Technische Universität München

TUM School of Engineering and Design

Deep Learning-based Radar, Camera, and Lidar Fusion for Object Detection

Felix Otto Geronimo Nobis, M. Sc.

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitz:	Prof. Dr.-Ing. Boris Lohmann
Prüfer*innen der Dissertation:	1. Prof. Dr.-Ing. Markus Lienkamp
	2. Prof. Dr.-Ing. Klaus Dietmayer

Die Dissertation wurde am 09. November 2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 28. Februar 2022 angenommen.

Contents

List of Abbreviations ..... **III**

1 Introduction **1**

1.1 Motivation **1**

1.2 Automotive Perception **1**

1.3 3D Object Detection **3**

2 Related Work **5**

2.1 Automotive Object Detection Data Sets **5**

2.1.1 Overview 5

2.1.2 nuScenes Data Set 7

2.2 Evaluation Metrics **7**

2.3 Object Detection **11**

2.3.1 Camera-based 12

2.3.2 Lidar-based 16

2.3.3 Radar-based 18

2.3.4 Data Fusion-based 22

2.3.5 Summary of Methods 27

3 Problem Statement **29**

3.1 Conclusions from Related Work **29**

3.2 Research Questions **30**

4 Algorithm Development **33**

4.1 Radar Point Cloud Segmentation **33**

4.2 Radar and Camera Fusion for Object Detection **55**

4.3 Radar, Camera, and Lidar Fusion for Object Detection **64**

5 Discussion **83**

5.1 Research Questions **83**

5.2 Practical Relevance **88**

5.3 Outlook **90**

5.3.1 Future Work 90

5.3.2 Radar Hardware and Data 91
5.3.3 Object Detection 92
6 Summary **95**
List of Figures ..... **i**
List of Tables ..... **iii**
Bibliography ..... **v**
Prior Publications ..... **xxv**
Supervised Students' Theses ..... **xxvii**

List of Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
ACC	Adaptive Cruise Control
AOE	Average Orientation Error
AP	Average Precision
AV	Autonomous Vehicle
AVOD	Aggregate View Object Detection
BEV	Bird's-Eye View
CAN	Controller Area Network
CFAR	Constant False Alarm Rate
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CRF-Net	Camera Radar Fusion-Net
CV	Computer Vision
DARPA	Defense Advanced Research Projects Agency
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DNN	Deep Neural Network
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continuous Wave
FN	False Negative
FOV	Field of View
FP	False Positive
GNSS	Global Navigation Satellite System
GRIF	Gated Region of Interest Fusion
HD	High Definition
IMU	Inertial Measurement Unit
IoU	Intersection over Union
KPConv	Kernel Point Convolution
KPLSTM	Kernel Point LSTM
lidar	Light Detection and Ranging
LSTM	Long Short-Term Memory
mAP	mean Average Precision
ODD	Operational Design Domain
PTP	Precision Time Protocol
RA	Range-Azimuth
RAD	Range-Azimuth-Doppler
radar	Radio Detection and Ranging
RCS	Radar Cross-Section

RD	Range-Doppler
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue, Depth
ROI	Region of Interest
RVF-Net	Radar Voxel Fusion-Net
SECOND	Sparsely Embedded Convolutional Detection
SLAM	Simultaneous Localization and Mapping
TN	True Negative
TP	True Positive

1 Introduction

1.1 Motivation

The development of Autonomous Vehicles (AVs) impacts society in a plethora of areas, stretching beyond the car manufacturing industry [1]. When the driving task is no longer performed by a human driver, time and space is freed to create a new form of transportation experience, e.g., allowing for relaxation or work while being on the road [2]. At the same time, automation enables individual mobility for a greater audience, such as children [3], the elderly [4], and people with disabilities [4]. AVs will transform our cities and our ways of labour, just as prior revolutions in the mobility sector, such as railroads and motor vehicles, have done in the past [2]. Studies project large economic advantages resulting from the development of AVs [5–8]. Clements and Kockelman [6] estimate the annual economic benefit per person in the United States at $3814.

Legal questions [9], ethical considerations [10], and technical challenges need to be solved for making autonomous driving a reality. Despite private companies operating AVs on public roads, current technology does not yet enable safe autonomous driving, even in restricted Operational Design Domains (ODDs). Recent reports of accidents and failures of Waymo [11], Tesla [12], and Uber [13] vehicles show the fragility of current automated vehicle systems. While current AVs have reached a noteworthy level of maturity, the complexity and diversity of unforeseen situations that happen on the roads every day overstrain the capabilities of today's most advanced autonomous driving softwares. With the goal of deploying mature autonomous driving functions to the mass market, researchers and companies around the globe work on new methods to make AVs more robust and reliable.

Autonomous driving software can be divided into the categories perception, planning, and control [14, p. 3]. Van Brummelen et al. [15, p. 1] point out that improving the perception system is a vital factor for enabling robust, reliable, and safe autonomous driving. All consecutive functions can only operate as specified if an accurate environment model is provided by the perception algorithms. This thesis, therefore, investigates this critical element of autonomous driving in more detail.

1.2 Automotive Perception

Automotive perception refers to all tasks that deal with gathering information from the environment to create an internal model for the driving task [14, p. 3]. Pendleton et al. [14, p. 3] further divide the perception task into (self-)localization, road sign detection, semantic segmentation, e.g., for drivable space estimation, and object detection. The subdivision of the perception task and its input and output data is shown in Figure 1.1.

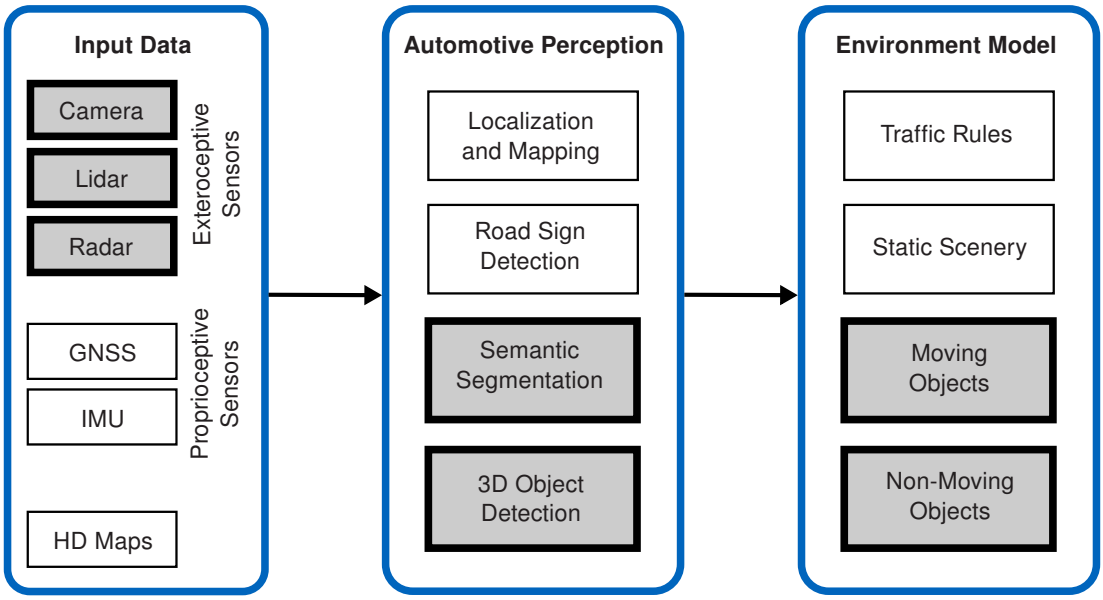


Figure 1.1: Subdivision of the perception task for autonomous vehicles. The topics in the grey boxes are the focus of this thesis.

The main input data for perception tasks are provided by exteroceptive sensors which collect information from the environment [16]. For autonomous driving applications, camera, Radio Detection and Ranging (radar), and Light Detection and Ranging (lidar) sensors are the most widely used exteroceptive sensors. Additional information is provided by proprioceptive sensors such as Global Navigation Satellite Systems (GNSSs) and Inertial Measurement Units (IMUs) which collect information about the vehicle state [16]. High Definition (HD) Maps provide additional information about the surroundings recorded from specialized mapping vehicles [17]. Arbitrary subsets of this input data can be used for the individual perception tasks:

- (Self-)localization of a vehicle from perception sensor input is a complex task and subject to ongoing research [18, 19]. Localization can be performed on a prior existing map or simultaneously to creating the map, called Simultaneous Localization and Mapping (SLAM) [20–22]. In certain environments the importance of perception-based localization can be mitigated by using the localization output of a combination of a high accuracy GNSS and IMU system with centimeter scale accuracy [23].
- Road sign detection algorithms are fairly mature and employed in production vehicles [24–26]. These algorithms enable AVs to adhere to local traffic rules. Road sign detection is a special use-case of Two-Dimensional (2D) object detection [27–29]. 2D object detection, e.g. of cars, is not listed as its own category. The 2D position estimation of a 2D object detection algorithm in the image space only serves as an intermediate result for Three-Dimensional (3D) object detection in the autonomous driving use-case.
- Semantic segmentation can be subdivided in the 2D segmentation of image data [30, 31] and 3D segmentation of the environment, e.g. in the form of point clouds [32–34]. In the 2D case, a class label for each pixel in the input image is predicted. In the 3D case, the prediction assigns a class label for all input points from the 3D space. Both tasks pose challenges for autonomous driving. Semantic segmentation

can serve as an intermediate result for object detection or provide a free space detection in the 3D case.

- 3D object detection [35–37] localizes moving and non-moving (but movable) objects in the 3D environment. While 2D detection in the image space is fairly mature, 3D object detection poses a major challenge in the development of autonomous vehicles [38].

A multitude of challenges exists in the field of perception. To narrow down the research field, this thesis focuses on 3D object detection with the sub tasks of 2D object detection and 3D semantic segmentation. The grey boxes in Figure 1.1 represent the focus of this thesis.

1.3 3D Object Detection

In public automotive perception data sets such as KITTI [39] and nuScenes [40], the object detection task is defined as the estimation of the 3D pose (position and orientation) and dimensions of objects in the surroundings, paired with a classification of these objects. An object detection method for moving vehicles needs to provide high accuracy results in real time. Due to the continuously changing poses of surrounding objects, it has to be especially precise and robust since errors cannot easily be mitigated by optimizing detections over several time steps, which is for example applied for map generation through loop closure [22].

Three sensor modalities are widely used to perform object detection for AVs. Camera and radar sensors are used in series production vehicles, e.g. for Adaptive Cruise Control (ACC) applications. Due to their high resolution and 3D capabilities, lidar sensors are mainly applied in research vehicles [41]. The higher cost and package size as well as the lower range and mechanical robustness have hindered the broad success in the mass market up to this point [42].

Currently, the most effective object detection methods use Deep Neural Networks (DNNs) to generate the final object estimations [39, 40]. Deep learning-based methods are a subgroup of machine learning methods [43, 44] and require extensive amounts of data and computational resources to generate effective algorithms. The basics of deep learning are presented in additional literature [45, 46].

Each sensor modality comes with different strengths, weaknesses, and data representations. Combining the data of different sensor modalities to achieve a higher (object detection) performance is called *sensor data fusion* or simply *fusion*. A fusion increases the total amount of available input data and is viewed as a promising research direction to increase the overall object detection result [35]. It can help to compensate individual sensor failures and enable detections in inclement weather conditions, e.g., fog, night, challenging illumination or snowy conditions when individual sensor data quality may be deteriorated [47–52]. The data of different sensor inputs can be fused at different abstraction levels from a low- to a high-level fusion. Despite the active research in this field, no best practice fusion method for object detection has been established so far [35, p. 1354].

Recently, methods that perform a low-level sensor data fusion with deep learning networks have gained more interest [35, 38, 53]. This low-level raw data fusion enables the usage of weak features—such as the barely visible outline of a vehicle in a camera image recorded in fog, together with weak radar signals—for obtaining a high-confidence detection result. Such

weak features could not be used jointly by processing the sensor modalities individually and performing a high-level fusion. The principles of different fusion methods are discussed in more detail in Chapter 2.

At the beginning of writing this thesis no low-level fusion methods involving radar data were published. To close this knowledge gap in the literature, this thesis is among the first works to develop and evaluate low-level object detection fusion methods with radar data being part of the input data. The processing methods are inspired by literature on camera and lidar object detection and data fusion.

An additional focus of the thesis is on object detection in inclement weather conditions which is an unsolved problem in the automotive sector. As radar sensors are relatively robust to environmental conditions, object detection in inclement weather poses a suitable use-case to evaluate the potential of sensor fusion algorithms involving radar data.

2 Related Work

This chapter presents literature in the context of object detection for autonomous driving as a prerequisite for defining a knowledge gap in the state of the art later in Chapter 3. Section 2.1 first reviews public data sets and their suitability to train and evaluate object detection algorithms. Performance metrics to evaluate semantic segmentation and object detection models are presented in Section 2.2. On these foundations, Section 2.3 reviews algorithms from these fields. The algorithms most relevant to this thesis are summarized in Section 2.3.5.

2.1 Automotive Object Detection Data Sets

As the foundation for any machine learning-based algorithm, an adequate data source is imperative for the successful development of DNNs [54, 55]. Training data are used to adapt a DNN to a specific task. Independent test data are used to evaluate the trained network. A realistic distribution of the data in the test set is needed so that the evaluation allows to draw conclusions for the intended future application, otherwise the evaluation only provides a theoretical benchmark on pre-selected test data. By selecting inadequate test data, evaluation scores close to or at 100 % can be achieved while the same algorithm performs poorly during the real use-case. As the data themselves arguably have the biggest impact on the performance of deep learning algorithms, publicly available data sets are imperative to enable a scientific, empirical evaluation and comparison of different deep learning algorithms.

As data can be valuable assets to companies, research data are often kept private to the organization. On the other hand, publicly available data enables additional research in the field of interest of the creator of the data. These results might create advances in the development that otherwise would not have been possible. Due to the complexity of object detection algorithm development, more and more universities and companies decide to publish their data to profit from the resulting development on their data.

The following section gives an overview of publicly available data sets for object detection in the automotive context. Section 2.1.2 analyses the nuScenes data set—which is used in the publications written for this thesis—in more depth.

2.1.1 Overview

A multitude of publicly available data sets exist for object detection in an automotive context [56]. Radar data are least available to the public and therefore a critical selection criterion for identifying a suitable data set for this thesis. Table 2.1 gives an overview of all automotive localization and object detection data sets known to the author that contain some form of radar data. Selected popular object detection data sets that do not contain radar data are added

to provide a more comprehensive overview. The data sets are divided into four subgroups distinguished by the different characteristics of the radar data.

Table 2.1: Overview of publicly available autonomous driving perception data sets. The table focuses on data sets which contain some form of radar data.

	Group	Data Set	Lidar	Camera	Radar	Notes	Year
no radar data	1	KITTI [39]	✓	✓	✗		2012
		Argoverse [57]	✓	✓	✗		2019
		Lyft [58]	✓	✓	✗		2019
		A2D2 [59]	✓	✓	✗		2020
		Waymo [60]	✓	✓	✗		2020
no object ground truth	2	EU Long-term [61]	✓	✓	✓		2019
		Oxford Radar RobotCar [62]	✓	✓	✓		2020
		MulRan [63]	✓	✗	✓		2020
		OLIMP [64]	✗	✓	✓		2020
radar cube data	3	CARRADA [65]	✗	✓	✓	staged scenarios	2020
		RADIATE [66]	✓	✓	✓	no range rate data	2020
		CRUW [67]	✗	✓	✓	no range rate data	2021
		RaDICAL [68]	✗	✓	✓	experimental radar	2021
point cloud radar data	4	nuScenes [40]	✓	✓	✓	low radar resolution	2019
		DENSE [50]	✓	✓	✓	very low radar resolution	2019
		Astyx HiRes* [69]	✓	✓	✓	low amount of samples	2019
		RadarScenes [70]	✗	✓	✓	no bounding boxes	2021

*no longer available online

The first group of data sets in the table provides a variety of scenes recorded with lidar and camera sensors. These data sets are widely adopted in research but do not provide any radar data.

The second group of data sets in the table provides some form of radar data. However, no object ground truth annotations are available for these data sets, making them only suitable for self-localization but not for object detection tasks.

The third group of data sets in the table provides radar cube data which are outside the scope of the development presented in this thesis. A more detailed introduction of the radar data abstraction levels will be given in Section 2.3.3. Some of these data sets further lack radar range rate information or only comprise a low amount of variety in the scenes rendering them unfit for this thesis.

The fourth group of data sets in the table provides object ground truth annotations and point cloud radar data. The nuScenes [40] data set provides camera, lidar, and series production radar data as well as object bounding box annotations. However, the radar data resolution in this data set is lower than expected from the used sensor type [71–73]. The even lower radar sensor resolution of the DENSE data set [50] can be used for moving object detection for ACC applications but limits the usability for general object detection of both, moving and non-moving objects. The Astyx HiRes data set [69] uses an experimental high resolution 3D radar sensor. However, the data set only comprises 546 labeled frames. The amount of data is too small to train DNNs and test the generalization of the final algorithms. Further, as of August 2021, the data set is no longer available online. The RadarScenes data set [70] uses four series production radar sensors. The radar point cloud is semantically labeled with ground truth classes for radar points of moving objects. Point reflections from non-moving vehicles are collected in a background class. Object Bounding box annotations and lidar data are not provided.

Due to the wide availability of data sets for lidar and camera data, object detection approaches based on these sensors are predominantly evaluated on public data sets for better comparability. The overview in Table 2.1 shows that no public data set fulfills the requirements for the development of deep learning methods using radar data without any restrictions. Radar research is therefore often evaluated on non-public data sets as no adequate public data source might be available for the respective research scope. However, an evaluation on non-public data cannot replace the evaluation on public data. A shared public data source is imperative to enable a fair comparison of the results obtained by different research groups. For this thesis, the public nuScenes data set [40] offers the best compromise for the development of radar-centric sensor fusion algorithms on production radar data for object detection. The data are described in more detail in the following section.

2.1.2 nuScenes Data Set

The nuScenes data set is recorded in the cities of Boston (United States of America) and Singapore. It contains 1,000 scenes of a length of around 20 s each. The data are labeled at 2 Hz leading to a total of 40,000 labeled sensor data frames. Temporal interjacent sensor data are available, additional labels are created for these data through interpolation. One lidar sensor, six cameras and five radar sensors provide the perception data. To maintain an accurate spatial calibration, the recording vehicles are re-calibrated two times a week during the original recording period.

The Frequency Modulated Continuous Wave (FMCW) radar sensors operate in a short range interval up to 70 m and a far range interval up to 250 m. The sensor distinguishes the distance of point targets at a resolution of 0.4 m in the short range and 1.8 m in the long range. The angular resolution is lowest with 12.3° at the lateral edges of the short range Field of View (FOV) and highest with 3.2° at the centre of the the long range FOV [74].

The data set provides data from diverse scenarios. About 19 % of the scenes were recorded in rain conditions, 12 % of the scenes were recorded at night time [75, p. 25]. The data set contains a total of around 1,400,000 objects from 23 classes [40]; around 490,000 of the total objects are cars [76]. Objects are labeled with additional attributes. Vehicles for example are further distinguishable by being in the state *moving*, *stopped*, or *parked* at the current frame. These attributes can be used to build sub data sets, e.g. to train an object detection DNN for moving vehicles only, as done for the ACC use-case.

Newly proposed object detection algorithms are often evaluated and compared to other algorithms on this data set. According to Google Scholar, the nuScenes paper has been cited over 760 times as of August 2021 [77], indicating its widespread usage and reputation as a resource for algorithm comparison.

However, the radar data quality in the nuScenes data is limited. The limited data quality is a result of both the radar configuration as well as the labeling process which focuses on lidar data, resulting in a lot of missing labels for the radar data [72, pp. 14-17].

2.2 Evaluation Metrics

Due to the influence of the data on the performance of deep learning models, a quantitative comparability of different methods is only possible to prior work that has been evaluated on

the same data. With a fixed known data source, the definition of performance metrics is a prerequisite to obtain evidence-based performance results for different methods.

While deep learning methods stand out due to their performance advantages over rule-based approaches for object detection, the inner workings and limits of the resulting algorithms cannot be explained thoroughly [78, 79]. Even though the deep learning networks presented in this thesis incorporate priors from physics—such as the structure of the three dimensional euclidean space and the dimensions of possible objects—the explainability of such networks remains subject to current research [80, 81]. The selection of appropriate evidence-based metrics is therefore crucial to evaluate deep learning-based methods.

Accuracy is a metric which measures the ratio between correct classifications to all classifications. Due to high class imbalances in the underlying sensor data, such a metric can generate misleading performance scores. In common traffic scenarios the majority of radar point targets originate from background objects and not from objects of interest, e.g. moving vehicles. A machine learning model which classifies all points as background, regardless of their features, could therefore achieve a high accuracy. By assigning all points to the background class, a radar data-based semantic segmentation DNN could achieve 97 % accuracy on the nuScenes data set by predicting all background points correctly, even though it did not classify a single car correctly [72]. The thesis therefore uses four composed metrics which allow for a robust and realistic comparison of different methods:

- Confusion Matrix,
- F1-score,
- 2D mean Average Precision (mAP),
- 3D mAP.

The metrics are briefly summarized to facilitate the understanding of the evidence-based evaluation of the networks presented in the state of the art and the development chapter of this thesis.

