[She+17] E. Shelhamer, J. Long, and T. Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Transactions on Pattern Analysis and Machine Intelligence* 39.4 (2017), pp. 640–651.

[Shu+16] B. Shuai, Z. Zuo, B. Wang, and G. Wang. "DAG-Recurrent Neural Networks For Scene Labeling". In: *Proceedings of the 29th Conference on Computer Vision and Pattern Recognition*. IEEE. Las Vegas, USA, 2016, pp. 3620–3629.

[Sik+14] I. Sikirić, K. Brkić, J. Krapac, and S. Šegvić. "Image Representations on a Budget: Traffic Scene Classification in a Restricted Bandwidth Scenario". In: *Proceedings of the 25th Intelligent Vehicles Symposium*. IEEE. Dearborn, USA, 2014, pp. 845–852.

[Sik+19] I. Sikirić, K. Brkić, P. Bevandić, I. Krešo, J. Krapac, and S. Šegvić. "Traffic Scene Classification on a Representation Budget". In: *Transactions on Intelligent Transportation Systems* (2019).

[Sim+19] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder. "Disentangling Monocular 3D Object Detection". In: *Proceedings of the 17th International Conference on Computer Vision*. IEEE. Seoul, South Korea, 2019, pp. 1991–1999.

[SJ19] I. Stančin and A. Jović. "An overview and comparison of free Python libraries for data mining and big data analysis". In: *Proceedings of the 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics*. IEEE. Opatija, Croatia, 2019, pp. 977–982.

[SK19] C. Shorten and T. M. Khoshgoftaar. "A survey on Image Data Augmentation for Deep Learning". In: *Journal of Big Data* 6.1 (2019), pp. 60.1–60.48.

[SL09] M. Sokolova and G. Lapalme. "A systematic analysis of performance measures for classification tasks". In: *Information Processing & Management* 45.4 (2009), pp. 427–437.

[Soc18] Society of Automotive Engineers International. "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles". In: *Annual Vehicle Electrification Subscription* J3016 (2018).

[Sti+15] C. Stiller, A. Bachmann, and A. Geiger. "Maschinelles Sehen". In: *Handbuch Fahrerassistenzsysteme*. Springer, 2015, pp. 369–393.

[SZ15] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the 3rd International Conference on Learning Representations*. San Diego, USA, 2015.

[Sze+15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going Deeper with Convolutions". In: *Proceedings of the 28th Conference on Computer Vision and Pattern Recognition*. IEEE. Boston, USA, 2015, pp. 1–9.

[Sze+16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the 29th Conference on Computer Vision and Pattern Recognition*. IEEE. Las Vegas, USA, 2016, pp. 2818–2826.

[Sze+17a] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer. "Efficient Processing of Deep Neural Networks: A Tutorial and Survey". In: *Proceedings of the IEEE* 105.12 (2017), pp. 2295–2329.

[Sze+17b] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the 31st Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence. San Francisco, USA, 2017, pp. 4278–4284.

[Tan+20] M. Tan, R. Pang, and Q. V. Le. "EfficientDet: Scalable and Efficient Object Detection". In: *Proceedings of the 33rd Conference on Computer Vision and Pattern Recognition*. IEEE. Virtual conference, 2020, pp. 10781–10790.

[Tei+18] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. "MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving". In: *Proceedings of the 29th Intelligent Vehicles Symposium*. IEEE. Changshu, China, 2018, pp. 1013–1020.

[TH12] T. Tieleman and G. Hinton. "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude". In: *COURSERA: Neural networks for machine learning* 4.2 (2012), pp. 26–31.

[Thr95] S. Thrun. "Is Learning The n-th Thing Any Easier Than Learning The First?" In: *Proceedings of the 8th International Conference on Neural Information Processing Systems*. MIT Press. Denver, USA, 1995, pp. 640–646.

[TL19] M. Tan and Q. V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *Proceedings of the 36th International Conference on Machine Learning*. MLR press. Long Beach, USA, 2019, pp. 6105–6114.

[Tra+15] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. "Learning Spatiotemporal Features with 3D Convolutional Networks". In: *Proceedings of the 15th International Conference on Computer Vision*. IEEE. Santiago, Chile, 2015, pp. 4489–4497.

[Uhr+16] J. Uhrig, M. Cordts, U. Franke, and T. Brox. "Pixel-level encoding and depth layering for instance-level semantic labeling". In: *Proceedings of the 38th German Conference on Pattern Recognition*. Springer. Hannover, Germany, 2016, pp. 14–25.

[Van+18] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran. "Autonomous vehicle perception: The technology of today and tomorrow". In: *Transportation Research Part C: Emerging Technologies* 89 (2018), pp. 384–406.