Confusion Matrix

In this thesis, confusion matrices [43, p. 209] are constructed to analyze the output of a semantic segmentation model. Each element is added to the two dimensional matrix depending on its ground truth class and the predicted class assigned by the model that is evaluated. Figure 2.1 shows an example confusion matrix for a classifier that is used to distinguish whether radar point cloud targets originate from a moving vehicle or from any other surface. In this example, 612 radar points are correctly classified as moving point targets. As this is the class of interest for the underlying task, these points are counted as True Positives (TPs). The majority of the background points are correctly classified as background, named True Negative (TN). Five moving vehicle points are wrongly classified as background, these are the False Negative (FN) cases. The example classifier confuses 100 background points to be moving vehicles, named False Positive (FP). The confusion matrix gives a visual overview of the performance of the classifier and can be compared to confusion matrices of alternative models. In practice, the confusion matrix can also be used to fine-tune the parameterization of the machine learning model, e.g. lowering the weight for the moving vehicle class in the training process to lower the amount of FP cases.

F1-Score

While the confusion matrix gives a fine-grained insight into the performance of a classification

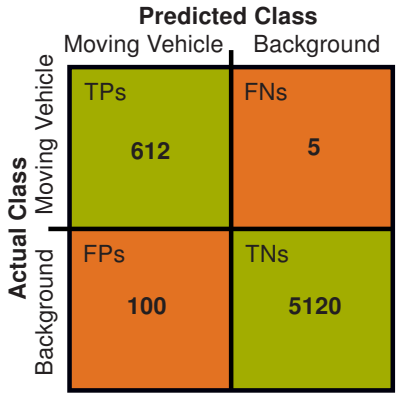


Figure 2.1: Confusion matrix: Classification of radar point cloud targets as moving vehicles.

model, it can be too detailed to compare a large amount of different classification models amongst each other. For such a comparison, the elements of the confusion matrix can be aggregated to high-level metric values that allow for a numerical comparison of different models. In our example, the *Recall* metric measures the amount of correctly classified moving vehicle point targets in relation to the amount of moving vehicle point targets in the data set. The *Precision* metric measures the amount of correctly classified moving vehicle point targets in relation to all point targets classified as a moving vehicle. Mathematically, the metrics are defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2.2}$$

These metrics are further aggregated to their harmonic mean which is called the F1-score and defined as [43, p. 397]:

$$F_1 = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.3}$$

The equal weighting of precision and recall in the F1-score is justified for the example task, as both, not-detected vehicles and wrongly detected vehicles in an empty space, can lead to hazardous reactions of an automated vehicle. The F1-score can be augmented to multiple classes by calculating a score for each class separately and calculating the mean over all classes. In this way underrepresented classes contribute the same amount to the final result as the majority class.

2D mean Average Precision

The task of object detection cannot be evaluated with the means of the confusion matrix or the F1-score. In addition to the classification, the predicted dimensions and the pose of the detected objects need to be considered for a comparison to the ground truth. The 2D mAP score considers all these factors for object detection on image data. This score is predominantly used in the literature for the comparison of different object detection models. The mAP is aggregated by averaging the Average Precision (AP) of all individual classes present in the data set. The composition of the class specific APs is more complex. It is summarized by performing the following steps [82]:

- First, one has to define the criteria for when a bounding box is considered as being detected correctly. For this, the classes of a ground truth bounding box and the predicted box have to coincide. Furthermore, some spatial proximity and similarity measure of the boxes has to be defined and reached. For 2D object detection, this measure is an Intersection over Union (IoU)-threshold of the overlap of both boxes. Commonly, IoU-thresholds of 0.5 or 0.7 are used to define a correct prediction.
- All predicted boxes of a class are put in a descending order sorted by the respective confidence score of their classification from 1 (sure) to 0 (unsure). An ideal classifier would predict all TPs with high confidence and all TNs with a confidence of 0 for the current class. In reality, this will rarely be the case. For the real world application, a classification confidence threshold will be defined to consider only predictions surpassing this threshold as output predictions by the model. This threshold leads to a specific precision and recall score combination for the class. Defining a lower classification confidence threshold normally leads to a precision decrease as more false positives with low confidence are predicted. At the same time the recall rises, as also some more difficult examples might be detected correctly with low classification confidence scores that were excluded before.
- The AP calculation averages different classification confidence thresholds to describe the performance of the model from a universal point of view independent of the confidence threshold applied in practice. For this, the AP is calculated as the area under the curve of precision-recall combinations for selected classification score thresholds.

An example ordering by classification score is shown in Table 2.2. The associated precision-recall curve is visualized in Figure 2.2. The scores are not visible in the figure of the precision recall-curve. In the example, two bounding boxes are correctly predicted with scores of 0.9 and 0.4. While two additional objects are wrongly predicted with a score of 0.8 and 0.7. The score threshold should be chosen at a score of 0.9 or 0.4 to maximize either precision or recall.

Table 2.2: Bounding box detection ordering for AP calculation. Inspired by Hui [82].

Rank	Confidence Score	True Positive	Precision	Recall
1	0.9	✓	1.0	0.5
2	0.8	✗	0.5	0.5
3	0.7	✗	0.33	0.5
4	0.4	✓	0.5	1.0

The AP performance metric can reach values between 0 and 1, where 1 would be a correct prediction of all instances. Slightly different parameterizations are used for the calculation of the mAP score depending on which data set or competition is considered. These definitions vary on the choice of the IoU-threshold and the approximation of the area under the precision-recall curve [82].

3D mean Average Precision

The calculation of the 3D mAP is analogous to the calculation of the 2D mAP. However, in the definition of nuScenes, the IoU-threshold is replaced with a distance threshold to match positive bounding boxes [83]. Additional work argues that an IoU-threshold might lead to undesired bounding box matches [84]. To capture both precise and rough object detections, the nuScenes AP definition considers four different distance thresholds ranging from 0.5 m–4.0 m. In practice,

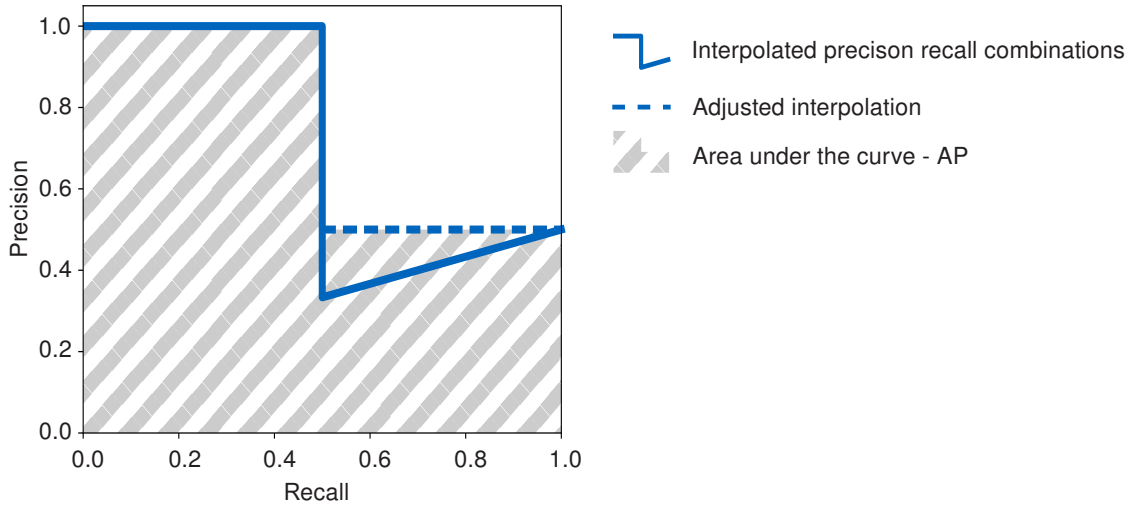


Figure 2.2: The precision-recall curve for four predictions where the most and least certain predictions are correct and the two remaining predictions are incorrect. The AP in this example is 0.75 which is the adjusted area under the precision recall curve as indicated by the dotted line. Selecting a classification threshold leading to precision recall pair below the dotted line is not reasonable for any use-case so that such thresholds are excluded in the AP calculation.

the IoU threshold is still commonly used. A comparison of IoU- and distance-based matching is given in Figure 2.3.

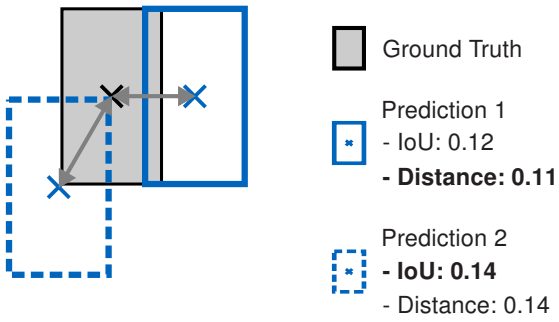


Figure 2.3: Comparison of IoU and distance metric for bounding boxes matching. Prediction 1 is the preferred match when using the distance metric. Prediction 2 is the preferred match when using the IoU metric. Inspired by Kim et al. [84].

2.3 Object Detection

Object detection algorithms can be categorized by the intended input sensor modality and the algorithmic principles used. Notable applications of moving object detection in practice were demonstrated during the Defense Advanced Research Projects Agency (DARPA) Urban Challenge in 2007 [41, 85]. Objects were detected from 2D lidar input and compared to a static map. Detected differences to the static map were classified as moving objects and tracked with a Kalman filter [41].

For more complex real world urban traffic scenarios, more features have to be taken into account to accurately identify clusters of points in the 3D space or pixels on images as objects. The performance of straightforward rule-based approaches for these highly diverse cases has been surpassed by deep learning-based methods [46, 86, 87] which learn a complex set of decision boundaries directly from diverse real world traffic data.

Advances of deep learning methods for object detection have been made possible due to increased computational capabilities and available data sets that can be used for supervised deep learning. Deep learning methods are composed of a multitude of stacked neural network layers whose weights are jointly optimized. The optimization takes into account vast amounts of input training data in an iterative fashion to generate discriminant features to solve the desired task. The basics of deep learning methods are assumed to be known for the remainder of the thesis. The following sections only review object detection specific research advancements. Section 2.3.1 first reviews the camera-based object detection methods as modern deep learning methods were first developed in this field. More advanced 3D detection methods for lidar sensors are shown in Section 2.3.2. Radar data are only recently processed with deep learning techniques so that Section 2.3.3 gives a more in-depth overview of rule-based methods as well as deep learning-based methods. Section 2.3.4 presents methods to fuse sensor data from different modalities for object detection.

2.3.1 Camera-based

Cameras are widely used passive sensors in automotive applications. Their high resolution and color perception makes them suited for tasks such as road marking [88], traffic sign [26], and human intention detection [89].

A single camera cannot measure the distance of the recorded pixels; thereby it cannot determine the 3D position of the underlying objects. This is a drawback for object detection for autonomous driving where a precise estimation of the 3D pose of surrounding objects is needed to maneuver the environment.

Deep learning methods are proposed to estimate the depth from camera images, but the performance of these methods is currently not sufficient for autonomous driving tasks [90]. Stereo camera systems calculate depth information from two cameras recording the same scene from different view points. However, stereo pixel matching and depth calculation increase complexity and processing time [91], while vibrations negatively impact the performance obtained by such systems [92]. The remainder of this thesis therefore focuses on monocular camera data.

Rule-based Processing

Object detection on image data is a subcategory of Computer Vision (CV). Rule-based CV image processing uses user-defined filters to generate characteristic features from images [93, 94]. For example, edges of objects in images can be detected by applying a sequence of user-defined filters over the image pixels [95]. A filter detecting horizontal edges can be applied as follows:

$$\underbrace{\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{input image}} * \underbrace{\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}}_{\text{horizontal edge filter}} = \underbrace{\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 4 & 4 & 4 \\ 4 & 4 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{\text{output edges}}.$$

In the example, the $*$ operator denotes a padded convolution which is used to generate an output image with the same dimensions as the input image. The example input image is white at the top and black at the bottom. The strong horizontal edge between the two different areas in the example image is correctly detected in middle of the output edges image by the filter.

Characteristic features such as edges can further be processed to solve an underlying task. This could be finding the number of steps of a stair case by counting the number of vertical edges in an image. For more complex tasks, more complex sequences of filters can be applied to image data, e.g. to generate depth information for a whole image by using some initial depth measurements [96]. The features can be processed further with machine learning techniques to perform image classification or object detection [94]. The usage of filters from classical CV and the combination of CV and machine learning techniques have accumulated in modern deep learning which uses optimization techniques to automate the filter definition and resulting algorithm generation.

Deep Learning Methods

Effective filters for the underlying task are automatically generated in the optimization of DNNs, replacing user-defined filters and handcrafted features. In practice, the optimized or learned filter weights have proven more effective than human conceived filters from rule-based CV for image classification and object detection. The use of learnable convolutional filters in DNNs increased the performance of image classification algorithms significantly [97]. The strong performance of convolutional filters, which consider the neighborhood pixels for every image pixel in the classification process, have led to the explicit naming of this sub group of DNNs as Convolutional Neural Networks (CNNs). The success of DNNs can be seen on the ImageNet leader board where such methods claim the first rank since 2012 [98]. Another type of popular network architectures are recurrent neural networks and especially Long Short-Term Memory (LSTM) networks [99] which model the time dimension in the neural network. LSTM cells can be employed standalone or as part of any variant of DNNs.

More recent, notable works develop more effective deeper and more efficient network architectures [100, 101] as well as novel training strategies [102] to further increase the performance of image classification DNNs. The same network principles developed for image classification are used and augmented for image object detection and adapted for further sensor data input. In the following, an overview of 2D and 3D object detection methods is given. A visual comparison of the different processing pipelines is given in Figure 2.4.

2D object detection on image data makes use of deep learning network architectures to locate and classify objects in different parts of the image simultaneously. Girschick et al. [103–105] propose a series of detection architectures which first identify regions of possible objects on images and in a second step precisely localize and classify objects in these regions. Other notable works approach object detection as a one step task, leading to shorter execution times for real-time application while reaching lower AP scores than two step models [106–108]. Variants of such models perform additional pixel-wise semantic segmentation of the image data, to receive pixel-level estimates for surrounding objects [109, 110].

Further areas of research focus on improving the detection score in comparison to respective baseline network architectures for specialized use-cases. Lin et al. [111] develop a widely-adapted feature pyramid network. The network proposes object detections on different scales which facilitates the joint detection of spatially big and small objects from the same input data. In

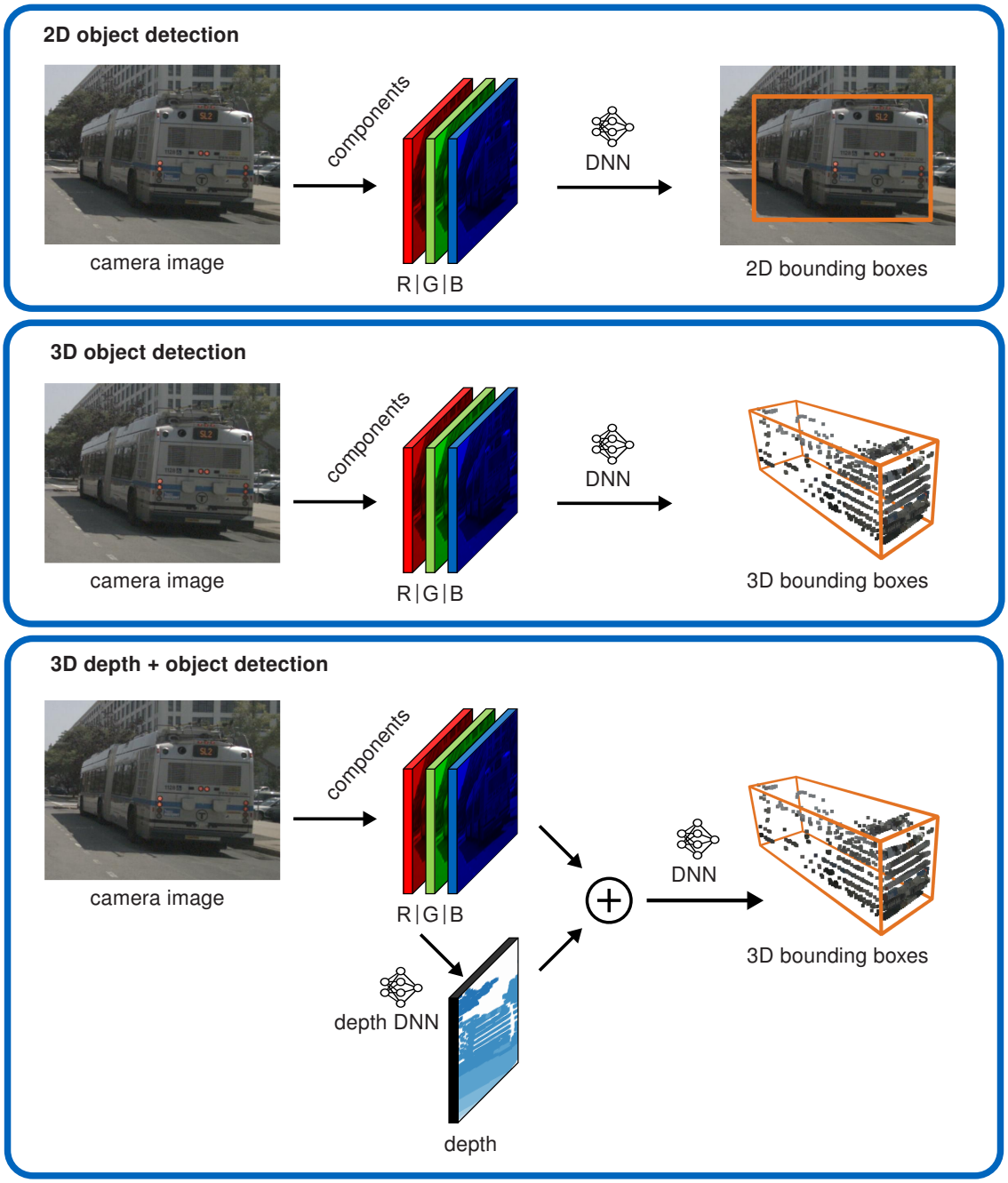


Figure 2.4: Comparison of camera-based object detection pipelines. The same Red, Green, Blue (RGB) input channels are processed to generate different outputs: bounding boxes in the 2D or 3D space and pixel-wise depth information.

autonomous driving, this can be helpful to detect close and far vehicles or pedestrians alongside with trucks from the same input representation.

Further important use-cases include class imbalances in recorded data. Most of the input image space consists of irrelevant background objects, while some classes such as emergency vehicles might be underrepresented in the data. *Focal loss* [112] steers the optimization process to consider underrepresented and wrongly classified examples more strongly. The stronger weighting of harder samples implicitly favors the learning of underrepresented classes. As the proposed approach only modifies the loss function of prior proposed 2D object detection DNNs, it is adaptable to a variety of models and domains, e.g. object detection on lidar and radar data.

Even though 2D object detection is less complex than 3D object detection, the used CNNs require a significant amount of computational resources. Much research is dedicated to derive efficient architectures from the current state of the art. Huang et al. [113] compare a variety of methods in terms of their quantitative performance and execution time. Other works focus on finding efficient network architectures by eliminating layers and nodes that only contribute little to the overall detection result [114].

The works of the different research directions in the field of deep learning mentioned above can be combined. For example, feature pyramid networks can be trained with *Focal loss* or individual network layers can be replaced with more efficient network layers. This overview of 2D object detection methods serves as an overview of important concepts that are re-used for more complex developments. Methods that serve as a direct inspiration for this thesis will later be summarized in Table 2.3 for all used modalities.

An exact quantitative comparison of literature methods is of secondary importance for this thesis. As the methods have proven beneficially in their context, they could prove of greater importance for further developments than the current quantitative state of the art model. Due to the rapid development in this field, the state of the art model changes frequently and the current leading model might not be the most important reference for future developments which set a new quantitative 2D mAP state of the art score. For completeness of the general review, the highest 2D mAP score on the challenging Common Objects in Context (COCO) data set [115] is at 60.6 % [116] as of August 2021. On the KITTI data set [39], the highest monocular 2D mAP score is at 95.2 % [117].

3D object detection from image data is an inherently harder task than 2D object detection as the depth as the third dimension is not explicitly present in the input image data. DNN architectures have been adapted for this challenge.

A first group of 3D object detection networks operates on the 2D data alone. It makes use of keypoints and intermediate targets [118, 119], and geometric constraints between 2D and 3D bounding boxes [120] to enhance the 3D detection performance. Specialized loss functions, training strategies [121] and proposal generations considering the geometry of the environment [122] shall further facilitate the training.

A second group of approaches first estimates depth information from every pixel with CNNs [90, 123] and then uses the Red, Green, Blue, Depth (RGB-D) data to detect objects in 3D [124, 125]. This learned depth information can also be used to project the image pixels back to the 3D space to operate on pseudo-lidar point cloud data [126–128]. Methods originally developed for lidar input data can then be applied on image data [117]. To further improve the performance, new works in monocular 3D object detection also consider lens effects in the learning process [129].

However, the achieved 3D mAP of even the highest ranking methods is around six times lower than that of lidar-based methods on public data sets [39, 130, 131]. That holds true despite monocular images providing higher resolution input data [132] as monocular depth estimation is less accurate than the estimation from stereo cameras [133] and the measurement from lidar sensors.

The current quantitative state of the art of monocular 3D object detection achieves a 3D mAP score of 13.9 % [130], using a combination of depth estimation, key points, and geometric constraints. The state of the art in stereo 3D object detection reaches 54.2 % 3D mAP [134]. The state of the art lidar method tops the KITTI ranking at 82.5 % 3D mAP [131].

2.3.2 Lidar-based

In contrast to cameras, lidars are active sensors that calculate the distance to objects by measuring the time interval between the send and receive event of the light waves they emit [135]. Different sensor hardware designs are used in the automotive context. Mechanically spinning lidars scan the environment consecutively while flash lidars use optical lenses to transmit and receive the sensor signal for all directions simultaneously at discrete intervals [136]. This thesis forgoes to discuss lidar sensor hardware specifications and challenges and refers to further literature [136–139]. Irrespective of the sensor hardware principles, lidar sensors output data in a point cloud format. Each point therein comprises its positions and respective reflectance value. In contrast to camera sensors, the data format is less structured, as no regular grid is used for the data representation. This leads to a variance in the number of data points, for example depending on whether the lidar is scanning an empty rural road or a busy intersection with many objects and static scenery around it.

Rule-based Processing

Rule-based lidar processing identifies characteristic structures in the point cloud data to detect objects. Clustering methods can be applied to lidar data to find physically close points in the point cloud and to identify the objects' class from the shape of these point clusters [140]. Alternatively, the internal geometry of the laser scan pattern can be used to cluster points [141, 142]. A third principle for object detection is to project the sparse point cloud data to a discrete (occupancy) grid [143–145]. Neighboring grid cells of similar properties, e.g. returning a measurement point, are identified and clustered to extract objects [146, 147].

The overall performance of rule-based methods cannot compete with the one of deep learning-based methods; this can be inferred from the comparison of methods on public data sets where deep learning-based methods have constantly achieved the highest scores in recent years [39, 40]. Following this short overview of rule-base algorithms, the next section therefore focuses on these state of the art methods.

Deep Learning Methods

Similar to image processing, DNNs are dominating object detection benchmarks for lidar data. PointNet [148], PointNet++ [149] and Kernel Point Convolution (KPConv) [150] are notable DNN architectures that directly process sparse point cloud input data. KPConv transfers the concept of convolutions of data points with their neighbors from 2D image grids to irregular 3D point cloud data.

Standard CNNs for image data cannot reasonably be applied to 3D lidar data due to the increased computational complexity caused by the additional spatial dimensionality of the 3D lidar data in comparison to 2D image data. To solve this, efficient implementations of 3D grid convolutions have been developed to process sparse point cloud data from a 3D voxel grid [151, 152]. These efficient implementations are possible as most 3D grid cells remain empty in typical traffic scenes and can be discarded in the sparse processing. Both, direct point processing and sparse convolutional layers are used as the building blocks of state of the art lidar processing. A visual comparison of the different processing pipelines for direct point processing and sparse convolutions is given in Figure 2.5.

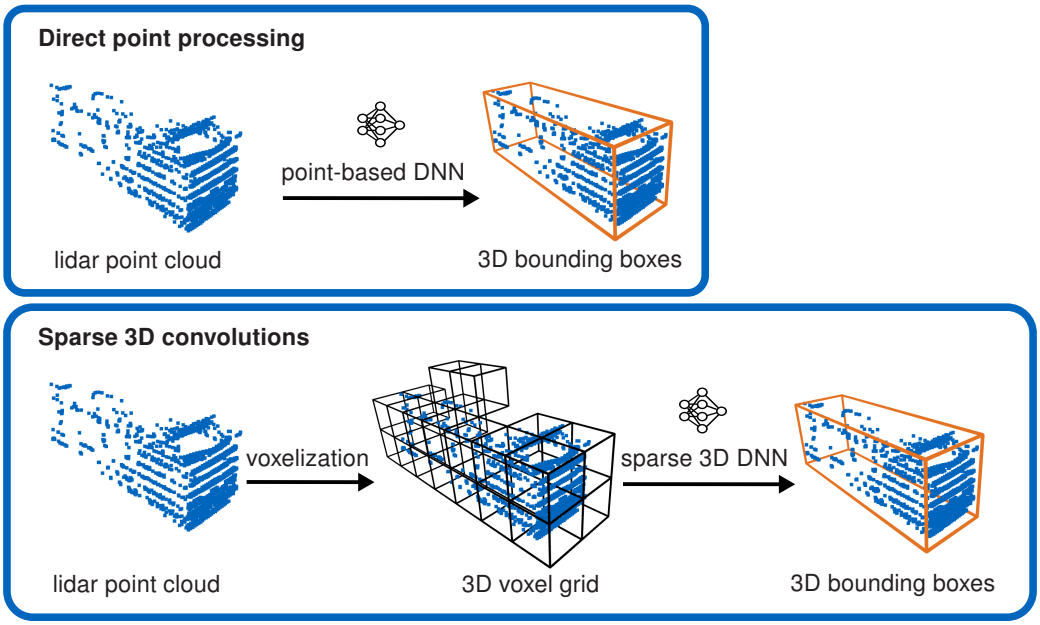


Figure 2.5: Comparison of point-based and grid-based processing pipelines for lidar object detection.

Liang et al. [153] project lidar data to a camera-like 2D front view plane. The resulting 2D image is processed with a 2D CNN to extract features for the input points. The generated features are then projected to a perpendicular Bird's-Eye View (BEV) plane to perform the 3D object detection. In this way, no convolutions in the 3D space are needed to obtain the final detection result.

VoxelNet [154] and Sparsely Embedded Convolutional Detection (SECOND) [151] propose using sparse 3D grids to perform object detection. The detection head is similar to the ones used in 2D object detection, making their respective sparse convolution implementations the major novelty of the works.

PointPillars [155] encodes point-based features into a 2D grid which incorporates all points and features along the height axis in a single combined cell, leading to a BEV representation. This is justified by the fact that the height of traffic objects does not vary much and is of less importance to the driving task. After encoding features with a point-based structure, the features are processed in a 2D fashion. This way the network makes use of a combination of PointNet and image object detection architectures.

Shi et al. [156] develop a point-based two stage approach adapted from Faster-RCNN [103] for 3D object detection and refine the achieved performance in a continuing work [157]. Shi et al. [158] further use a combination of grid- and point-based processing, creating a rough estimation from the grid representation and a local refinement on point-based data. At the time of writing this

thesis, the best performing method [159] on the nuScenes data set [40] similarly uses a VoxelNet network structure with a second refinement step for the final object detection output.

As human performance in object recognition increases when viewing an object for a longer consecutive time [160], further efforts have been made to use the time dimension during point cloud processing. LSTM networks are adapted to accept point clouds as input data [161]. Additionally, motion information can only be inferred by considering time-dependent information in the perception task [162]. Despite these motivations for the processing of time-dependent information in DNNs, time-dependent methods could not outperform networks processing a singular combined point cloud of the input data as of writing this thesis.

Despite their sparsity, processing point clouds with DNNs remains a calculation intensive task. Han et al. [163] provide a review on filtering point cloud data to make the detection processing in the DNN less computationally expensive. Simple filtering methods may limit the sensor field of view or area of interest to reduce the point cloud size. When applying these methods for real-time applications, one needs to consider the trade-off between the additional processing time for the pre-processing operations versus the time saved through the reduced input size for the DNN.

The current quantitative lidar-based state of the art for object detection on the KITTI data set reaches a 3D mAP of 82.5 % [131]. The highest 3D mAP score on the nuScenes data set is 67.1 % [159]. A direct quantitative comparison of both models is not given as they are only evaluated on different data sets.

2.3.3 Radar-based

The third widely used sensor type for automotive object detection is the radar sensor. Similar to lidars, radars emit electromagnetic waves and calculate the distance to objects by measuring the time it takes until the receiver module records the reflection signal of the transmitted wave. The doppler effect [164, pp. 420-425] is used to directly measure the radial velocity component of objects—hereafter called *range rate*—in one time step. Due to the different wavelengths used, radars are more robust against inclement weather conditions such as heavy rain or fog than cameras or lidars.

Figure 2.6 gives a high-level overview of an exemplary radar signal processing chain adapted from Engels et al. [73]. The raw radar samples are processed in several stages in the pre-processing module. The fast-time Fast Fourier Transform (FFT) is used to measure the distance or range to an object. The slow-time FFT is used to extract the range rate information. The phase shift of the receive signal at different antennas of the radar is used in the beamforming module to measure the azimuth angle to an object. In current production radar sensors, the three dimensions *range rate*, *range*, and *azimuth angle* form the radar cube. Peaks in these signals are detected with the target detection and parameter estimation modules to create point targets which are the elements of a radar point cloud. The received radar signal strength of the point targets, also called Radar Cross-Section (RCS), is strongly dependent on the surface material, shape, and size of the reflecting object. The RCS alone is not a discriminative indicator to classify different point targets [165, 166]. A radar point cloud consists of both moving and non-moving point targets and a notable number of noise targets. The developments of this thesis operate on data in form of a point cloud. Further common names for the point cloud are *target list* or *radar clusters*. Winner [167] provides further details on radar processing.

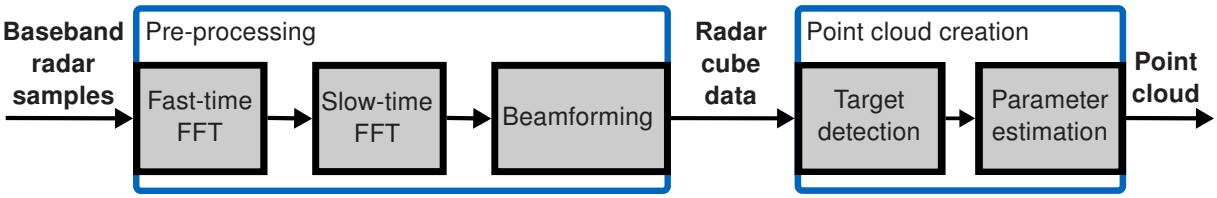


Figure 2.6: Exemplary radar signal processing chain, adapted from Engels et al. [73]. This section reviews object detection methods based on both, radar cube and point cloud data. The developments of the thesis use point cloud input data.

The frequency bandwidth and antenna characteristics limit the resolution of radar sensors [73, p. 1]. Noise and multi-path reflections can lead to unwanted point target detections. Due to the high background noise level and low resolution, non-moving objects of interest, such as cars, are not easily distinguishable from other background surfaces. For this reason radar sensors are mostly used for moving object detection, e.g. for an ACC application where the range rate feature delivers discriminative information to distinguish objects from the non-moving environment. Another disadvantage of current radar sensors is that no height information is measured, making the classification and distinction between different object classes more difficult. Recently, sensor hardware advances enable the measurement of 4D information (3D space + range rate) [73, 168]. Due to the experimental status of such systems, the remainder of the thesis focuses on the processing of mature production radar point cloud data with a maximum of one range rate and two spatial dimensions.

Rule-based Processing

The noise level of radar point cloud data has to be reduced further to provide confident object detections. For this, the radar point clouds are tracked with Bayesian filtering methods such as the Unscented Kalman filter [169] to generate object detections. Due to the high noise level in radar data, a tracking of a point target over time, before deciding on whether or not an object is present at a certain location, is of greater importance than for lidar and camera sensors. Despite their close relation, tracking and data association are their own research fields which enable radar-based object detection. Data association can be performed with simple and computationally cheap algorithms such as the Hungarian method [170]. The distances between past tracked objects and the current measurements are used to minimize the data association costs. More advanced algorithms make use of probability hypotheses to find an optimal assignment in more complex circumstances [171].

In an automotive environment, spatially close objects need to be distinguished and associated over multiple time steps. Pedestrian crowds are an especially challenging scenario for tracking and association applications, as the movement properties of these objects can change quickly, distances of passing individuals are small and their paths cross regularly [172]. Advances in tracking and association are discussed in additional literature [173, 174].

With advances in radar technology, single objects can be represented by a multitude of points originating from the same object of interest. The number of points per object can vary greatly from no point at all to 20 or even more points per object depending on the sensor type, distance to, and the type of the object of interest. Tracking and associating this many points is no longer feasible with the aforementioned methods. Instead, the measurement points are clustered per object and then tracked to reduce the computational complexity. For clustering, several authors adapt the Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [140]

to the usage on radar data. For this, the radar points are converted to a grid and then clustered based on their spatial similarity [175–177]. Kellner et al. [177] incorporate the known changes in Cartesian resolution over the distance into the clustering. Li et al. [176] use the range rate as an additional input dimension to perform clustering on a 3D grid.

Stolz et al. [178] adapt border following approaches from image processing to identify radar clusters in either Range-Azimuth (RA) or Range-Doppler (RD) projections of the point cloud. The algorithm is demonstrated for several objects in a static scene.

Yuxuan et al. [179] learn the geometric distribution of radar measurements from the underlying nuScenes data set [40]. The work argues that radar point target distributions cannot accurately be modeled by contour or surface models and that the data driven approach provides more accurate results than pre-defined distributions for the tracking application. The presented rule-based object detection method is in an experimental stage and tested for detecting a single vehicle.

Rule-based methods are successfully applied for moving object detection on radar data in series production automotive systems. In contrast, general object detection of both, moving and non-moving objects from radar data is still in a research phase. Due to the state-of-the-art results of deep learning methods for object detection on camera and lidar data, the transfer of such methods to the radar domain is recently explored.

Deep Learning Methods

This section provides an overview of different research directions in this emerging field. Radar deep learning research is more actively evolving since around 2016. Lombacher et al. [180] apply a grid-based CNN on radar data to classify grid patches according to the type of object contained in the patch. In a continuing work [181], an ensemble of a CNN and random forest is used to boost the performance of the grid classification. The two step approach of Schumann et al. [182] clusters radar point cloud data with a DBSCAN approach and then classifies those clusters with either a random forest or an LSTM network. The neural network shows favorable results in comparison. These early works originate from the same research group, using similar data in their publications. Since 2019, more radar data became available to the public, facilitating additional research.

A broad overview of machine learning applications for radar data can be found in review papers [73, 183, 184]. The reviews of Engels et al. [73] and Abdu et al. [183] focus on the application to automotive radar data. Only selected works were presented in the review of camera- and lidar-based object detection methods in the previous sections. Due to the significantly lower amount of literature, this section gives an overview of all automotive radar deep learning-based object detection literature known to the author. Many of the works are developed concurrently and independently from each other so that this section mainly provides an extensive structured list of works in the field. Relations between papers are given in case they exist.

Automotive radar deep learning methods can be subdivided based on the used input format. The first group of approaches uses radar cube data in either 2D or 3D grids as an input. The second group of algorithms processes the point cloud data directly with DNNs. Unsurprisingly, the first group of approaches uses algorithms more closely related to image processing while the second group is more heavily influenced by lidar point cloud approaches.

Radar cube data-based methods use the dense radar information before a point cloud representation is extracted from the peaks in the signal. Cheng et al. [185] propose to use an

encoder-decoder CNN to convert the input Range-Doppler image into a grid of probabilities of an object being present at a certain coordinate. The algorithm is evaluated on a non-public data set. The quality of the resulting point cloud is compared to a rule-based extracted point cloud with Constant False Alarm Rate (CFAR) [186]. The deep learning approach generated a point cloud with greater similarities to a lidar point cloud which is used as benchmark data.

The following methods perform both point cloud extraction and/or some form of object detection from either Range-Azimuth (RA) or Range-Azimuth-Doppler (RAD) matrices.

Stroescu et al. [187] manually extract data of objects and their close environment from the Range-Azimuth-plane. Their model detects objects on these extracted data. The work shows favorable results on a non-public data set. However, as only one object is present per data patch, the work performs a task similar to the classification performed in previous work [180] with an additional refinement for the localization rather than a complete object detection.

The network of Major et al. [188] operates on the whole RAD cube tensor. The network processes 2D projections of the cube as input data to generate the object detections. Their work is the first to perform peak extraction and object detection in one combined DNN. The use of the doppler information enhances the detection performance.

Azam et al. [189] operate on Range-Azimuth data. The work uses three different color spaces to represent the input data. All three color space parameterizations of the same data are concurrently processed in the network to enhance the feature extraction performance with an attention mechanism. The paper states the robustness of the approach to inclement weather conditions, as the performance is not affected by different environment conditions present in the used data.

Another work sets its focus on the detection of pedestrians and bicycles from radar data [190]. Lidar and camera data are leveraged to label a non-public data set. The network takes the Range-Azimuth-Doppler (RAD) cube data as input. To limit the data amount and computational expense of the processing, only moving targets in a distance closer than 20 m to the ego vehicle are used as input data. The network is trained with a combination of noisy labels from automatic annotations and precise hand-labeled annotations. The addition of noisy automatically labeled data increased the achieved performance by providing more variance to the training set.

Palffy et al. [191] use a combination of RAD radar cube and point cloud data to classify radar points and cluster them to objects. The approach shows better classification results on the same data as Schumann et al. [182] which operate on point cloud data only.

Point cloud-based methods are first applied by Schumann et al. [192] in 2018. The work adapts a PointNet++ approach for point cloud segmentation as a preliminary result for object detection. The segmentation result outperforms prior work from the same group using a clustering and LSTM network [182]. Ablation studies show that using range rate and RCS measurements as additional features improve the overall result.

Danzer et al. [193] are the first to adapt a point-based approach for both segmentation and object detection on radar data. The network is evaluated on a non-public data set.

Dreher et al. [194] adapt the previous work [193] and compare point-based and grid-based object detection on the public nuScenes data set. The point-based approach outperforms the grid-based approach by a large margin for all evaluated input data. However, the low data density of the used data set leads to overall less favorable scores than works using similar methods on non-public data. Problems with the nuScenes radar data are also pointed out by additional literature [71–73].

A recent work from Schumann et al. [195] provides an extended framework for radar point cloud processing. Non-moving radar points are accumulated in a grid and segmentation is performed with a CNN. Moving points are classified with an augmented point-based approach. The work extends the point-based network with a fixed-size memory point cloud whose features influence the current segmentation step to incorporate the time dimension into the segmentation process.

From a quantitative point of view, only one paper reports radar-based detection results on the publicly available nuScenes data set. The paper achieves a 3D AP for the car class of 10.2 % [196]. In comparison the CenterPoint lidar method achieves an AP of 87.0 % [159].

2.3.4 Data Fusion-based

A single sensor always measures a specific subset of the whole information content that is available. This is true for both, the human sensory system and technical sensor systems. Even more importantly, the data that a sensor provides are only valid to a certain accuracy and resolution subject to certain conditions. Optical illusions are an everyday example that show the limitations of the human perception system. To compensate sensory imperfections, humans combine several sensory modalities, e.g. smell and sight to determine whether food is still edible, or hearing and sight to localize movements in the surrounding. While humans combine their sensory systems intuitively, a lot of research is performed to find suitable fusion methods of several sensor systems in technical contexts.

Back in 1988, Moravec [197] equipped a robot with several sensing systems such as sonar, camera, and contact sensors to create an occupancy grid map of the environment. The sensor characteristics are taken into account in order to calculate the occupancy probabilities for each cell in a grid map built up from multiple sensor measurements. Bayesian principles are used in the sensor fusion via a Kalman filter [169, 198].

However, the relationships between the measurement content and quality of different sensors cannot always be estimated precisely. This motivates research in data-based methods that learn these characteristics from real world data from the target domain. As early as 1990, Chaudhuri and Das [199] fuse the measurements of thermal and radar sensor information in a simple neural network to classify airborne targets.

Recently, advanced deep learning networks are used to fuse data of multiple sources, also called *multi-modal data*. Liu et al. [200] fuse multi-modal data, in the form of different features extracted from social media to estimate the gender of a user. In all three tasks presented in the paper, the multi-modal features lead to an improvement in the achieved performance.

In the autonomous driving context, a review for deep learning fusion methods [201] focuses on localization and mapping tasks. Velasco-Hernandez et al. [202] review autonomous driving software solutions in general. The work focuses on perception tasks and sensor fusion aspects. It highlights the importance of sensor data fusion to enable safe autonomous driving. This includes the need for a robust framework where different software components do not interfere with each other's functionality and resources. A more extensive review by the same authors [203] provides further insights into different sensor technologies, calibration methods, and sensor fusion algorithms for object detection.

Lidar and camera data are widely available to the public. Cui et al. [204] review fusion approaches for both sensor data. A comprehensive review of deep learning sensor fusion methods is provided by Feng et al. [35]. The work reviews both segmentation and object detection algorithms. Additionally, it presents data sets available to the public. The research shows that no single

fusion technique stands out in all circumstances. Furthermore, it stresses that radar data sets and radar processing research is under-represented in recent research. While all presented review works motivate sensor data fusion, no best practice methods have been established so far. In the following, a review of three conceptually different fusion strategies for object detection is presented: Consecutive fusion, probabilistic fusion, and deep learning fusion. A visual comparison of the different fusion strategies is given in Figure 2.7.