[Vog18] M. Vogt. "An overview of deep learning techniques". In: *at-Automatisierungstechnik* 66.9 (2018), pp. 690–703.

[Wan+19a] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger. "Pseudo-LiDAR From Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving". In: *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition*. IEEE. Long Beach, USA, 2019, pp. 8445–8453.

[Wan+19b] Z. Wang, Z. Cheng, H. Huang, and J. Zhao. "ShuDA-RFBNet for Real-time Multi-task Traffic Scene Perception". In: *Proceedings of the 5th Chinese Automation Congress*. IEEE. Hangzhou, China, 2019, pp. 305–310.

[Wei+13] G. Weiss, F. Grigoleit, and P. Struss. "Context Modeling for Dynamic Configuration of Automotive Functions". In: *Proceedings of the 16th International Conference on Intelligent Transportation Systems*. IEEE. The Hague, Netherlands, 2013, pp. 839–844.

[WG15] H. Wu and X. Gu. "Max-Pooling Dropout for Regularization of Convolutional Neural Networks". In: *Proceedings of the 22nd International Conference on Neural Information Processing*. Springer. Istanbul, Turkey, 2015, pp. 46–54.

[Win15] H. Winner. "Quo vadis, FAS?" In: *Handbuch Fahrerassistenzsysteme*. Springer, 2015, pp. 1167–1186.

[WK19] X. Weng and K. Kitani. "Monocular 3D Object Detection with Pseudo-LiDAR Point Cloud". In: *Workshops of the 17th International Conference on Computer Vision*. IEEE. Seoul, South Korea, 2019, pp. 857–866.

[Wu+17] B. Wu, A. Wan, F. Iandola, P. H. Jin, and K. Keutzer. "SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving". In: *Workshops of the 30th Conference on Computer Vision and Pattern Recognition*. IEEE. Honolulu, USA, 2017, pp. 129–137.

[Wu+19a] C. Wu, M. Tygert, and Y. LeCun. "A hierarchical loss and its problems when classifying non-hierarchically". In: *PLoS ONE* 14.12 (2019), e0226222.1–e0226222.17.

[Wu+19b] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu. "FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation". In: *arXiv preprint arXiv:1903.11816* (2019).

[Wu+19c] Z. Wu, C. Shen, and A. v. d. Hengel. "Wider or Deeper: Revisiting the ResNet Model for Visual Recognition". In: *Pattern Recognition* 90 (2019), pp. 119–133.

[XC18] B. Xu and Z. Chen. "Multi-level Fusion Based 3D Object Detection from Monocular Images". In: *Proceedings of the 31std Conference on Computer Vision and Pattern Recognition*. IEEE. Salt Lake City, USA, 2018, pp. 2345–2353.

[Xia+15] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. "Data-Driven 3D Voxel Patterns for Object Category Recognition". In: *Proceedings of the 28th Conference on Computer Vision and Pattern Recognition*. IEEE. Boston, USA, 2015, pp. 1903–1911.

[Xie+19] S. Xie, H. Zheng, C. Liu, and L. Lin. "SNAS: stochastic neural architecture search". In: *Proceedings of the 7th International Conference on Learning Representations*. New Orleans, USA, 2019.

[Yan+19] B. Yang, G. Bender, Q. V. Le, and J. Ngiam. "CondConv: Conditionally Parameterized Convolutions for Efficient Inference". In: *Proceedings of the 33rd Conference on Neural Information Processing Systems*. Vancouver, Canada, 2019, pp. 1307–1318.

[YH17] Y. Yang and T. M. Hospedales. “Trace Norm Regularised Deep Multi-Task Learning”. In: *Workshops of the 5th International Conference on Learning Representations*. Toulon, France, 2017.

[YK16] F. Yu and V. Koltun. “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *Proceedings of the 4th International Conference on Learning Representations*. San Juan, Puerto Rico, 2016.

[Yos+14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. “How transferable are features in deep neural networks?” In: *Proceedings of the 28th Conference on Neural Information Processing Systems*. Vol. 27. MIT Press. Montréal, Canada, 2014, pp. 3320–3328.

[Zha+17] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. “Pyramid Scene Parsing Network”. In: *Proceedings of the 30th Conference on Computer Vision and Pattern Recognition*. IEEE. Honolulu, USA, 2017, pp. 2881–2890.

[Zha+18] X. Zhang, X. Zhou, M. Lin, and J. Sun. “ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices”. In: *Proceedings of the 31std Conference on Computer Vision and Pattern Recognition*. IEEE. Salt Lake City, USA, 2018, pp. 6848–6856.

[Zho+15] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. “Object Detectors Emerge in Deep Scene CNNs”. In: *Proceedings of the 3rd International Conference on Learning Representations*. San Diego, USA, 2015.