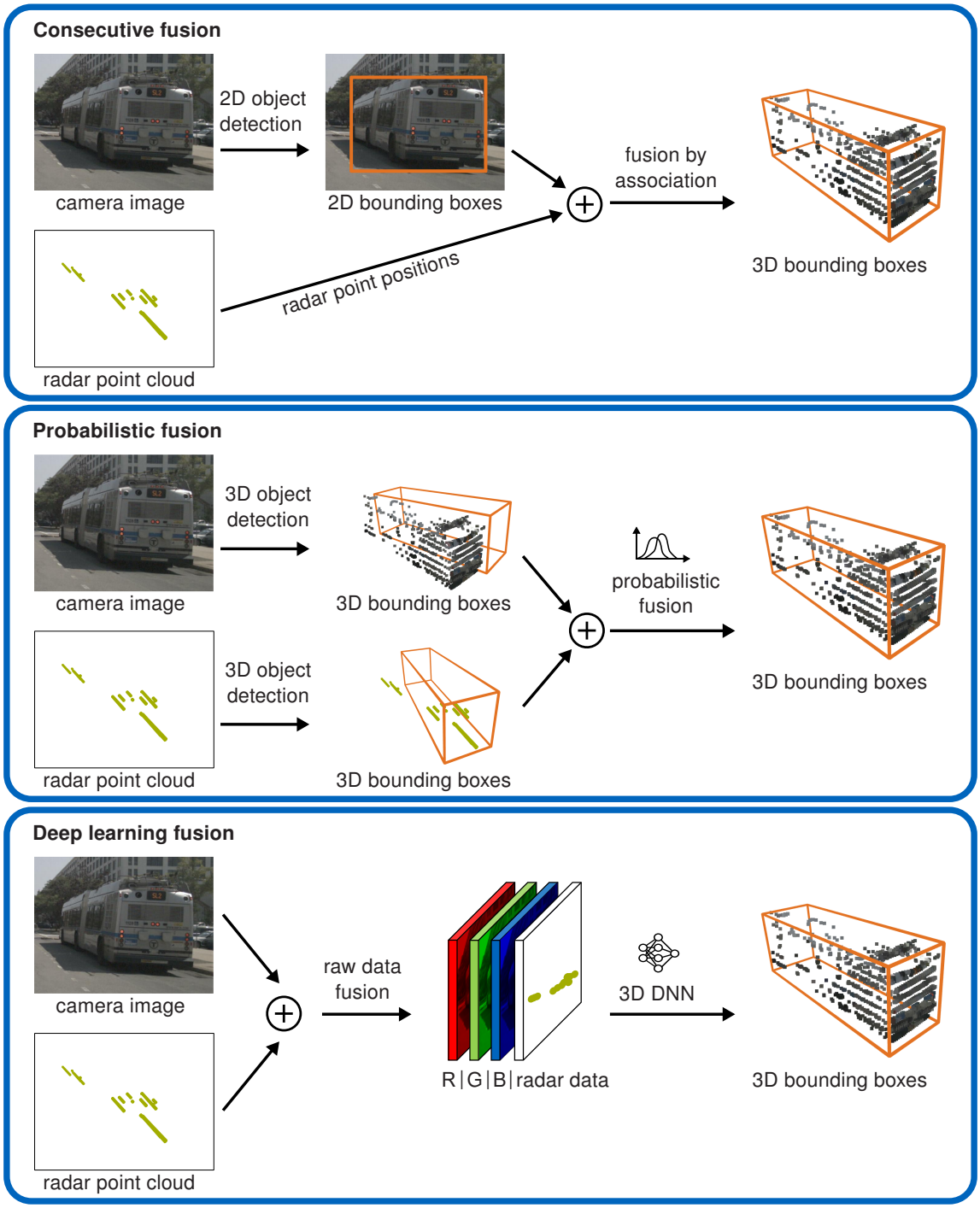


Figure 2.7: Comparison of consecutive, probabilistic, and deep learning fusion pipelines.

Consecutive Fusion

Consecutive fusion methods first process data from one sensor modality and then process additional modalities to generate additional insights. Each sensor is responsible for a fixed task and no redundant information, e.g. independent estimates of an object position from different sensor inputs, are fused.

Han et al. [205] project radar detections to the image space. The surrounding Region of Interest (ROI) of the radar projection is then fed to a machine learning-based image classifier. Nabati and Qi [206] and Zhang et al. [207] use the projection of radar point cloud data as the anchor locations for 2D object detection with a DNN. While these methods leverage the strengths of the sensors, the resulting detection performance is strongly dependent on accurate radar data. Should the radar not detect a vehicle, the image processing does not serve as a backup to compensate a misdetection.

Jha et al. [208] first use a CNN to detect the objects on the image plane. If a radar target is present at the same location, its distance information is associated to the 2D bounding box to complete a 3D object detection. In this case, the camera detection is the main limiting factor for the overall performance. The fusion pipeline of this work is visualized in Figure 2.7. Such methods would not detect objects which are clearly visible in the radar data for cases where the prior camera detection fails. To be able to compensate errors in one processing pipeline, a different processing strategy is proposed.

Probabilistic Fusion

Probabilistic fusion methods create independent object detection outputs from each sensor data, before computing a final object detection output by considering estimated uncertainties of the sensor-wise object detections.

The fusion of different sensor modality input data for object detection of automotive applications in series production is often performed in the form of a Kalman filter fusion [169, 198]. The individual object detections can either be tracked per sensor or fused directly in an asynchronous Kalman filter with adequate sensor inputs.

A performance study comparing different fusion strategies [209] favors individual tracking and a subsequent fusion, also called *track-to-track fusion*. Data fusion on multiple levels is performed by Sun et al. [210] for multiple radar sensor data inputs. Liu et al. [51] use a CNN for image-based object detection and fuse the detections of camera and radar data with a Kalman filter and data association [211]. Recently, more track-to-track fusion algorithms have been applied to automotive applications for camera and radar fusion [212] as well as camera, lidar, and radar fusion [213].

For all these methods, assumptions about the noise-level and measurement accuracy are used for obtaining the final results. Assumptions need to be made during the steps of data association, tracking, and the fusion itself. In practice, it is difficult or even impossible to obtain general and accurate assumptions for all parameters as the detection accuracy of different sensors varies depending on environment conditions and the type of object that is to be detected. To mitigate this problem, data-based solutions are explored that learn the fusion output from an adequate set of examples.

Deep Learning Fusion

Deep learning fusion methods fuse raw data of all sensors in a DNN, providing a shared object detection result. The weighting of different sensor input data is performed by the network itself.

While adequate data are a necessary condition for the development of a successful algorithm, just providing data to a DNN will not lead to the desired result if the network architecture and loss function are not tailored to the specific task. Convolutions that capture spatial relationships are an important structure for image object detection. The structure of 3D scenes can be captured by voxel grids in the network structure [154] or by advanced point-based networks which model the relationships of points in the provided network connections [149]. This structure helps the optimizer to find a minimum that is in alignment to the abstract task, e.g. object detection, that is to be performed. However, it has not been proven whether or not these structures are beneficial for sensor fusion tasks. Finding an optimal network structure and loss function for sensor fusion networks is therefore subject to current research. In contrast to probabilistic fusion, assumptions about the sensor characteristics do not need to be modelled manually. On the downside, the performance of a DNN has to be experimentally explored, requiring vast amounts of computational resources for comparing different network parameterizations. This section reviews existing deep learning-based fusion methods, with an emphasize on radar-centric methods.

Depending on the processing level on which different sensor modalities are fused in the DNN, Feng et al. [35] distinguish between *early*, *middle* and *late fusion* methods. Early fusion methods directly fuse the raw sensor data and process them jointly in a DNN. Late fusion methods use separate networks or algorithms to process each sensor modality and fuse the outputs, for example with a Kalman filter as presented in Section 2.3.4. The advantage of early fusion methods lies in the potential to use all raw input data information jointly and learn from interdependencies of the data [214]. Late fusion approaches on the other hand can use specialized algorithms for processing data of each sensor type. Middle fusion methods, also called *feature fusion* methods, processes the input data separately for parts of the network layers, before fusing them and using the joint features to calculate the shared output. In this thesis, early and feature fusion methods are jointly called *low-level fusion* methods as compared to the high-level late fusion approaches. *High-level fusion* methods are most widely applied in production as of writing this thesis. Due to the possibility to compensate sensor weaknesses and add complementary—and not only redundant—information, more research is recently directed towards early and feature fusion algorithms. Such methods are reviewed in the following paragraphs, distinguishing between methods for 2D and 3D object detection.

2D data fusion methods locate objects in a 2D projection space. Chadwick et al. [215] are the first to perform 2D object detection based on a feature fusion of radar and camera data. Compared to the image baseline, the performance of the network is especially effective for distant objects, which are relatively small in the image data.

RV-Net [216, 217] performs a similar fusion and reports a performance gain for more distant objects as well. This approach is evaluated on the public nuScenes data set. The feature fusion approach provides better results than the baseline image-only and high-level fusion approaches. The successor of this work, SO-Net [218] performs a joint segmentation and object detection with a similar architecture. While individual segmentation and object detection networks produce higher AP scores, the joint network is motivated by computational performance gain.

Yadav et al. [219] use feature level fusion of camera and radar as well. The network uses a Faster-RCNN [103] inspired structure. An implementation using both, 2D grid and radar projection guided anchors, provides the best results in their research.

Wang et al. [220] use radar data to localize objects. The projection of the radar objects is used to generate a ROI on the image data. The network then uses image features and associated micro doppler signatures from the radar data in a feature fusion to classify the objects. Consequently, the work uses a mix of consecutive and deep learning fusion methods.

3D data fusion methods in an automotive context are first presented in the form of a feature fusion network by Chen et al. [221]. The network fuses camera and lidar data, where the lidar data are projected to a BEV and a front view. The features of the different sensor processing branches are fused with a mean operation. A similar method, the Aggregate View Object Detection (AVOD) architecture [222] uses camera data and a BEV projection of the lidar data to perform a feature fusion for object detection. The architecture and novel loss set the quantitative state of the art on the KITTI data set in 2018 [39].

The works of Kim et al. [84] and Meyer et al. [223] adapted the AVOD architecture for radar and camera fusion. Instead of a BEV projection of point cloud data, Kim [84] uses Range-Azimuth tensors from the radar domain as input data. The network is evaluated on a non-public data set, outperforming their radar- or image-only baselines. The paper further argues that the IoU threshold for anchor matching is not suitable for 3D object detection. A distance-based anchor matching is proposed which increases the detection performance.

Lim et al. [224] project camera data to a BEV representation before fusing it with the Range-Azimuth data of the radar. A feature fusion is performed by a concatenation of the sensor individual features. The performance of the camera projection is limited especially for distant parts of the sensor FOV. The fusion approach is therefore evaluated for short ranges where it outperforms the sensor-individual detection approaches.

The following groups of approaches work with radar data on a point cloud level. The methods are ordered according to the additional fusion modalities used. GRIF-Net [225] applies a feature fusion of radar and camera data. The fusion is performed by a Gated Region of Interest Fusion (GRIF) which uses learned weights to associate the sensor data. The approach outperforms their radar baseline on the public nuScenes data set by a small margin. Nabati and Qi [226] propose object candidates in the 3D space generated by a camera data-based network. In a second stage, the object detection is refined with both camera and front view-projected radar features. Only radar points which originate from a distance-based interval around the initial object proposal from camera data processing are considered for the detection output. This compensates the problem of associating distant radar points in a 2D projection with unrelated camera data. The joint detection outperforms several camera-based and camera-radar-fusion-based baselines on the public nuScenes data set.

A fusion of lidar and radar point cloud data is performed by RadarNet [227]. After transforming the point clouds to a regular grid structure and initial processing, a feature fusion by concatenation is performed. In later stages, an additional fusion of the radar range rate data is performed to augment the object detection by a velocity estimation. The radar data input increases the performance in comparison to the lidar baseline by a small margin. The LiRaNet [228] approach performs radar, lidar and map data fusion. The network does not only perform object detection, but also predicts future object positions in an end-to-end fashion outperforming their baseline.

Wang et al. [229] fuse camera, lidar, and radar data to perform object detection. Camera data colors are projected onto the lidar point cloud before using a point-based architecture for the object detection. Additional radar features are used to predict velocity information for the object. The fusion of all three sensor modalities increases the performance in comparison to their baseline on the nuScenes data set slightly.

The highest quantitative 3D mAP score of a fusion method on the nuScenes data set reaches 66.8 % [230] which is just short of the highest lidar score at 67.1 % [159]. The highest ranking radar and camera fusion method reports a 3D mAP score of 32.6 % [226]. For radar and lidar fusion, as well as radar, camera, and lidar fusion, no official scores with related publications are released.

2.3.5 Summary of Methods

The thesis gives an overview of object detection methods using different sensor input data. Though a quantitative comparison of the performance of all methods is not conclusively possible due to the limited data availability, the dominance of lidar-based approaches in the upper ranks of public object detection data sets is recognized [39, 40]. This shows that lidar sensor data are an integral building block to achieve quantitative state of the art object detection results. Fusion methods of different sensor combinations gain more popularity due to the potential of complimentary processing and for increased robustness. In the KITTI and nuScenes rankings however, the absolute quantitative performance of published fusion methods, could not surpass that of lidar data-based methods as of August 2021.

The multitude of works mentioned in this chapter contribute in different ways to current research ideas. To condense the shown related research work, Table 2.3 gives an overview of the most relevant publications which inspired the following development of this thesis in Chapter 4.

Table 2.3: Overview of related work most relevant to this thesis. ✓: Official implementation is available open source. (✓): Unofficial implementation is available open source. Publications in **bold** are developed as part of this thesis in Chapter 4.

	Publication	Impact	Open source	Year
camera	VGG [100]	CNN for image-based feature extraction	✓	2015
	SSD [108]	Efficient CNN for object detection	✓	2016
	Feature Pyramid [111]	Detection of objects of different sizes	✓	2016
	RetinaNet [112]	Loss function to handle imbalanced data	✓	2017
lidar	PointNet [148]	Semantic segmentation from point cloud data	✓	2017
	VoxelNet [154]	Efficient point cloud processing in discretized 3D grids	(✓)	2018
	KPConv [150]	Neighborhood convolutions for point clouds	✓	2019
radar	Radar PointNet++ [192]	Semantic segmentation from radar point clouds	✗	2018
	Radar PointNet [193]	Object detection from radar point clouds	✗	2019
	Radar KPConv [72]	Neighborhood convolutions for radar point clouds	✓	2021
2D fusion	MV3D [221]	CNN fusion for camera and projected lidar data	(✓)	2017
	Fusion ResNet [215]	CNN fusion for camera and projected radar data	✗	2019
	CRF-Net [231]	CNN fusion for camera and projected radar data	✓	2019
3D fusion	RadarNet [227]	CNN fusion for radar and lidar data	✗	2020
	RVF-Net [232]	CNN fusion for radar, camera, and lidar data	✓	2021

3 Problem Statement

This chapter draws conclusions from the literature research and identifies open research questions in the field of object detection in Section 3.1. Section 3.2 gives an overview of the deduced research questions and the resulting structure of this thesis.

3.1 Conclusions from Related Work

Chapter 2 discussed recent developments in the field of object detection for autonomous driving. For camera- and lidar-based object detection, deep learning methods have already proven as the most reliable and effective group of algorithms. However, even the highest ranking published lidar method on the nuScenes data set only reaches a 3D mAP score of 67.1 % [159] which shows that current object detection methods are not robust enough to handle all scenarios of interest. Radar object detection and multi-modal radar-centric fusion is still in an early stage of research and development. As shown in Section 2.3.3, radar sensors possess advantageous characteristics for enabling more robust object detection. The low data availability and historically lower spatial resolution limited the development of deep learning approaches in the past years. Radar-based object detection with deep learning methods is identified as the first knowledge gap in the state of the art. In rule-based processing, tracking objects over time is imperative to produce reliable object detection results. Using the time dimension in DNN processing is deemed a possible approach for radar object detection.

The radar data density and labelling availability of the nuScenes data set are the best compromise of publicly available data sets for object detection. The nuScenes data set is selected as the data set used for the evaluation in this thesis. As the data density of radar data remains low in comparison to other modalities, it is questionable whether the data provided by current radar technology are discriminant enough to perform object detection on this sole input source.

The sensor modality specific drawbacks on the one hand and the complementary potential on the other hand motivate further research on sensor fusion for object detection. Fusion methods applied in series production applications perform a high-level fusion of sensor individual object detections. However, to make use of the strengths and compensate the weaknesses of the sensors, a shared processing is needed. This need is reflected in the recent surge of low-level fusion algorithms in literature. Though the motivation from a practical point of view is strong, none of the proposed fusion methods reaches the recent quantitative state of the art performance of object detection with lidar data. Radar data show great potential for practical applications due to their weather robustness, range rate measurements, and durability advantages compared to lidar sensors. Even for lidar-centric object detection, no best-practice low-level fusion method is identifiable from the literature.

The methods presented in the related work for radar-based object detection and radar-centric deep learning fusion for object detection were only developed concurrently to this thesis. This is exemplary shown by the *Year* column in Table 2.3. At the same time, radar-centric object detection algorithms have been open sourced less often than camera and lidar algorithms in the past which set higher boundaries for further research. Radar-centric sensor fusion with additional sensor modalities for object detection is identified as the second knowledge gap.

While there are certain similarities in the presented radar-centric works, different directions of development show that the research field is still presenting a variety of unsolved problems. Exploring of different directions is performed until a dominant processing approach can be deduced from literature. Even though Engels et al. [73, p. 4] state that point cloud data are an efficient basis for data fusion, with the current lack of adequate data, it is not possible to confidently determine an ideal input data abstraction level. Point cloud data are selected in this thesis due to the motivation from related work, the public data availability, and resulting comparability potential to future methods. Due to the little research in this field and the shown potential, this thesis develops algorithms with the goal to set a baseline performance of radar-centric object detection approaches. The performance of the models is evaluated on the public nuScenes data set.

This thesis explores new architectures motivated by practical considerations inspired from classical radar processing and physical relationships of point targets. Both early and feature fusion approaches need to be further explored for radar processing. The usage of an early fusion is motivated by the potential to leverage all raw data jointly. Feature fusion methods are motivated by the fact that different sensor data have different abstraction levels. The exploration of low-level fusion methods in general is motivated by several works [214, 233]. To start off with a strong but not overly complex baseline model, this thesis adapts established deep learning models from camera and lidar processing such as CNNs, KPConv, and VoxelNet to the radar domain.

3.2 Research Questions

As the literature does not provide a conclusive approach for deep learning-based radar-centric object detection, the goal of this thesis is to explore different approaches and provide evidence-based directions for future research guided by the main research question:

Can the usage of radar data make camera and lidar object detection more robust?
Object detection is chosen as the focus of the research question since established methods in the radar processing field are only able to detect moving objects in a limited area of interest, e.g. for ACC applications, in comparison to the need of detecting a plethora of objects in a large FOV for the use-case of autonomous driving. Robustness refers to the application of object detection methods in challenging environment conditions such as rain or night conditions, where the resulting lidar or camera data quality is limited. The research question is further subdivided into sub research questions, which are deduced from the knowledge gap in the state of the art:

- **How to process radar data with deep learning methods?**
- **Should radar and camera data be fused with low-level or high-level fusion methods?**
- **Should radar, camera, and lidar data be fused with low-level or high-level fusion methods?**

Sections 4.1–4.3 of this thesis propose methods to answer these questions in the form of published papers:

- Section 4.1 presents a direct point-based deep learning method to classify radar points. A straightforward approach is researched in order to obtain a first benchmark for transferring existing lidar-based methods into a new domain. The suitability of different deep learning architectures for the application to radar sensor data is explored. Furthermore, the time-dependent processing of consecutive input data samples with deep learning methods is investigated due to the importance of tracking point targets in classical radar processing.
- Section 4.2 fuses radar data and camera data with a combined early and feature fusion deep learning network. The approach is motivated by the maturity of camera-based object detection methods in literature. The fusion approach is explored to obtain a quantitative benchmark in this new field on a public data set. As no prior fusion results are available, the performance is benchmarked against a strong camera-only baseline network. To measure the robustness of the fusion method, additional experiments in inclement weather conditions are performed and discussed in Chapter 5.
- Building up on the knowledge of the prior studies about the importance of lidar input data for object detection, Section 4.3 fuses radar, camera, and lidar data with an early fusion deep learning network. Motivated by lidar-based literature, the fusion of the different sensor types is explored in the 3D space. Different fusion modality configurations are studied in inclement weather conditions to examine the robustness of the sensor fusion approach.

Figure 3.1 gives an overview of the structure of this work. Algorithm development and evaluation is performed in Chapter 4. The thesis discusses the answers to the research questions and the practical implications of the results in Chapter 5. Directions for future work from open questions of this thesis are suggested in Section 5.3.

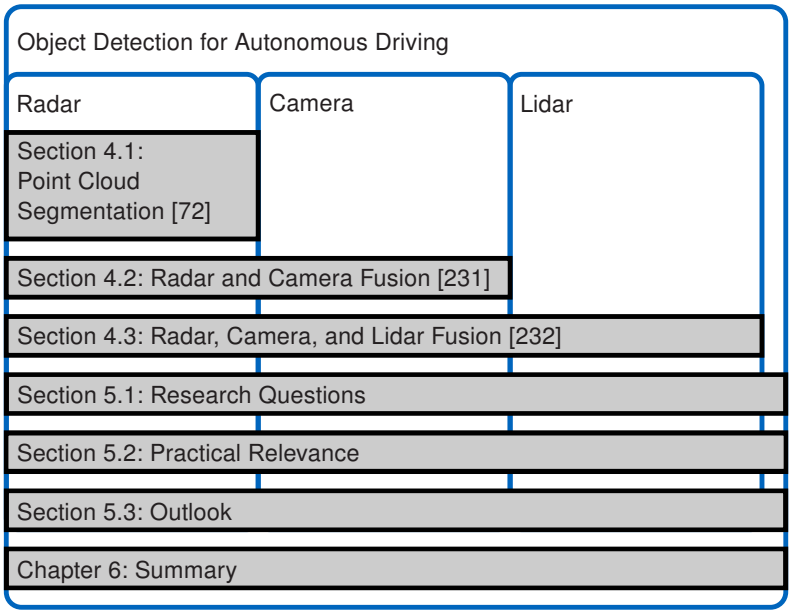


Figure 3.1: Structure of the remainder of the thesis.

4 Algorithm Development

This chapter develops radar-centric deep learning methods to answer the research questions formulated in the previous chapter.

4.1 Radar Point Cloud Segmentation

This work explores the application of deep learning methods on radar data. The content of this section has been published in [72]. The developed semantic segmentation DNN for radar data is available on GitHub [234].

Summary
A point cloud segmentation method for a combined moving and non-moving target classification with radar input data is presented in this paper. The radar data of five calibrated radars are combined for this task. This combination of unimodal sensors is not labeled as a fusion as the thesis reserves the terms of low-level and high-level fusion for data originating from sources of different modalities. A direct point processing approach is selected as the processing method. To model the relationships between radar points more directly than with previous PointNet++-based methods used on radar data [192], a KPConv [150] architecture is adapted to the radar domain. This network architecture leads to more accurate segmentation results on lidar data by modelling spatial relationships more directly [150]. This work is the first to use these neighborhood point-based convolutions on radar data. To decrease the complexity of the learning process, a semantic segmentation is preferred against an object detection head to explore the general feasibility of the use of deep learning methods in a first step.

The segmentation of moving vehicles leads to an F1-score of 75.8 % with the PointNet++ baseline. The proposed KPConv architecture reached an F1-score of 74.7 %. Additional efforts were made to adapt LSTM networks to the irregular point processing domain to mimic the strengths of tracking methods for radar processing with deep learning, called Kernel Point LSTM (KPLSTM). The modelling with point cloud-adapted LSTM layers leads to a score of 75.3 %. All further tested models of the paper have a relative performance difference to the respective baseline lower than 2 %, leading to the conclusion that the more complex network architecture could not significantly improve the performance of the baseline. For the remainder of the thesis, relative differences in the performance scores are given without decimal places while absolute scores are presented with one decimal place to prevent confusion.

The spatial coordinates, radar cross section, and range rate features of the radar data are used as input features to the network. The F1-score for segmenting both moving and non-moving vehicles is relatively 21 % lower than that for semantic segmentation of moving vehicles. An ablation study without the range rate feature in the input space, showed a relative decrease in the segmentation F1-score by 31 % for semantic segmentation of moving vehicles with the

proposed KPConv architecture. Similar performance differences are seen for all tested network parameterizations. The evaluation of the networks shows that the range rate feature is of great importance to the segmentation result which was also deduced from literature of rule-based approaches. The lower segmentation performance achieved without the range rate feature hints that the used radar data are not discriminant enough to detect non-moving objects reliably. A summary of the results is given in Table 4.1.

Table 4.1: Radar point cloud segmentation results. The developed KPLSTM and KPConv methods do not outperform the PointNet++ baseline. Ablation studies are found at the bottom of the table.

Network	Radar Input	Vehicles of Interest	F1-Score	Relative Difference
PointNet++	with velocity input	moving	75.8%	0 %
KPLSTM	with velocity input	moving	75.3%	−1 %
KPConv	with velocity input	moving	74.7%	−1 %
KPConv	with velocity input	moving and non-moving	59.9%	21 %
KPConv	without velocity input	moving	52.2%	31 %

Due to the relatively low performance of the baseline network on the public data in comparison to the performance of the same network on non-public data, the data themselves are investigated in more detail. The paper identifies several weaknesses in the radar data of nuScenes; these include the general sparsity of the radar data as well as issues in the ground truth labels for radar processing. The ground truth data are obtained from lidar data which leads to missing bounding boxes for objects visible in the radar data but not in the lidar data and vice versa. A visual examination of random scenes has shown this pattern of missing labels regularly, as objects where not detected by the used lidar due to the limited vertical resolution while the radar generates point targets from these objects. This is visualized in an exemplary sample of the data set in Figure 4.1.

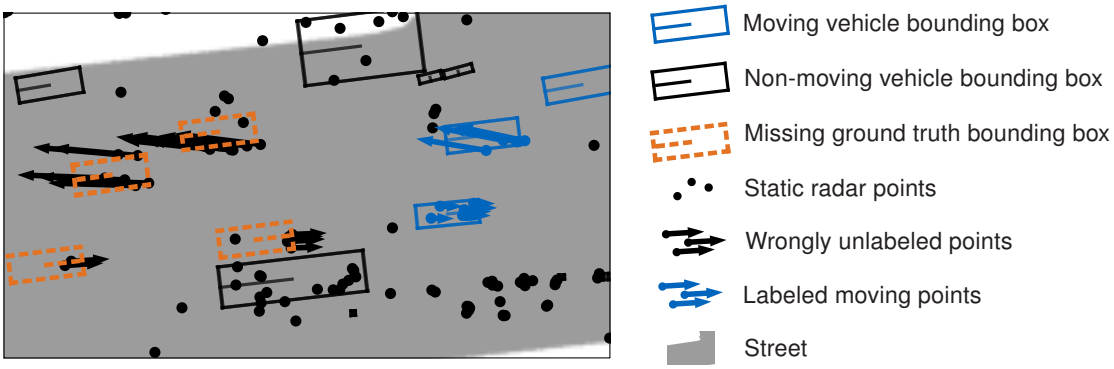


Figure 4.1: BEV of radar point cloud data of the nuScenes data set. Three moving vehicles are labeled by ground-truth boxes. Four moving vehicles are not labeled, even though they are clearly visible in the radar data indicated by their relative range rate vector. Adapted from [72].

These drawbacks make an accurate semantic segmentation on the whole data set with radar data as the sole sensor input unfeasible. The paper therefore foregoes to present an object detection architecture. The results of more recent works [194, 196] confirm that object detection on this limited quality radar data, while theoretically feasible, is not able to produce reasonably accurate results. Instead, the paper finishes by identifying important aspects for the creation of a radar focused object detection data set to enable future research.

Research Questions

This publication is related to the first sub research question. The paper develops a KPConv DNN for radar point cloud segmentation. The limited radar data quality did not enable the development

of a radar-based object detection method. The achieved semantic segmentation performance is on par with that of a state of the art DNN. A PointNet++ network is determined as a fitting DNN for processing the limited quality radar data. The answer to the research question will be discussed in more detail in Section 5.1.

Contributions

Felix Nobis (F.N.), as the first author, initiated the idea of this paper and contributed essentially to its concept and content. Conceptualization, F.N. and Felix Fent (F.F.); methodology, F.N. and F.F.; software, F.F. and F.N.; data curation, F.F. and F.N.; writing—original draft preparation, F.N.; writing—review and editing, F.F., Johannes Betz (J.B.) and Markus Lienkamp (M.L.); visualization, F.N. and F.F.; project administration, J.B. and M.L. All authors have read and agreed to the published version of the manuscript.

Imprint of the Paper

applied sciences

MDPI

Article

Kernel Point Convolution LSTM Networks for Radar Point Cloud Segmentation

Felix Nobis [1,*], Felix Fent [1], Johannes Betz [2] and Markus Lienkamp [1]

[1] Institute of Automotive Technology, Technical University of Munich, 85748 Garching, Germany; felix.fent@gmx.de (F.F.); lienkamp@ftm.mw.tum.de (M.L.)
[2] mLab:Real-Time and Embedded Systems Lab, University of Pennsylvania, Philadelphia, PA 19104-6243, USA; joebetz@seas.upenn.edu
* Correspondence: nobis@ftm.mw.tum.de

Abstract: State-of-the-art 3D object detection for autonomous driving is achieved by processing lidar sensor data with deep-learning methods. However, the detection quality of the state of the art is still far from enabling safe driving in all conditions. Additional sensor modalities need to be used to increase the confidence and robustness of the overall detection result. Researchers have recently explored radar data as an additional input source for universal 3D object detection. This paper proposes artificial neural network architectures to segment sparse radar point cloud data. Segmentation is an intermediate step towards radar object detection as a complementary concept to lidar object detection. Conceptually, we adapt Kernel Point Convolution (KPConv) layers for radar data. Additionally, we introduce a long short-term memory (LSTM) variant based on KPConv layers to make use of the information content in the time dimension of radar data. This is motivated by classical radar processing, where tracking of features over time is imperative to generate confident object proposals. We benchmark several variants of the network on the public nuScenes data set against a state-of-the-art pointnet-based approach. The performance of the networks is limited by the quality of the publicly available data. The radar data and radar-label quality is of great importance to the training and evaluation of machine learning models. Therefore, the advantages and disadvantages of the available data set, regarding its radar data, are discussed in detail. The need for a radar-focused data set for object detection is expressed. We assume that higher segmentation scores should be achievable with better-quality data for all models compared, and differences between the models should manifest more clearly. To facilitate research with additional radar data, the modular code for this research will be made available to the public.

Keywords: perception; deep learning; radar segmentation; radar point cloud; object detection

Citation: Nobis, F.; Fent, F.; Betz, J.; Lienkamp, M. Kernel Point Convolution LSTM Networks for Radar Point Cloud Segmentation. *Appl. Sci.* **2021**, *11*, 2599. https://doi.org/10.3390/app11062599

Academic Editor: Oscar Reinoso García

Received: 11 February 2021
Accepted: 11 March 2021
Published: 15 March 2021

1. Introduction

In everyday driving, humans perceive and react to a variety of scenarios and environment conditions. For a self-driving car to deal with such diversified situations, an extensive model of the environment is needed. For this, a variety of sensors are used to gather information about different aspects of the scene, e.g., road layout, conditions, traffic participants and traffic lights. Currently, object detection of traffic participants in 3D is most accurately performed with lidar sensor data input [1,2]. Notwithstanding, even the most accurate methods only achieve about 67% mAP (mean Average Precision) on the nuScenes data set [2] and 83% mAP on the KITTI 3D data set [1]. To gain a better understanding of the environment, object detection research is therefore pursuing two complementary approaches: Improving the detection accuracy of lidar-based algorithms; or leveraging additional sensor modalities and fusing the detection results.

This paper develops a segmentation model for radar point cloud data to complement the lidar processing. In comparison to lidar sensors, radar sensors have several advantages and disadvantages for performing object detection:

Appl. Sci. **2021**, *11*, 2599

- Radar signals are affected significantly less by rain or fog than lidar [3].
- Radar sensors measure the radial velocity of surrounding objects directly.
- The spatial resolution of production radar sensors is lower than that of lidar [3].
- Current production radar sensors do not measure the elevation component of the radar returns.
- Several processing steps are necessary to obtain a radar point cloud from the raw sensor signals. These processing steps are based on additional assumptions, e.g., prioritizing moving objects, which may not lead to the desired point cloud signal in all environmental conditions.

Due to these characteristics, especially the lower resolution and the radar internal processing focus on moving objects, general object detection based on radar data is a challenging task. Figure 1 shows an example of a scene containing both radar and lidar returns. Estimating the position and the class of the shown object just from the radar returns seems more difficult than using the denser lidar data.

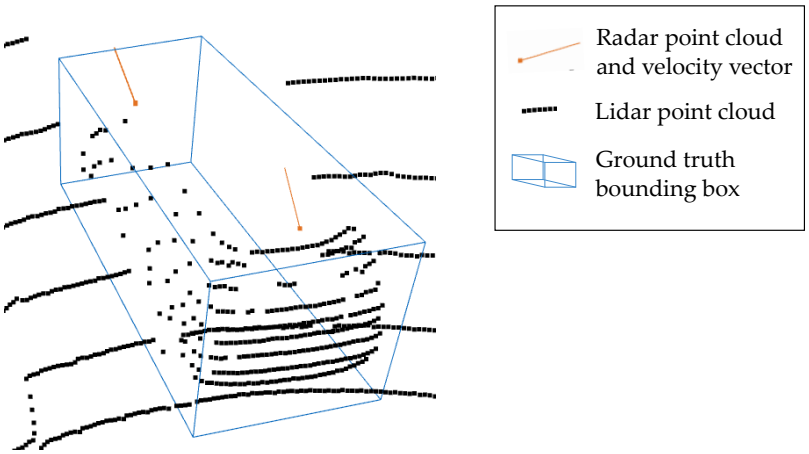


Figure 1. A vehicle detected by both radar and lidar sensors. The radar point cloud density is lower than the lidar point cloud density. Taken from nuScenes sample with the unique identifier token: 77fc24547ab34182a945eecb825b6576.

This paper compares the performance of different segmentation models on three subsets of the nuScenes data set; first on a data set distinguishing between vehicles, bikes, pedestrians and background detections; secondly on a data set only distinguishing vehicles from the background; thirdly on a data set that distinguishes moving vehicles from the background. The performance of the networks is limited by the input data quality. As data availability is one of the biggest barriers to public automotive radar research, we study the available nuScenes data set in detail and formulate requirements for an adequate radar data set for machine learning applications.

The contribution of the paper is fourfold:

- The paper adapts KPConv network architectures for radar point cloud data processing.
- The paper proposes modified LSTM network architectures to process irregular radar point cloud input data. The advantages and disadvantages of different time modelling and association techniques are discussed. However, an empirical study on public radar data does not motivate a preferred network architecture.
- The paper analyzes publicly available radar data for autonomous driving. It concludes that the data quality of the public data is not sufficient. Furthermore, it proposes which key points to consider when composing a radar data set for autonomous driving research.
- The code for this research is released to the public to make it adaptable to further use cases.

Section 2 discusses related work for lidar and radar point cloud processing. Section 3 describes the proposed models. The results are shown in Section 4 and discussed in Section 5. Our conclusions from the work are presented in Section 6.

2. Related Work

As object detection and segmentation networks often use similar backbones, we discuss both types of networks in this section. First, we give an overview of the state of the art in lidar point cloud segmentation and object detection. This motivates our choice for radar processing networks. Second, we present the state of the art in radar object detection and radar point cloud segmentation.

2.1. Lidar Point Cloud Segmentation

The task of point cloud segmentation describes the process of assigning a class label to every point of the input point cloud. As the input point cloud is a sparse, irregular 3D tensor, classic convolution neural networks cannot be directly applied to this form of input data. [4] therefore projects the data to a spherical image format to process it with 2D convolutions. The network consists of an encoder–decoder structure, to be able to generate a class label for every input pixel.

Similarly, [5] projects the point cloud data to a 2D frame. The paper augments the previously mentioned approach by taking into account a full 360° point cloud. Furthermore, it performs additional filtering for projection errors and provides an efficient implementation open source.

Qi [6] is the first to segment point cloud data without performing an intermediate feature transformation. The paper introduces the pointnet network structure, which can process an arbitrary number of input points by processing the points individually. However, their network does not consider neighboring points, as it is standard in convolutional neural networks. Their continued work [7] enhances the implementation to create local features from neighboring points while still operating on point cloud data directly. The applied operations still only process one point at a time.

Thomas [8] transfers the convolution approach from grid-based data to 3D point cloud data. The network processes a target point and its neighboring points by a defined number of kernel points analogous to 2D image convolutions. In this manner, spatial relationships can be learned directly from the input points by performing a so-called Kernel Point Convolution (KPConv).

2.2. Lidar Object Detection

3D Object detection networks estimate the position of 3D bounding boxes and associated classes in the 3D space. As they process the same input data as the segmentation networks, they can make use of the same backbone structure. Similar to point cloud segmentation, it is possible to project the point cloud data to a grid or voxel structure, or directly process the points of the point cloud to perform object detection from lidar data.

Liang [9] builds upon a spherical projected point cloud representation to extract features. In a second processing step, they project the input point cloud to a bird's-eye view (BEV) representation to estimate 3D bounding boxes.

Shi [10] presents the current state of the art in object detection according to the KITTI 3D Car leader board. They use both a projection to voxels followed by point-wise processing, to perform object detection.

VoxelNet [11] first generates point-wise features and then uses a max pooling operation to project the point features to a voxel grid structure. The object predictions are then generated by a convolutional detection head. Similarly, [12] combines pointnet and voxel-based processing to generate the object proposals.

Recurrent long short-term memory (LSTM) cells [13] are used for video processing [14] and object detection [15]. Lidar point cloud data, unlike video data, does not have a fixed structure over consecutive timeframes. When we look at a specific lidar point at a given

time, in the past time frame there might be no lidar return originating from the same location. The association of consecutive point cloud frames is therefore less intuitive than that of video data. [16] adapts LSTM cells for point cloud processing. They associate each point from the current point cloud with the past point cloud by learning a relation over the *k-nearest neighbors* and their relative distance to the current point. These association functions are used as the gate functions in the LSTM cell to create their PointLSTM to predict future point clouds from a time series.

2.3. Radar Point Cloud Processing

Radar signals can be represented in a similar data structure as lidar data. However, due to the greater sparsity and the lack of available radar data for algorithm development, the development in this area is less evolved. Many advances in the radar field are accomplished by industry research with direct access to labeled radar data. Recently, lidar processing techniques are transferred to the radar domain, taking into account the aforementioned differences between the two types of data.

Schumann [17] uses a two-step approach to classify radar objects. A DBSCAN algorithm [18] is used to cluster the radar detections. In a second step, features are generated for these clusters and classified by a random forest or a simple LSTM cell. The LSTM network outperformed the random forest approach. Difficulties with generating adequate training data for the LSTM network are mentioned. Furthermore, the drawbacks of the two-step approach and the manual corrections for the clustering algorithm are mentioned.

Another work [19] from the same group abstains from using a two-step clustering approach. Instead, they accumulate radar data over several time steps in a grid map. They create patches of 8×8 m, which are classified with a deep neural network approach. They train one-vs-all classifiers in a static environment to simplify the use case for the sparse radar data. The approach is evaluated on a proprietary data set.

Schumann [20] is the first to apply a pointnet-based approach to automotive radar data for semantic segmentation. They argue that due to the lower point cloud density grid-based discretization is not feasible for radar processing, as most grid cells would remain empty. Their approach segments different moving object classes against a static background class. It is important to note that moving objects cannot simply be distinguished from the environment by taking into account the doppler-measured velocity of the radar. Effects of an imperfect time synchronization, multi-path reflections and general noise induce non-zero doppler measurements for static locations. Furthermore, they stress the importance of taking into account the time domain for radar data processing.

A recent work by Schumann et al. [21] gives a comprehensive overview of a segmentation approach for both static and dynamic objects. They apply a convolutional encoder–decoder network on an extended grid map to classify static points. Moving points are segmented with a pointnet-based approach. The network integrates a recurrent structure by associating point features of past time steps with current point features. The time dependency is limited by the removal of old points from their memory point cloud to keep it at a fixed size. The approach is evaluated on a proprietary data set.

Palffy [22] performs radar point segmentation by using both the low-level radar cube and the processed radar point cloud data level. They evaluate their approach on the same proprietary data as [17] and show that they can set a new benchmark score by taking into account the additional data source.

Danzer [23] also uses a pointnet approach. Their network first segments patches of the environment for possible objects. Consecutively, the network processes the segmented object points to regress the bounding box dimensions for object detection. As with the other works, they evaluate their approach on a proprietary data set.

3. Methodology

Taking into account the related work above, we emphasize the importance of the following points for the development of a semantic segmentation network for radar data:

- *Data Availability*: In contrast to lidar research, where a vast amount of labeled data is publicly available, the access to radar data is more restricted. Accurately labeled data is the basis for any successful supervised learning model, which means that reduced data availability could be one reason for the smaller amount of research work in this area. A more detailed view of radar data available to the public is given in Section 5.2.
- *Data Level*: Due to the sensor principle of the radar, a variety of intermediate processing stages occur before the raw receive signal is converted into object proposals.
  The unprocessed low-level raw antenna signals theoretically contain the most information; however, there are no methods of labeling data on this abstraction level for object categories. Additionally, depending on the antenna characteristics and configuration, the data would differ a lot for different radar types, which would require a specific learning approach for different sensor hardware designs. Consequently, to the best of our knowledge, no one-step learning approach exists for classifying objects directly on the raw antenna signals.
  One can perform object detection at the radar cube level or on any of its 2D projections, e.g., range-azimuth plot. However, data labeling of this 3D representation (2D location + velocity) would still be a tedious task. [22] performs the labeling by extracting data from the cube at manually labeled point target locations. However, in this way the label quality can only approximate the ground-truth, as the point target might not include all raw signals that originate from an object. Furthermore, the point targets only represent a compression of the radar signal, which means that the point signal cannot be re-projected precisely on the cube-level signal. To our knowledge, no full 3D cube-level radar data set has been released to the public. However, due to the high informativeness and the similarity to image data, which is heavily processed with learning-based techniques, the cube-level data is an interesting use case for future deep-learning research on radar data.
  Point level data is the most explored data level for deep learning on radar data. It represents a compromise between data informativeness and data amount. Due to the similarity to lidar point cloud data, algorithms can be transferred from the lidar domain to the radar domain. This work operates on point level radar data. We benchmark our models on the public nuScenes data set [2], which includes labeled point cloud radar data.
- *Data Sparsity*: For lidar data processing, both grid and point-based approaches, are used in the state of the art. Similar to [20], we argue that for radar data, a point-based approach is more feasible than the intermediate grid representation as most of the grid cells would remain empty. For fusion approaches of radar data with denser data such as images or lidar both grid-based and point-based approaches [24–26], might be suitable. This work presents a point-based approach. To the best of our knowledge, we are the first to adapt the KPConv architecture for radar point cloud data.
- *Time Dependency*: Due to the low spatial resolution of radar sensors, the time domain plays an important role in radar processing. Current production radar systems track point level detections over several timeframes before recognizing a track as an object. This decreases the number of false positive object detections due to clutter. These false positives would otherwise be harmful to driver assistant systems such as adaptive cruise control (ACC) or emergency brake assist. In the literature, approaches that use the time domain in learning models such as simple LSTM cells or memory point clouds have proven successful for radar data processing. This paper integrates KPConv layers into an LSTM cell, which we call KPConv-based LSTM (KPLSTM). In this way, we can encode the time dependency for points individually. Additionally, we combine KPConv encoding with LSTM and ConvLSTM cells [27] in the latent space to integrate the time domain in the model on a global feature level.
- *Moving Object Recognition*: In addition to the time dependency, production applications only react to moving objects, as the amount of clutter among the static points causes

additional challenges for the processing. It is desired to bring radar processing to a level of maturity to detect static objects just as with the lidar sensor. However, the research above shows that the focus is still mainly on the moving object case. Even in the simplified moving objects scenario, the detections are only reliable enough for simplified use cases, such as vehicle following in the ACC case. In this work, we evaluate our network for both static and dynamic cases and show the disadvantages of the radar for the complex static scenarios.