[Zhu+19] R. Zhu, S. Zhang, X. Wang, L. Wen, H. Shi, L. Bo, and T. Mei. “ScratchDet: Training Single-Shot Object Detectors From Scratch”. In: *Proceedings of the 32nd Conference on Computer Vision and Pattern Recognition*. IEEE. Long Beach, USA, 2019, pp. 2268–2277.

[Zie+14] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, et al. “Making Bertha Drive — An Autonomous Journey on a Historic Route”. In: *Intelligent Transportation Systems Magazine* 6.2 (2014), pp. 8–20.

[ZL17] B. Zoph and Q. V. Le. “Neural Architecture Search with Reinforcement Learning”. In: *Proceedings of the 5th International Conference on Learning Representations*. Toulon, France, 2017.

[Zop+18] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. “Learning Transferable Architectures for Scalable Image Recognition”. In: *Proceedings of the 31st Conference on Computer Vision and Pattern Recognition*. IEEE. Salt Lake City, USA, 2018, pp. 8697–8710.

Related Peer-Reviewed Publications

Publications in the context of this dissertation:

M. Oeljeklaus, N. Stannartz, M. Schmidt, F. Hoffmann, and T. Bertram. “Fahrzeugdetektion mit stationären Kameras zur automatischen Verkehrsüberwachung”. In: *Forschung im Ingenieurwesen* 83.2 (2019), pp. 163–171.

M. Oeljeklaus, N. Stannartz, M. Schmidt, F. Hoffmann, and T. Bertram. “Fahrzeugdetektion mit stationären Kameras zur automatischen Verkehrsüberwachung”. In: *Tagungsband der 9. Fachtagung AUTOREG*. VDI/VDE-GMA. Mannheim, Germany, 2019, pp. 67–76.

M. Oeljeklaus, F. Hoffmann, and T. Bertram. “A Fast Multi-Task CNN for Spatial Understanding of Traffic Scenes”. In: *Proceedings of the 21st International Conference on Intelligent Transportation Systems*. IEEE. Maui, USA, 2018, pp. 2825–2830.

M. Oeljeklaus, F. Hoffmann, and T. Bertram. “A Shared Encoder DNN for Integrated Recognition and Segmentation of Traffic Scenes”. In: *Studies in Computational Intelligence* 739 (2018), pp. 103–120.

M. Oeljeklaus, F. Hoffmann, and T. Bertram. “A Combined Recognition and Segmentation Model for Urban Traffic Scene Understanding”. In: *Proceedings of the 20th International Conference on Intelligent Transportation Systems*. IEEE. Yokohama, Japan, 2017, pp. 2292–2297.

M. Oeljeklaus, F. Hoffmann, and T. Bertram. “Kameragestützte Segmentierung von Verkehrsszenen für automatisierte Fahrzeugsysteme”. In: *Tagungsband der 2. Fachtagung IFToMM D-A-CH*. IFToMM. Innsbruck, Austria, 2016, pp. 11.1–11.8.

M. Oeljeklaus, F. Hoffmann, and T. Bertram. “Kontextmodellierung für Verkehrssituationen auf Grundlage von Kamerabildern”. In: *at-Automatisierungstechnik* 64.5 (2016), pp. 375–384.

M. Oeljeklaus, F. Hoffmann, and T. Bertram. “Bildbasierte Detektion von Kontextinformationen über Verkehrssituationen”. In: *Tagungsband der 7. Fachtagung AUTOREG*. VDI/VDE-GMA. Baden-Baden, Germany, 2015, pp. 243–254.

M. Oeljeklaus, F. Hoffmann, and T. Bertram. “Convolutional Neural Networks für die semantische Segmentierung von Szenen”. In: *Proceedings of the 25th Workshop Computational Intelligence*. VDI/VDE-GMA. Dortmund, Germany, 2015, pp. 241–254.

M. Oeljeklaus, F. Posada, F. Hoffmann, and T. Bertram. “Analyse globaler Bildmerkmale zur Klassifikation von Verkehrsszenen”. In: *Proceedings of the 24th Workshop Computational Intelligence*. VDI/VDE-GMA. Dortmund, Germany, 2014, pp. 299–314.

M. Schmidt, M. Krüger, C. Lienke, M. Oeljeklaus, T. Nattermann, M. Mohamed, F. Hoffmann, and T. Bertram. “Fahrstreifenerkennung mit Deep Learning für automatisierte Fahrfunktionen”. In: *at-Automatisierungstechnik* 67.10 (2019), pp. 866–878.

M. Schmidt, M. Oeljeklaus, C. Lienke, F. Hoffmann, M. Krüger, T. Nattermann, M. Mohamed, and T. Bertram. “Fahrspurerkennung mit Deep Learning für automatisierte Fahrfunktionen”. In: *Proceedings of the 28th Workshop Computational Intelligence*. VDI/VDE-GMA. Dortmund, Germany, 2018, pp. 147–173.