- *One-vs-all Classification*: Due to the difficulties in creating reliant object detections from radar data, one compensation strategy is the simplification of the scenarios. In the literature, one-vs-all classification are an efficient way to simplify the training use case, while keeping a relevant task in focus. This paper shows the results for different training configurations and their impact on the segmentation performance.

Radar *Data Availability* is discussed in Section 5.2, The *Data Level* is given as only point level data is available. *Data Sparsity* and *Time Dependency* are major considerations for the design of the KPConv-LSTM networks presented, which are discussed in Section 3.1. The evaluation of focusing on *Moving Object Recognition* and *One-vs-All Classification* is shown in Section 4.

3.1. Model

The proposed radar models are inspired by lidar processing and general LSTM networks from literature. The models are implemented in a modular fashion so that different architectures, e.g., for the encoder or decoder part of the network, can be replaced and combined for greater model flexibility. For additional details and analysis, we refer to the master thesis of Fent [28] as the main contribution to the implementation. In the following, we recapitulate the ideas of the KPConv layer and describe our LSTM extensions, including a KPConv-based LSTM (KPLSTM) cell. The proposed models can process radar input points clouds of varying size. The KPLSTM cell associates point clouds of varying sizes over consecutive time steps. In contrast to existing LSTM cells, we incorporate the time dimension as early as in the encoding part of the network.

3.2. KPConv with LSTM Cells in Latent Space

The KPConv layer is inspired by the KPConv model introduced by [8]. The idea behind the Kernel Point Convolutions is to model the spatial relationships in a point cloud directly. Sliding a standard convolutional kernel over an unordered space is not feasible. The KPConv layer applies convolutional weights at Kernel Point locations with a distance factor to neighboring points of the input point cloud. The number of kernel points K is fixed, while the number of neighboring points N_x and the respective association function h_{ij} varies with the number of points in the local neighborhood. The considered neighbor points lie within a specified radius r from the center point of the convolution. The output feature dimension d_{out} for every center point is determined by the number of KPConv filters applied. Here, the KPConv layer follows the same principle as grid-based convolutions. Figure 2 shows a graphical representation for the processing of a center point x.

Due to the sparsity of the radar data, the maximum radius for neighbor points that affect the feature output of the current center point in the first KPConv layer is set to 8 m. In the original KPConv implementation [8] for lidar data, this radius is set to a maximum of 0.15 m for outdoor scenarios. The larger radius empirically provided the best results on our data set. From a theoretical perspective, the greater sparsity of the radar data motivates a greater radius of influence to include sufficient information of the surroundings. In this radius, all radar reflections of a passenger vehicle could influence all other reflections from the same object. For larger objects, such as trucks and buses, this is not always the case. Nonetheless, an even greater radius did not result in better detection scores, as at the same time more clutter information could be associated in the neighborhood. Figure 1 shows that radar points do not necessarily originate from the visible edges of an object.

Appl. Sci. **2021**, *11*, 2599

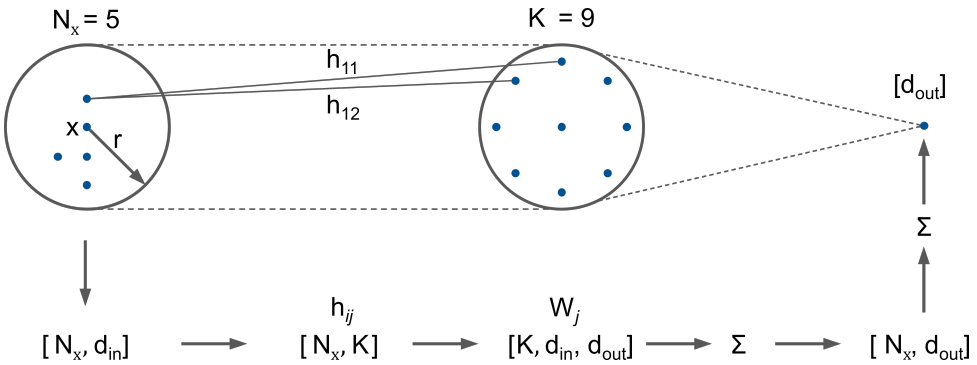


Figure 2. Processing of the center point x with d_{out} kernel point filters. Based on [8].

Figure 3 shows the high-level model structure of a KPConv model with a ConvLSTM center layer (KPConv-CLSTM). The inputs to the network are the n input radar points consisting of the last three radar time steps with five feature dimensions $feat$: Location x, y, z; radial velocity components v_x, v_y; and the radar signal strength, called radar cross section RCS. Following the original implementation, each *KPConv ResNet Block* consists of an MLP layer, a KPConv layer and another MLP layer. The skip link branch consists of an MLP layer to adapt the feature dimension before adding it to the processed KPConv branch. The number of points is compressed with a farthest point sampling method. Once the input point cloud has been encoded, convolutional LSTM cells [27] are applied to the features in the latent space representing the entire input point cloud. The decoding or upsampling of the point cloud is performed by a combination of three-nearest neighbors upsampling and MLP layers for feature generation. When features have been obtained for every input point, three MLP layers serve as a classification head. The upsampling and classification head is inspired by [20]. A class-weighted focal loss function [29] is used for training to handle class imbalances, e.g., in the *Moving Vehicle Data Set* used below over 97% of the points belong to the background class.

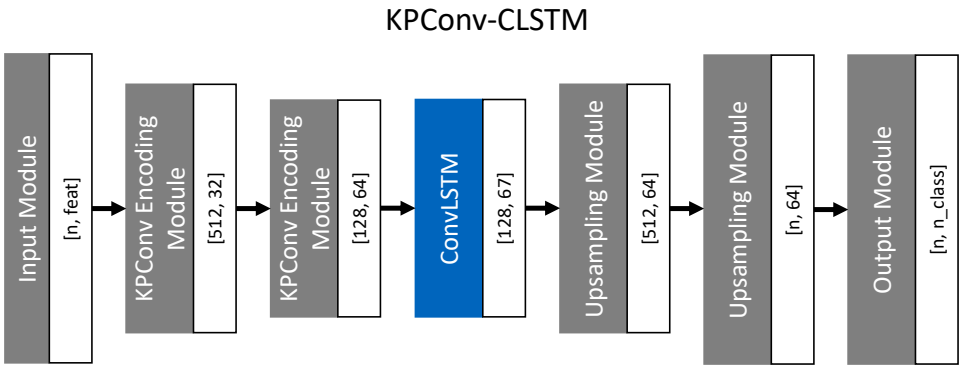


Figure 3. High-level structure of our KPConv model with ConvLSTM cells in the latent space. The blue color shows layers that consider time dependencies.

Due to the modularity of the model, we can replace the ConvLSTM layer in the latent space with a standard LSTM cell or omit the time dependency altogether for our evaluation in Section 4. Additional details of the implementation can be found directly in the provided configurations in the repository released with this publication.

3.3. KPLSTM

From an architecture point of view, the KPConv-CLSTM comes with the drawback that the time dependency is only applied on the global feature of the point cloud in the latent space. A direct application of the LSTM cells to the input points would not be feasible

Appl. Sci. **2021**, *11*, 2599

due to the irregular structure and size of the input point cloud over several time steps. Schumann [21] associates points of the current radar point cloud with neighboring points from a memory point cloud by using the grouping method of PointNet++, using only the current point cloud as center points. Similar to [16], we propose a new variant of the LSTM cell in such a way that it can process and associate point cloud data directly. Instead of pointnet-based processing, we use KPConv kernel gates for the KPLSTM cell. To align the dimension of the past and present point clouds, we explore different sampling strategies from nearest neighbor sampling to learnable sampling via the KPConv weights. For the use case of the paper, the nearest neighbor sampling proved most reliant. Figure 4 shows a comparison of a standard LSTM cell and the KPLSTM cell.

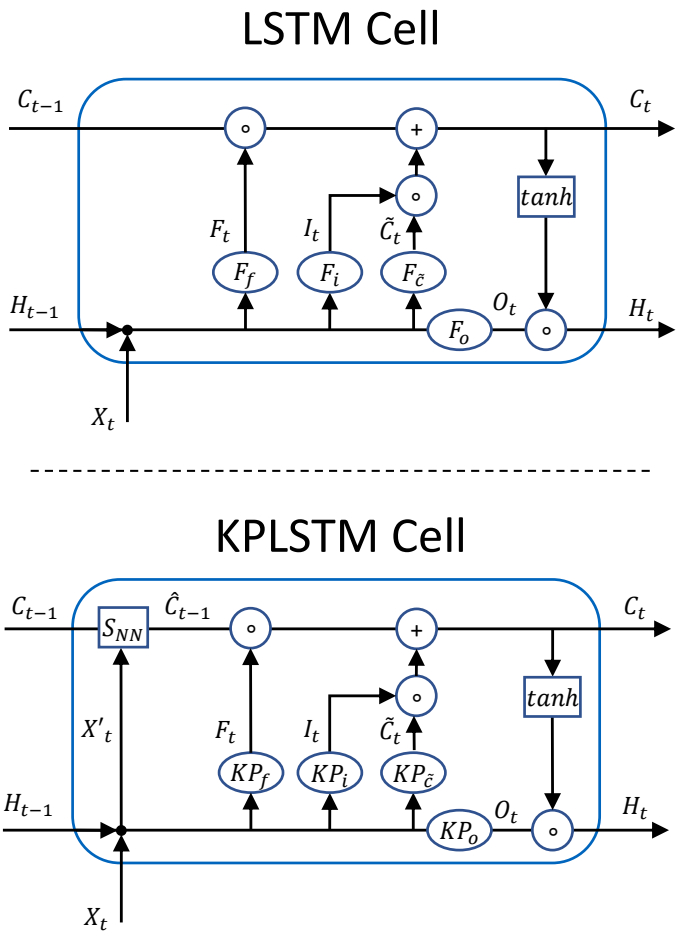


Figure 4. Comparison of standard LSTM cell and KPLSTM cell.

The inputs to the LSTM cells are the old cell state C_{t-1}, the old hidden state H_{t-1} and the current point cloud X_t. The output of the cells are the current cell state C_t and the current hidden state H_t. For the input data, the old hidden state and current point cloud are concatenated. This structure is the same as in the standard LSTM cell. Instead of fully connected layers, KPConv operations KP_x are used to model the input gate I_t, output gate O_t, forget gate F_t and internal cell state $\tilde{C}_t$. The subscript x denotes the respective weights. The KP_x layers perform feature generation and point cloud association of the old and current coordinates at the same time. Additionally, the old cell state C_{t-1} needs to be mapped to the new point coordinates. As we do not want to apply the features of the new point cloud but only propagate the old features to new coordinates, we take the current point coordinates without the point features X'_t and use them as the center points for the

Appl. Sci. **2021**, *11*, 2599

nearest neighbor sampling S_{NN} to obtain the intermediate cell state $\hat{C}_{t-1}$. The remaining structure is analogous to the standard LSTM cell. The symbol $\circ$ denotes the element-wise product. We remove old features from the cell state via the forget gate and add new ones via the input gate. The equations describing the KPLSTM cell are as follows:

$$\hat{C}_{t-1} = S_{NN}(X'_t, C_{t-1}), \tag{1}$$
$$I_t = \sigma(KP_i(X_t, H_{t-1})), \tag{2}$$
$$F_t = \sigma(KP_f(X_t, H_{t-1})), \tag{3}$$
$$O_t = \sigma(KP_o(X_t, H_{t-1})), \tag{4}$$
$$\tilde{C}_t = \tanh(KP_{\tilde{c}}(X_t, H_{t-1})), \tag{5}$$
$$C_t = F_t \circ \hat{C}_{t-1} + I_t \circ \tilde{C}_t, \tag{6}$$
$$H_t = O_t \circ \tanh(C_t) \tag{7}$$

Figure 5 shows the overall KPLSTM network architecture. The encoding is performed in two consecutive blocks. Each block comprises a KPConv layer to downsample the point cloud, a KPLSTM cell and another KPConv layer for feature generation. The upsampling is done in the same manner as in the KPConv-CLSTM. It is worth mentioning that the time dependency is already introduced in the network encoding layers, making an intermediate global feature LSTM model obsolete. With this, we capture local structure over time instead of only being able to keep track of the features globally.

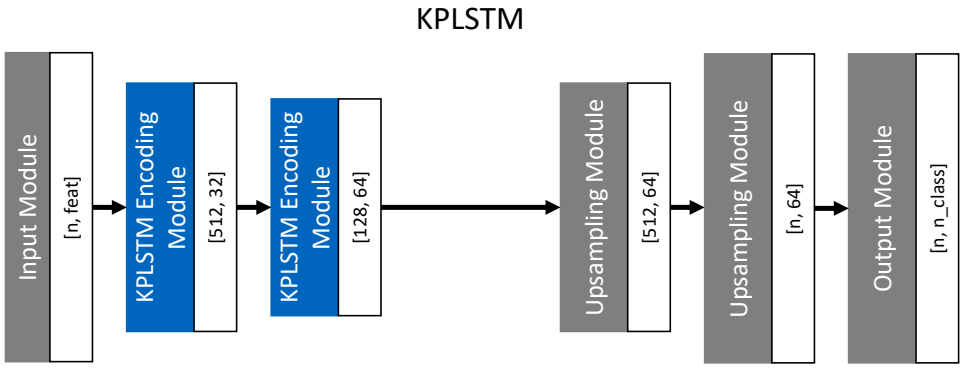


Figure 5. High-level structure of the KPLSTM model. The time dependency is already considered in the encoding stage as indicated by the blue color-coding.

3.4. Data Processing

The nuScenes data is labeled using lidar data. Radar detections are not time synchronized with the labels. Due to the time delay, bounding box labels do not always coincide with the radar detections. Further errors can be introduced through slight angular miscalibrations of the radar sensors, which has a particular impact on far-range detections. The transformation from the radar and lidar frame to the car coordinate system can be another source of spatial offset between ground-truth (GT) bounding boxes and radar data. To account for spatio-temporal calibration errors, nuScenes added the so-called *wlh*-factor to increase the bounding box size. For small bounding boxes, e.g., pedestrians, increasing the bounding box size by 50% might be a reasonable factor, whereas this would include too much environment information in the case of vehicles or buses. Similarly, the *wlh*-factor scales the tolerance for the length and width of vehicles differently, as usually the length of a vehicle is larger than its width. We propose an absolute *wlh*-tolerance in meters that is equally suited for all object categories. Figure 6 shows the misalignment of bounding boxes and the effect of the *wlh*-factor vs the *wlh*-tolerance. Although choosing a higher *wlh*-factor could also include respective points in the bounding box in Figure 6b, this would; however, increase the tolerance in longitudinal direction too much.

Due to the sparsity of radar data, we process three consecutive time steps as a single point cloud. As these past time steps are not available for the first sample of each scene of the public data set, we omit the first sample, both for training and evaluation.

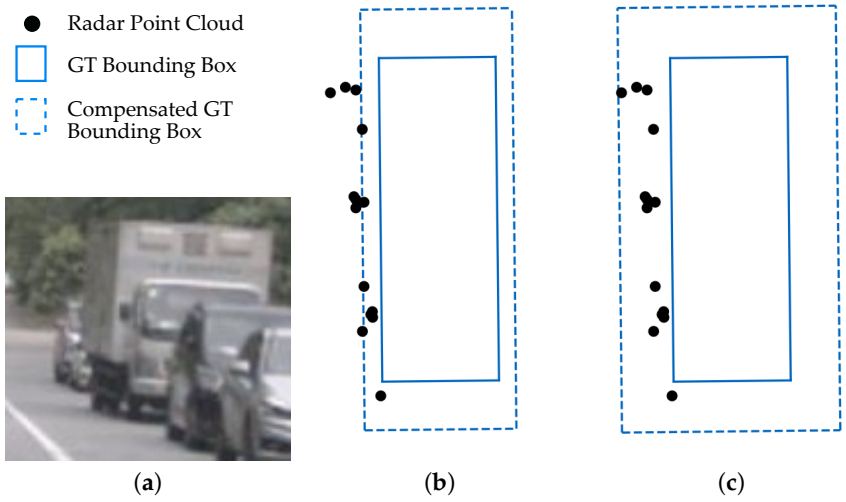


Figure 6. Compensating the misalignment of the manually labeled ground-truth bounding box with a *wlh*-tolerance creates a more precise ground-truth for the network training. (**a**) The truck of interest in camera view. (**b**) BEV: The *wlh*-factor = 1.3 includes only 5 points in the compensated GT bounding box. (**c**) BEV: The *wlh*-tolerance = 1.2 includes all 14 points in the compensated GT bounding box. NuScenes sample token: a3c6db2751a54c8590c4d8241e01ac8c.

3.5. Training

We train the segmentation networks on the official training split of the public nuScenes data set. We test the performance of the models on the official validation split as nuScenes does not provide segmentation ground-truth for their test set. We do not perform hyperparameter optimization on the validation split. During the training process, we batch several scenes for one training step. We keep the order of the samples inside each scene, to be able to learn the time dependency of the radar data.

4. Evaluation

The model performance is measured with the macro-averaged F1-score as in the works of [20–22]. However, none of the related work benchmarked their implementations on a public radar data set. To compare our approach to the literature, we implement a PointNet++ approach in the style of [20]. We use our implementation of the PointNet++ approach as the state-of-the-art baseline as there are no radar pointnet implementations publicly available.

4.1. Traffic Data Set

First, the models are trained on a data set distinguishing between the classes: vehicle, cycle, pedestrian and background. The classes consist of moving and static objects alike. Figure 7 shows the confusion matrices for three different class weight configurations. The *Veh Weights* configuration shows a strong bias towards the vehicle class, whereas almost no points are classified as pedestrians or bikes. For the *Lower Veh Weights* configuration the class weight on the vehicle class is halved. We see a better fit for the background class in this way. Additional equilibration between the class weights lead to the result of the *Distributed Weights* confusion matrix. All classes are associated while the overall quality stays low. The radar data is too sparse to reliably distinguish between four classes.

Appl. Sci. **2021**, *11*, 2599

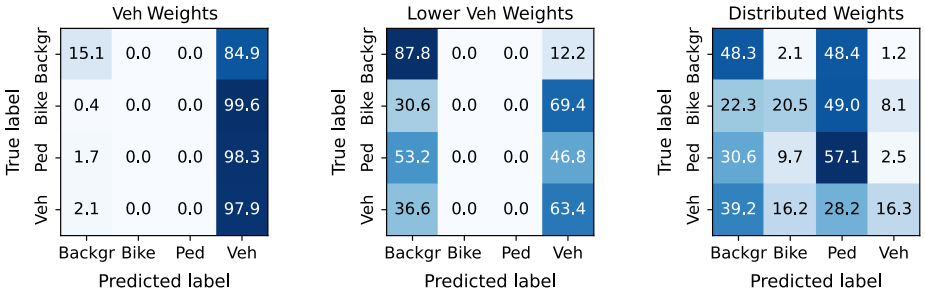


Figure 7. Segmentation confusion matrices for different class weight configurations of the data set classes.

4.2. All Vehicle Data Set

In the following, we therefore limit the segmentation task, distinguishing between vehicles and background in the *All Vehicle Data Set*. The left column of Table 1 shows the resulting F1-score for different model configurations on this data set. The different network architectures are trained with the same hyper parameters to enable comparability. Due to the low data density in the underlying data set, a fitting parameter set needs to be established first to generate reasonable results for any configuration. Due to the high class imbalance, a vanilla network does not produce a reasonable learning result as all classes would just be segmented as background. However, after a fitting configuration is found it becomes evident that the different models reach more or less the same result. While experimenting with different learning strategies, we found that there is no clear ranking between the models, but that the performance levels out in the range of the F1-scores shown in Table 1. It seems that the radar data basis does not include enough discriminative features to separate the classes further. The low data density and quality limits the performance for the proposed models.

Table 1. Segmentation macro-averaged F1-scores on the nuScenes validation data set.

Network	All Vehicle Data Set	Moving Vehicle Data Set
PointNet++	59.91%	75.83%
KPConv	59.88%	74.68%
KPConv-LSTM	59.69%	75.81%
KPConv-CLSTM	60.05%	75.42%
KPLSTM	57.89%	75.34%
KPconv w/o vel	52.36%	52.15%

4.3. Moving Vehicle Data Set

Additionally, we evaluate the models for segmenting moving vehicles against the background in the *Moving Vehicle Data Set*. The moving vehicle class compromises all bounding boxes with the nuScenes attribute *vehicle.moving*. This includes cars, trucks and buses. Table 1 shows the results for this data in the right column.