Additional Peer-Reviewed Publications

M. Oeljeklaus, C. Rösmann, F. Hoffmann, and T. Bertram. “Trajektorienplanung mit Timed-Elastic-Bands für die proxemische Interaktion zwischen Menschen und mobilen Robotern”. In: *Proceedings of the 23rd Workshop Computational Intelligence*. VDI/VDE-GMA. Dortmund, Germany, 2013, pp. 385–400.

A. Hugenroth, F. Albers, M. Oeljeklaus, and T. Bertram. “Object Classification in the Fixation of a Car Driver”. In: *Proceedings of the 11th Symposium Automotive meets Electronics*. VDE/VDI-GMM. 2020, pp. 20–25.

M. Lütkemöller, M. Oeljeklaus, T. Bertram, K. Rink, U. Stählin, and R. Grewe. “Derivation and Application of an Observer Structure to Detect Inconsistencies Within a Static Environmental Model”. In: *Tagungsband der 4. Internationalen ATZ-Fachtagung Automatisiertes Fahren*. Wiesbaden, Germany: Springer, 2018, pp. 67–79.

C. Rösmann, M. Oeljeklaus, F. Hoffmann, and T. Bertram. “Online Trajectory Prediction and Planning for Social Robot Navigation”. In: *Proceedings of the International Conference on Advanced Intelligent Mechatronics*. IEEE. Munich, Germany, 2017, pp. 1255–1260.

Supervised Theses

N. Gratza. “Detektion, Klassifizierung und Rekonstruktion von Verkehrsleitkegeln im Raum am Beispiel eines Formula Student Rennwagens”. Bachelor’s thesis. TU Dortmund University, 2019.

R. Beckmann. “Entwicklung von Entscheidungsalgorithmen für das automatisierte Fahren”. Master’s thesis. TU Dortmund University, 2018.

Z. Hu. “Instance segmentation for automotive environment modelling”. Master’s thesis. TU Dortmund University, 2018.

A. Hugenroth. “Objektklassifikation in der Fixation des Fahrers”. Master’s thesis. TU Dortmund University, 2018.

B. Möllenbeck. “Bildanalyse und Umfeldmodellierung zur Erkennung von Fahrzeugen an Autobahn- und Parkplatzzufahrten”. Master’s thesis. TU Dortmund University, 2018.

N. Smolnikow. “Tiefe neuronale Netze zur Beobachtung von Inkonsistenzen in einer radarbasierten, statischen Umgebungserfassung”. Master’s thesis. TU Dortmund University, 2018.

D. Jiang. “Deep residual networks for causal semantic segmentation of traffic scene videos”. Master’s thesis. TU Dortmund University, 2017.

B. Polenz. “Rekonstruktion und Modellierung von Verkehrsszenen aus monokularen Ansichten”. Master’s thesis. TU Dortmund University, 2017.

H. u. M. Riaz. “Deep Learning for Mobile Robot Navigation”. Master’s thesis. TU Dortmund University, 2017.

M. Schmidt. “Online-Kalibrierung eines Kamerasystems in Straßenverkehrsszenarien zur Validierung automatisierter Fahrfunktionen”. Master’s thesis. TU Dortmund University, 2017.

M. Waldner. “Modellprädiktive Regelung der Vorfeldausleuchtung eines Kraftfahrzeugs bei verschiedenen Fahrszenarien”. Master’s thesis. TU Dortmund University, 2017.

T. Kuhl. “Bildverarbeitungsgestützte Bestimmung der Umgebungshelligkeit mittels einer Fahrerassistenzkamera”. Master’s thesis. TU Dortmund University, 2016.

M. Lütkemöller. “Herleitung einer domänenspezifischen Beobachterstruktur für die Betriebssicherheit und den Betriebsschutz am Beispiel eines Umfeld-Modells für Fahrerassistenzsysteme”. Master’s thesis. TU Dortmund University, 2016.

M. Rühl. “Selektive Suche für die Kamerabasierte Verkehrszeichendetektion”. Bachelor’s thesis. TU Dortmund University, 2015.

P. Schulze Vohren. “Bewertung von Schwingungen an Nutzfahrzeugen”. Bachelor’s thesis. TU Dortmund University, 2015.

C. M. Treviño Campa. “Adapting pre-trained convolutional neural network models for traffic scene labeling”. Master’s thesis. TU Dortmund University, 2015.

M. E. M. Hassan. “Advanced Self-Tuning Controllers for Compliant Motion Control Solutions with Friction”. Master’s thesis. TU Dortmund University, 2014.

P. Weyers. “Kamerabasierte Umfelderfassung zur Interpretation von Verkehrssituationen”. Bachelor’s thesis. TU Dortmund University, 2014.