The results show that moving vehicles can be distinguished from the background more successfully than static vehicles, due to the overall higher F1-score achieved. At the same time, the performance levels out at higher overall scores for this type of input data.

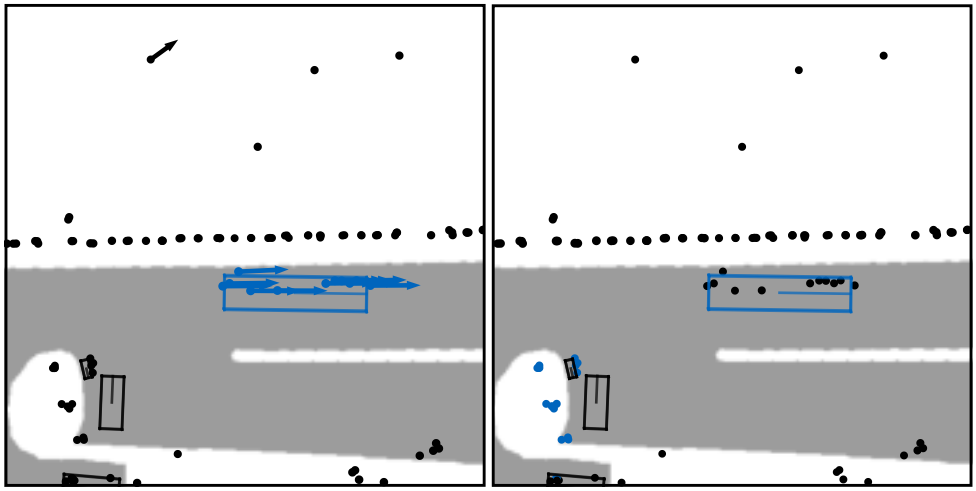
4.4. Velocity Input Feature Study

As the velocity component is a strong feature used in classical radar processing, we evaluate its relevance to the result of our segmentation model. We retrain our KPConv model with a reduced input feature space without velocity information. The results are shown in the last row of Table 1. Without the velocity information, the F1-scores are

Appl. Sci. **2021**, *11*, 2599

significantly lower for both data sets. Especially for segmenting moving vehicles, the velocity dimension seems to be a discriminative feature. Without using the strong velocity feature, the segmentation quality of moving and static vehicles reaches a similar score.

Figure 8 shows a qualitative comparison of the segmentation result of a moving bus with and without the velocity feature dimensions. The radar returns from within the bounding box are segmented correctly in Figure 8a. The radar return in the top of the figure, which also comprises a non-zero relative velocity, is correctly segmented as background. Figure 8b shows that the segmentation failed when not using velocity features. Points in the lower left part of the figure are segmented as a moving vehicle. Geometrically, these points show a great similarity to the points of the moving vehicle or even the wall along the road. The RCS feature is not discriminative enough to compensate for this fact.



(**a**) (**b**)

Figure 8. Classification of a moving vehicle. Ground-truth bounding boxes and radar classifications are color-coded. Blue: moving vehicle. Black: rest. The segmentation in (**a**) is more precise than the segmentation in (**b**) due to the inclusion of velocity features. NuScenes sample token: 9a7dc9c8adce4ae68ca26ad5b5f93366.

5. Discussion

A definite conclusion of the model performance cannot be inferred from these results above. The data quality constrains the model performance, meaning that the potential of the KPConv models, as proven for lidar data, could not be shown for the underlying radar data. However, the main reason for this does not seem to lie in the model itself but rather in the data. In recent publications of radar deep-learning processing, the data source is only mentioned briefly. However, many important considerations, when dealing with radar data and especially the particularities of the nuScenes data set, are not discussed in the literature. We believe there is a need to shift the focus from solely the model architectures to the data itself to develop functioning radar deep-learning applications in public research. Therefore, in the following discussion, we not only carefully assess our model, but add a focus on the data side and discuss barriers for the widespread application of machine learning techniques for radar data.

5.1. Model Critique

Due to the low information content in the underlying radar data, model training requires a careful calibration. Although higher segmentation scores could not be achieved, slight variations in learning rate could result in models barely outperforming a guessing model. The low radar density is partly the result of the radar processing itself. In the range-velocity dimension of the radar cube, moving objects are more differentiable from the

Appl. Sci. **2021**, *11*, 2599

static background than static vehicles. Consequently, many labeled static vehicle bounding boxes do not include any or only a few radar points. This is one reason the moving vehicle segmentation performed better than the static case.

The presented LSTM models, in comparison to [21], do not forget old points once an intermediate memory point cloud is full. The data is added or forgotten by the memory point cloud according to the LSTM structure. We assume this to be an advantage of the presented architecture. However, due to the different underlying data source, a direct comparison of the models is not possible here.

We experimented with different functions to associate consecutive radar point clouds in the KPLSTM cell. In general, we would expect learning methods to provide the most accurate data association capability on noisy radar data. However, nearest neighbor sampling outperformed learning-based methods on the presented data set. We assume that the learning of an appropriate association function is not feasible due to the corruption of the radar data in the data set. A detailed study of different learning methods for data association is postponed until an appropriate data is available.

Radar data comes with a class imbalance towards the background class. Most radar reflections originate from non-traffic participants, which makes a high accuracy score possible through the classification of all points as background. The class-weighted focal loss function helps mitigate this problem, by assigning strongly misclassified examples of the minority classes with a higher weight during backpropagation.

At this stage, it is difficult to pinpoint whether the model or the data has the greatest effect on drawbacks of the model performance. To further study the discriminative content in the data set, we overfit the KPConv model to two select scenes from the nuScenes data set. When training a model to segment static and moving vehicles alike, a perfect score is not achieved within 10,000 epochs of training with the same scene. Figure 9a shows the resulting confusion matrix for a *All Vehicle* configuration. Figure 10 shows an extract from this scene. The radar returns of the vehicles have a high similarity to points returned from the background, making even overfitting to the data a challenging task.

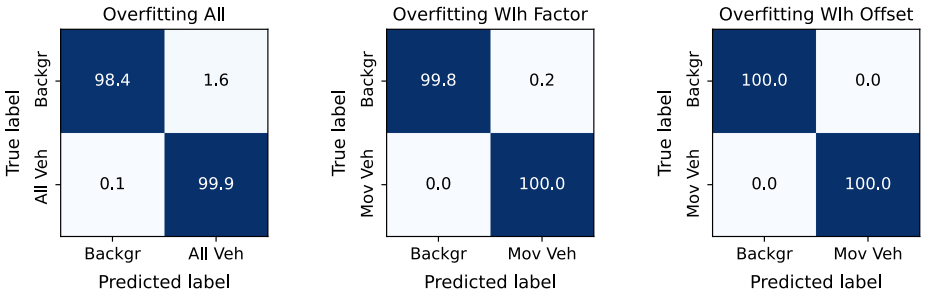


Figure 9. Overfitting confusion matrices. (**a**) Perfect overfitting is not achieved for the *All Vehicle* configuration scene on the left. (**b**) The *Moving Vehicle* data scene is not perfectly segmented when a *wlh*-factor is used. (**c**) Training with the same scene as in (**b**) results in a perfect score when an adequate *wlh*-tolerance is used.

For moving vehicle segmentation, we were able to achieve a perfect segmentation for a scene of a vehicle following scenario. Figure 9b,c shows the confusion matrices for this scene with two different compensations methods for the spatial misalignment. Please note that it was only possible to achieve a perfect score when a *wlh*-tolerance was applied in Figure 9c. When a *wlh*-factor was used in Figure 9b, not all points from the moving object class were correctly associated with the ground-truth bounding box. The model learned to segment these points to the moving vehicle class due to its similarity to points inside the ground-truth box; however, they count negatively towards the metric score.

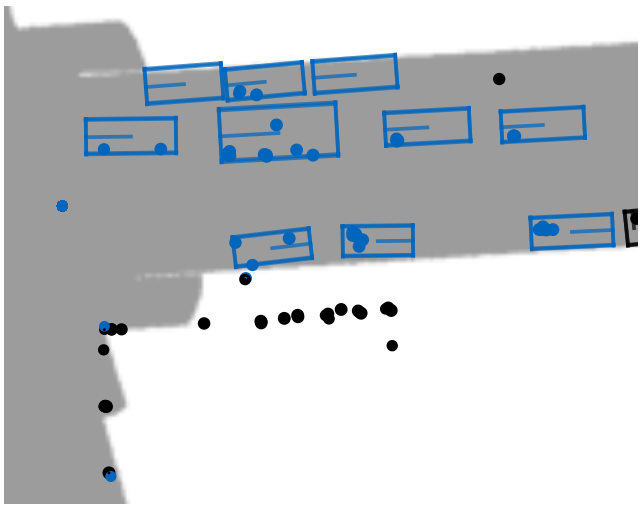


Figure 10. Even in the overfitting scenario, misclassifications are present when static objects are segmented. NuScenes sample token: 205c60f1248343a09bf4c6b6c05d8525.

Labeling bounding boxes from radar data directly is a challenging task itself due to the data sparsity. Nonetheless, if we want to create machine learning models for radar data for a real-world application, it would be beneficial to adapt the lidar bounding boxes to the radar data during the labeling and have sets of ground-truth labels for the sensor modalities independently. Thus far, the *wlh*-tolerance has helped mitigate the problem, but this cannot be the solution when highly precise results are necessary in production.

For the overall performance, further studies with selected data would be necessary to investigate the source of errors in the segmentation result. Due to the overall data quality of the nuScenes radar data, however, it is debatable whether the creation of a high-quality subset of the data set is worth the effort or even achievable. The next section discusses this in detail and motivates our choice to refrain from further optimization due to the quality of the training data.

5.2. Data

This section discusses the particularities of radar data and the nuScenes data set. Data sparsity is a general difficulty when dealing with radar data. Though [21] shows favorable results on a private data set, it becomes clear that even with high-quality-high-resolution input data, the challenge of general semantic segmentation on automotive radar data is far from solved.

In 2019, a review [30] stated that they expect more radar data sets to be released in the future. However, at the start of 2021, only nuScenes [2], Astyx HiRes2019 [31] and DENSE [32] data sets provide point level radar data with 3D bounding box annotations. The HiRes2019 data consists of only 546 frames, which makes training of a learning model on this data impractical. The DENSE data set uses an outdated radar model with even greater sparsity than the nuScenes data set while providing fewer labels. The nuScenes data set consists of 40,000 annotated frames of a production radar. It is the only data set that contains the consecutive samples needed for training time-dependent models. In conclusion, this is the only reasonable public data set for deep-learning applications of radar data.

Despite its unique position, nuScenes has some major drawbacks regarding its radar data. Scheiner [33] compared the performance of their models on high-resolution and production radar data. They state that the nuScenes radar density is far worse than that of their production radar. A comparison of the specifications of the radars of [2,33] and shows that they use a comparable or even the same radar model as nuScenes. The huge differences in data density between the two data sets cannot be explained by the radar

Appl. Sci. **2021**, *11*, 2599

hardware alone. We calculated that over 40% of the annotated vehicles are not covered by any radar detection when considering three radar sweeps for a nuScenes sample. For vehicles labeled as moving, the same holds true for over 24% of the labeled bounding boxes. Consequently, bounding boxes of objects detected by the radar sensor, include fewer point returns than expected from this sensor type. In the following, we look at some factors that decrease the suitability of the nuScenes data set for the development of radar algorithms and conclude with the aspects that a radar data set should consider.

The nuScenes data set includes information about the number of radar points per annotation. However, this number does not distinguish between its five different radar modules. The labels include information about whether or not a vehicle is moving. Although this information is correct for many cases, by manually examining random samples of the data set, we encountered a significant number of annotations where moving objects were labeled as static objects and vice versa. This not only negatively impacts the training, but also impacts the evaluation scores of moving object detection, which is the main application of radar data. Figure 11 shows the ground-truth classes from a scene of a bus moving towards the sensor vehicle. Even though the bus is approaching at a high speed and is close to the sensor vehicle, it is wrongly labeled as a non-moving vehicle.

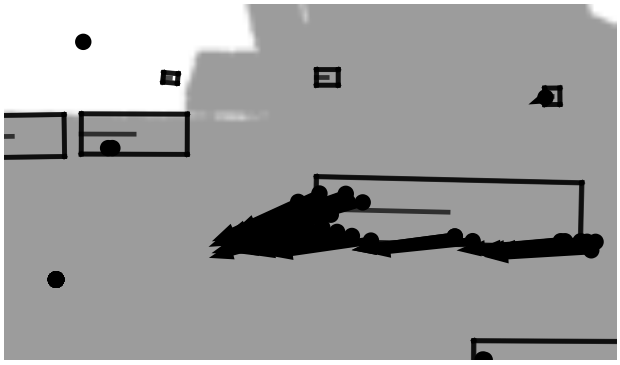


Figure 11. A bus approaching the sensor vehicle in BEV projection. The black color of the ground-truth points and bounding box indicates a static vehicle, despite the bus clearly approaching as indicated by its measured relative velocity. NuScenes sample token: 87a927d7e61345d3a098eb66908bddcb.

The radar data is limited to 125 detections per sweep by the sensor interface. Additional detection filters are applied in the standard configuration of the data set API of nuScenes. The hardware limitation to 125 detections is enforced by cutting off detection signals at an *RCS* value of less than $-5\,\mathrm{dB\,m^2}$. The works of Yamada and Yasugi [34,35] measure the expected *RCS* value of a pedestrian at a distance of less than 10 m to be less than $-8\,\mathrm{dB\,m^2}$ for 76 GHz and 79 GHz radar sensors, respectively. The chosen cut-off value thereby likely filters out many pedestrian detections that cannot be used for training even when the additional filter settings in the data set API are deactivated. By visualizing the data set, we found that not only pedestrian detections are affected by this, but also passenger vehicle are often not detected in the data set. Figure 12 shows a vehicle in line-of-sight direction from the radar sensor. However, no radar points are present inside or near the ground-truth bounding box.

Appl. Sci. **2021**, *11*, 2599

(**a**) (**b**)

Figure 12. Vehicle in line-of-sight direction from the radar sensor. Despite its exposed position, it is not detected with the sensor settings used. (**a**) The vehicle in camera view in front of the sensor vehicle. (**b**) BEV: Zoom onto the GT bounding box in blue and the radar data in the surroundings. No radar return originates from the vehicle. NuScenes sample token: bbba31d91e334f4abf6d1f96bf699980.

While introducing the *wlh*-tolerance, we mentioned that the ground-truth labels are not always accurate regarding the radar returns. Radar reflections seem to originate from outside the ground-truth bounding boxes. The *wlh*-tolerance can mitigate this problem but not completely erase it. No matter the choice of a bounding box tolerance, there will always be returns that lie outside of it and will either "confuse" the learning algorithm or count negatively towards the segmentation score, while seeming to be classified correctly intuitively.

The biggest artificial barrier for successful training that we encountered is the absence of labels for objects that are clearly visible in radar view. Figure 13 shows several moving vehicles from a BEV perspective. Although the radar points on the right are correctly labeled as moving vehicles, the vehicles further to the left in the scene, on both sides of the road, are not labeled. Their presence can be estimated from the point clusters with non-zero velocity. The nuScenes labels are created from lidar view. The resulting number of samples with missing labels for radar data limits the data set quality.

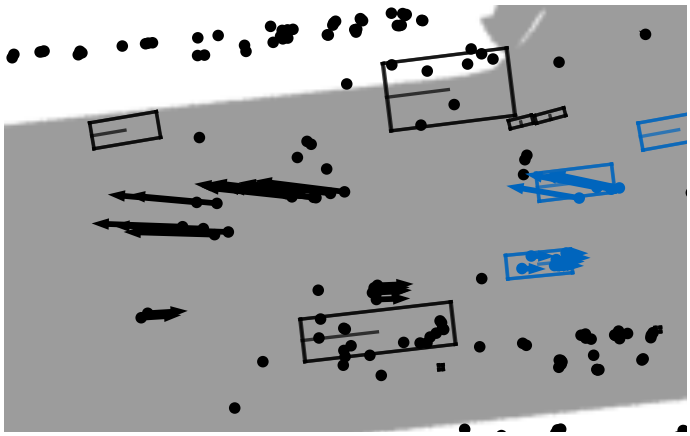


Figure 13. Moving vehicles ground-truth color-coded in blue. Ground-truth bounding boxes are not labeled for four vehicles in the sensor field of view, though they are clearly distinguishable by their relative velocity. NuScenes sample token: e2c7c91b4ea2462090d866539ff6b9e5.

The nuScenes data set is a potent data set for measuring the quality of lidar and camera detection algorithms. The effects mentioned decrease the data quality for the training of radar algorithms significantly. Nonetheless, it is an important contribution to the radar community as it remains the only usable 3D-labeled source of radar point cloud data available to the public. Although it can serve as a comparison benchmark for radar detection algorithms, it cannot measure the absolute performance to be expected

when the algorithms are applied to a real-world use case with a comparable radar module. Slight corrections, such as using the *wlh*-tolerance or filtering for samples with high data density, can be performed to increase the data quality. The creation of a production quality data set, however, would require manual filtering of the instances and possibly even relabeling to create a generally applicable sub data set. This kind of complete revision of the labels seems unreasonable, considering the low raw radar data quality in the data set.

We therefore see a great demand for the release of a data set tailored for radar object detection. This would make it easier to compare different radar processing approaches. The absolute model performance that is measured on private data sets can hardly serve as a metric for comparison due to the huge impact of the underlying data set on the segmentation result. In an ideal data set, the dimensions, the classes and movement properties of the objects are a focus of the scene selection and have been critically revised. A high-resolution radar should be used and its calibration evaluated before data recording. Filtering of radar points, when required by the sensor interface, should be performed, e.g., along the longitudinal and lateral distance dimension to include all nearby objects of interest in the data. As a uniform data density is preferred for model training, sensors around the vehicle should possess little field of view overlap. If sensor overlap is preferred for important directions, e.g., in front of the vehicle, these areas should be separated from remaining areas during training and inference to preserve the homogeneity of the data. lidar data can be used to help the labeling process, though the final labels should be set regarding the radar data.

6. Conclusions and Outlook

This paper develops and benchmarks KPConv-based segmentation networks on radar data. Due to the sparsity of radar data, we motivate the use of direct point processing and one-vs-all segmentation strategies. Furthermore, we model the time dependency during point feature encoding within a KPLSTM cell and in the global feature space with standard LSTM cells. Despite their theoretical motivation, the models could not outperform a pointnet-based network on the underlying data set. Advantages and disadvantages of models are discussed. An in-depth analysis of the data set shows that the public data source itself is a major performance barrier for the models. Absolute performance metrics cannot be measured on currently available public radar data sets. The slower progress of development for radar processing in comparison to lidar data processing can partly be attributed to the lack of publicly available data. Building a high-quality data set of automotive radar data is therefore a major steppingstone in accelerating the progress of radar processing development. We expect the proposed models to show their increased potential when high-quality radar data is available. For that, we make the code for the proposed network architectures and the interface to the nuScenes data set available to the public at: https://github.com/TUMFTM/RadarSeg (accessed on 6 March 2021).

Author Contributions: F.N. as the first author, initiated the idea of this paper and contributed essentially to its concept and content. Conceptualization, F.N. and F.F.; methodology, F.N. and F.F.; software, F.F. and F.N.; data curation, F.F. and F.N.; writing—original draft preparation, F.N.; writing—review and editing, F.F., J.B. and M.L.; visualization, F.N. and F.F.; project administration, J.B. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: We express gratitude to Continental Engineering Services for funding for the underlying research project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets Robotics: The KITTI Dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
2. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR2020), Washington, DC, USA, 16–18 June 2020.
3. Yoneda, K.; Suganuma, N.; Yanase, R.; Aldibaja, M. Automated driving recognition technologies for adverse weather conditions. *IATSS Res.* **2019**, *43*, 253–262. [CrossRef]
4. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a lidar Point Cloud. In Proceedings of the International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
5. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. RangeNet ++: Fast and Accurate lidar Semantic Segmentation. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 4213–4220. [CrossRef]
6. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
7. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv* **2017**, arXiv:1706.02413.
8. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
9. Liang, Z.; Zhang, M.; Zhang, Z.; Zhao, X.; Pu, S. RangeRCNN: Towards Fast and Accurate 3D Object Detection with Range Image Representation. *arXiv* **2020**, arXiv:2009.00206.
10. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
11. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
12. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. STD: Sparse-to-Dense 3D Object Detector for Point Cloud. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
13. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neur. Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
14. Finn, C.; Goodfellow, I.; Levine, S. Unsupervised Learning for Physical Interaction through Video Prediction. *arXiv* **2016**, arXiv:1605.07157.
15. Lu, Y.; Lu, C.; Tang, C.K. Online Video Object Detection Using Association LSTM. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2363–2371. [CrossRef]
16. Fan, H.; Yang, Y. PointRNN: Point Recurrent Neural Network for Moving Point Cloud Processing. *arXiv* **2019**, arXiv:1910.08287.
17. Schumann, O.; Wohler, C.; Hahn, M.; Dickmann, J. Comparison of Random Forest and Long Short-Term Memory Network Performances in Classification Tasks Using Radar. In Proceedings of the 2017 Symposium on Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 10–12 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6. [CrossRef]
18. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
19. Lombacher, J.; Hahn, M.; Dickmann, J.; Wohler, C. Potential of radar for static object classification using deep learning methods. In Proceedings of the 2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), San Diego, CA, USA, 19–20 May 2016; pp. 1–4. [CrossRef]
20. Schumann, O.; Hahn, M.; Dickmann, J.; Wohler, C. Semantic Segmentation on Radar Point Clouds. In Proceedings of the 2018 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 10–13 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 2179–2186. [CrossRef]
21. Schumann, O.; Lombacher, J.; Hahn, M.; Wohler, C.; Dickmann, J. Scene Understanding With Automotive Radar. *IEEE Trans. Intell. Veh.* **2020**, *5*, 188–203. [CrossRef]
22. Palffy, A.; Dong, J.; Kooij, J.F.P.; Gavrila, D.M. CNN based Road User Detection using the 3D Radar Cube. *IEEE Robot. Automat. Lett.* **2020**, *5*, 1263–1270. [CrossRef]
23. Danzer, A.; Griebel, T.; Bach, M.; Dietmayer, K. 2D Car Detection in Radar Data with PointNets. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019.
24. Chadwick, S.; Maddern, W.; Newman, P. Distant Vehicle Detection Using Radar and Vision. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
25. Nobis, F.; Geisslinger, M.; Weber, M.; Betz, J.; Lienkamp, M. A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection. In Proceedings of the Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 15–17 October 2019; pp. 1–7. [CrossRef]

26. Yang, B.; Guo, R.; Liang, M.; Casas, S.; Urtasun, R. RadarNet: Exploiting Radar for Robust Perception of Dynamic Objects. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
27. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015.
28. Fent, F. Machine Learning-Based Radar Point Cloud Segmentation. Master's Thesis, Technische Universität München, Munich, Germany, 2020.
29. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
30. Feng, D.; Haase-Schuetz, C.; Rosenbaum, L.; Hertlein, H.; Duffhauss, F.; Glaser, C.; Wiesbeck, W.; Dietmayer, K. Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1341–1360. [CrossRef]
31. Meyer, M.; Kuschk, G. Automotive Radar Dataset for Deep Learning Based 3D Object Detection. In Proceedings of the 2019 16th European Radar Conference (EuRAD), Paris, France, 2–4 October 2019; pp. 129–132.
32. Bijelic, M.; Gruber, T.; Mannan, F.; Kraus, F.; Ritter, W.; Dietmayer, K.; Heide, F. Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
33. Scheiner, N.; Schumann, O.; Kraus, F.; Appenrodt, N.; Dickmann, J.; Sick, B. Off-the-shelf sensor vs. experimental radar—How much resolution is necessary in automotive radar classification? In Proceedings of the 2020 IEEE 23rd International Conference on Information Fusion (FUSION), Rustenburg, South Africa, 6–9 July 2020.
34. Yamada, N.; Tanaka, Y.; Nishikawa, K. Radar cross section for pedestrian in 76GHz band. In Proceedings of the Microwave Conference, Paris, France, 4–6 October 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 4–1018. [CrossRef]
35. Yasugi, M.; Cao, Y.; Kobayashi, K.; Morita, T.; Kishigami, T.; Nakagawa, Y. 79GHz-band radar cross section measurement for pedestrian detection. In Proceedings of the Asia-Pacific Microwave Conference proceedings (APMC), Seoul, Korea, 5–8 November 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 576–578. [CrossRef]

4.2 Radar and Camera Fusion for Object Detection

The previous section has shown that the radar data quality of nuScenes does not enable general object detection based on this single modality. As the data source remains fixed, this section fuses camera and radar data to increase the input data density to enable object detection. The content of this section has been published in [231]. The developed 2D object detection network for radar and camera data is available on GitHub [235].

Summary
The fusion of camera and radar data in this paper is motivated by the complementary characteristics of both sensor data. While radar data provides an accurate distance measurement, camera data offers a high resolution measurement for the lateral positioning of objects and rich features for classification. This complementary characteristic comes with the downside that radar and camera data are measured in perpendicular planes of the 3D space. A spatial fusion of the two data types is therefore non-trivial. As camera-based object detection networks have proven effective in the past, the radar data are projected onto the camera image plane for the data fusion. In this way, a shared convolution operation can be performed on both data sources at the same time. As the radar data do not provide an elevation measurement, the data are assumed to originate from somewhere between the ground plane and a height of 3 m for the projection. The projected features of the radar data are concatenated as additional channels to the three channel RGB image. Figure 4.2 visualizes the principle and the resulting tensor of the fusion.

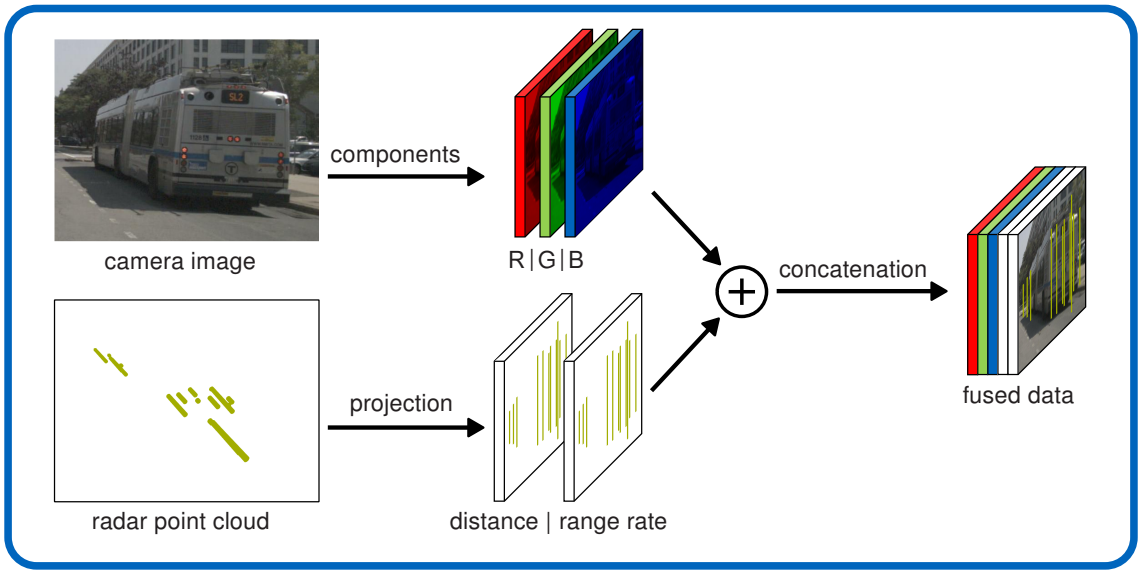


Figure 4.2: Fusion of radar point cloud and camera data in a DNN. The vertical dimension of the radar projection is enlarged to account for the uncertainty of the origin of the measurement in the elevation direction. Projected radar and camera channels are concatenated to fuse the data in the representation of an augmented image.

For the processing itself, a RetinaNet network architecture [112] is enhanced for data fusion. Each additional layer of a DNN creates more and more abstract information from its input data, until the network produces a result at the desired abstraction level at the output layer, in this case an object detection result. The radar sensor data directly provide important information about an object such as its position and range rate. The camera data provide low-level pixel values that require additional aggregation before it can be used to detect an object. This comparison suggests that the abstraction level of camera and radar data is different, but the difference in abstraction is not quantifiable. The proposed CRF-Net architecture therefore performs data

fusion at different depths of the processing network and lets the network learn a combined optimal fusion level by the adjustment of the respective weights in the optimization process.

The network input can be configured to use camera and radar input data for a fusion, or either modality on their own. For the network configuration with the radar-only input, no configuration was found to produce reasonable detection performance. The RetinaNet camera-only baseline reaches an 2D mAP score of 43.0 %, the combined early and feature fusion of image and radar data increases the 2D mAP score by about 4 % to 44.9 %. The paper further applies ground truth-based filters to investigate the performance potential of such a fusion method if noise-free radar data and radar-centric labels were available. For this case, the fusion performance surpasses the camera baseline by 30 %. While a noise-free radar remains a theoretical concept, the fusion performance shows promising results for future research with more accurate and dense radar data. Table 4.2 summarizes these results.

Table 4.2: 2D object detection results with different input data. The fusion of camera and radar data outperforms the camera baseline network.

Input data	2D mAP	Relative Difference
Camera	43.0%	0 %
Camera and radar (CRF-Net)	44.9%	4 %
Camera and noise-free radar	56.0%	30 %

The paper motivates further research in augmenting the network to perform an evaluation on data with severe weather conditions. Furthermore, an augmentation to a 3D object detection approach is motivated by adding additional output layers to the network. Such additional studies were performed for this thesis and selected results are discussed in Section 5.1.

Research Questions

This publication is related to the second sub research question. The developed combined early and feature fusion DNN increases the 2D mAP in comparison to a camera baseline for object detection. The proposed low-level fusion approach is determined as a suitable processing method for both input data. The answer to the research question with regard to the robustness and 3D capabilities of the approach are discussed with reference to additional studies in Section 5.1.

Contributions

Felix Nobis initiated the idea of this paper and contributed essentially to its conception and content. Maximilian Geisslinger and Markus Weber wrote their master theses in the research project and contributed to the conception, implementation and experimental results of this research. Johannes Betz revised the paper critically. Markus Lienkamp made an essential contribution to the conception of the research project. He revised the paper critically for important intellectual content. He gave final approval of the version to be published and agrees to all aspects of the work. As a guarantor, he accepts the responsibility for the overall integrity of the paper.

Imprint of the Paper

A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection

Felix Nobis*, Maximilian Geisslinger†, Markus Weber‡, Johannes Betz and Markus Lienkamp
Chair of Automotive Technology, Technical University of Munich
Munich, Germany
Email: *nobis@ftm.mw.tum.de, †maximilian.geisslinger@tum.de, ‡markus.weber@tum.de

***Abstract*—Object detection in camera images, using deep learning has been proven successfully in recent years. Rising detection rates and computationally efficient network structures are pushing this technique towards application in production vehicles. Nevertheless, the sensor quality of the camera is limited in severe weather conditions and through increased sensor noise in sparsely lit areas and at night. Our approach enhances current 2D object detection networks by fusing camera data and projected sparse radar data in the network layers. The proposed CameraRadarFusionNet (CRF-Net) automatically learns at which level the fusion of the sensor data is most beneficial for the detection result. Additionally, we introduce *BlackIn*, a training strategy inspired by Dropout, which focuses the learning on a specific sensor type. We show that the fusion network is able to outperform a state-of-the-art image-only network for two different datasets. The code for this research will be made available to the public at: https://github.com/TUMFTM/CameraRadarFusionNet**

***Index Terms*—Sensor Fusion, Object Detection, Deep Learning, Radar Processing, Autonomous Driving, Neural Networks, Neural Fusion, Raw Data Fusion, Low Level Fusion, Multi-modal Sensor Fusion**

I. Introduction

In recent years convolutional neural networks (CNN) have been established as the most accurate methods for performing object detection in camera images [1]. The visual representation of the environment in camera images is closely linked to human visual perception. As humans perceive the driving environment mainly via their visual sense, it is well motivated for autonomous vehicles to rely on a comparable representation. However, in adverse conditions like heavy rain or fog, the visibility is reduced, and safe driving might not be guaranteed. In addition, camera sensors get increasingly affected by noise in sparsely lit conditions. Compared to camera sensors, radar sensors are more robust to environment conditions such as lighting changes, rain and fog [2]. The camera can be rendered unusable through weather-induced occlusion e.g. if water droplets stick to the camera lens and block the view, as shown in Figure 1.

In this paper, we investigate the fusion of radar and camera sensor data with a neural network, in order to increase the object detection accuracy. The radar acquires information about the distance and the radial velocity of objects directly. It is able to locate objects in a two-dimensional plane parallel to the ground. In contrast to the camera, no height information can be obtained by the radar sensor. We develop a network architecture that deals with camera and radar sensor data jointly. The proposed method is able to detect objects more reliably in the nuScenes dataset [3] and the TUM dataset which is created for this research. Additionally, we show the limitations of our fusion network and directions for future development.

Fig. 1: Van occluded by a water droplet on the lens

Section II discusses related methods for object detection and sensor fusion. Section III describes our method to preprocess the radar data before fusing it into the network. We continue to describe the network architecture in Section IV. The evaluation and discussion of the approach is performed in Section V. Finally, our conclusions from the work are presented in Section VI.

II. Related Work

[4] were the first to successfully implement a convolutional neural network for the classification of images that outperformed the state-of-the-art in the ImageNet competition. This marked a starting point for increased interest in research into image processing with neural networks. Sebsequently,

neural network architectures for classification are augmented to perform additional tasks such as object detection [5] and semantic segmentation [6]. Several network meta-architectures for object detection exist, which build upon a variety of convolutional layer designs for feature extraction. In terms of real-time application, single shot architectures have been shown to perform accurately while keeping computational times reasonably low [7]. In recent years, new feature extraction architectures have been proposed which increase the object detection performance when employed in a given meta-architecture [8]–[11]. Recently, further studies emerged to automatically fine-tune an initial neural network design to increase the detection performance or minimize the run-time, without effecting the detection performance significantly [12], [13].

The success of neural networks for image data processing has led to an adaption to additional sensor principles and to sensor fusion. By incorporating multi-modal sensor data in the sensor fusion, researchers aim to obtain more reliable results for the different tasks involved in environmental perception for autonomous vehicles. [14] projects lidar data onto the 2D ground plane as a bird's-eye view and fuse it with camera data to perform 3D object detection. [15] projects the lidar onto the ground plane and onto a perpendicular image plane, and fuses both representations with the camera image in a neural network for object detection. [16] fuses lidar and camera data in a neural network to segment the driveable road. The paper proposes a network structure which consists of two branches for lidar and camera input. The interconnections of these branches are trainable so that the network can learn an optimized depth level in the network for the data fusion during the training process. [17] uses a similar fusion approach while operating with a bird's-eye view projection for both camera and lidar.

Convolutional neural networks are widely applied to operate on regular 2D grids (e.g. images) or 3D grids (e.g. voxels). The 3D lidar object detection approaches discussed above apply the idea of transforming unstructured lidar point clouds onto a regular grid before feeding it into a neural network. We employ the same process to the radar data.

[18] uses radar detections to create regions of interest in camera images, in order to classify objects in these regions using a simple neural network. A similar approach of using the radar to guide the object detection in the image space is performed in a series of other works [19]–[22]. [23] fuse independent detections of the camera and radar in order to associate the distance measurements of the radar with objects in the image space. [24] fuses independently tracked detections of each sensor to generate one final position estimation which incorporates the readings of both sensors. [25] present a deep learning approach with Generative Adversarial Networks (GANs) to fuse camera data and radar data, incorporated into a 2D bird's-eye view grid map, in order to perform free space detection.

[26] gives an overview of deep learning methods for sensor fusion. They conclude that raw level fusion methods for image and radar data have merely been investigated to date, and that more research needs to be conducted in this respect. [27] projects low level radar data onto a camera image plane perpendicular to the road, and proposes a neural network for the fusion with the camera image. They use the range and the range rate of the radar as additional image channels. The paper proposes two fusion strategies by concatenation or by element-wise addition on a fixed layer after initial separated layers for the sensors. They show the benefit of the fusion strategy for a self-recorded dataset.

In this paper, we use a similar projection approach to [27] to project the radar data onto the vertical plane of the camera image with which it is fused. We propose a fusion network that is able to learn the network depth at which the fusion is most beneficial to reduce the network loss. We operate in the image space to operate with 2D ground-truth data which significantly facilitates training data generation in comparison to 3D labels.

Due to the range rate measurement, moving objects can be distinguished from their surroundings in the radar data. For practical applications such as Adaptive Cruise Control (ACC), filtering for moving objects is applied to reduce the amount of false postives in the radar returns. At the same time, important stationary objects, e.g. cars stopped in front of a traffic lights, will be filtered out as well. In this approach no filtering for moving objects is performed, so that we are able to detect stationary and moving traffic objects alike.

III. Radar Data Preprocessing

This section describes the projection of the radar data to the image plane that is used in our fusion approach. We describe the spatial calibration of the camera and radar sensors, how to deal with the missing height information from the radar returns, how to deal with the sparsity of the radar data, and ground-truth filtering methods to reduce the noise or clutter in the radar data.

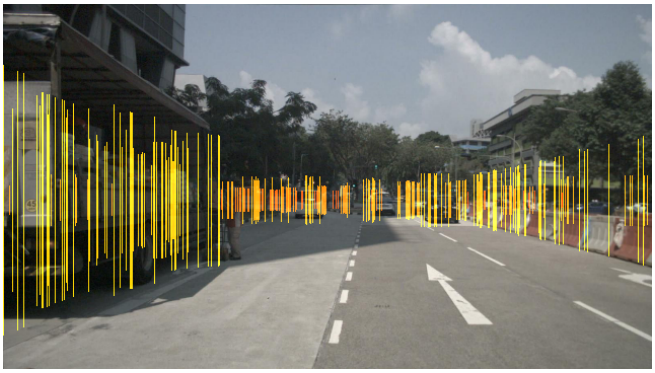
The radar sensor outputs a sparse 2D point cloud with associated radar characteristics. The data used for this work includes the azimuth angle, the distance and the radar cross section (RCS). We transform the radar data from the 2D ground plane to a perpendicular image plane. The characteristics of the radar return are stored as pixel values in the augmented image. At the location of image pixels where no radar returns are present, the projected radar channel values are set to the value 0. The input camera image consists of three channels (red, green, blue); to this we add the aforementioned radar channels as the input for the neural network. In our own dataset, the field of view (FOV) of three radars overlap with the FOV of the front-facing fish-eye camera. We concatenate the point clouds of the three sensors into one and use this as the projected radar input source. The projection differs, as the nuScenes dataset

uses a $70°$ FOV camera while the TUM dataset uses a $185°$ FOV fish-eye camera. In the nuScenes dataset, camera intrinsic and extrinsic mapping matrices are provided to transform a point from world coordinates into image coordinates. The non-linearities of a fish-eye lens cannot be mapped with a linear matrix operation. We use the calibration method presented by [28] to map the world coordinates to the image coordinates for our own data.

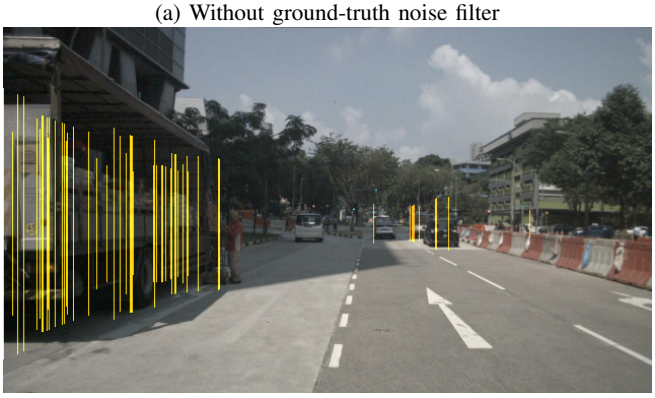
The radar detections give no information about the height at which they were received, which increases the difficulty to fuse the data types. The 3D coordinates of the radar detections are assumed to be returned from the ground plane that the vehicle is driving on. The projections are then extended in perpendicular direction to this plane, so as to account for the vertical extension of the objects to be detected. We detect traffic objects which can be classified as cars, trucks, motorcycles, bicycles and pedestrians. To cover the height of such object types, we assume a height extension of the radar detections of $3\,\mathrm{m}$ to associate camera pixels with radar data. The radar data is mapped with a pixel width of one into the image plane.

The camera data in the nuScenes dataset is captured at a resolution of $1600 \times 900 = 1,440,000$ pixels at an opening angle of $70°$ for the front camera. The lidar returns up to $14,000$ points for the same horizontal opening angle [29]. On a fraction of the nuScenes dataset (nuScenes mini), we calculated an average of 57 radar detections per cycle for the front radar. The greater variety in the density of the radar and the camera data - in comparison to the lidar and the camera - poses the challenge of finding a suitable way to fuse the data in one shared network structure. For our own dataset, we use the Continental ARS430 radar which has a different output format but comparable radar characteristics to the radar used in nuScenes. To deal with the sparsity of radar data, [25] uses probabilistic grid maps to generate continuous information from the radar. In this work, we increase the density of radar data by jointly fusing the last 13 radar cycles (around $1\,\mathrm{s}$) to our data format. Ego-motion is compensated for this projection method. Target-vehicle motion cannot be compensated. The fusion of previous time steps adds to the information density of the radar input. At the same time, it can also add noise to the input data as the detections of moving objects at previous time steps do not align with the current object position. This drawback is tolerated to obtain an information gain due to the additional data. Figure 2a shows the input data format for the neural network in an exemplary scene. The radar channels (distance and RCS) are mapped to the same locations and therefore shown in a uniform color.

The radar returns many detections coming from objects which are not relevant for the driving task, such as ghost objects, irrelevant objects and ground detections. These detections are called clutter or noise for the task at hand. In the evaluation, we compare the fusion of the raw noisy radar data with two additionally filtered approaches. First, in the nuScenes dataset,



(a) Without ground-truth noise filter



(b) With ground-truth noise filter

Fig. 2: nuScenes sample with radar projection to the image plane for the last 13 radar cycles. Radar channels are shown in yellow. The red color shift depicts increasing distances. Best viewed in color.

only a fraction of the labeled objects is detected by the radar. In training and evaluation, we therefore apply an annotation filter (AF), so that the filtered ground-truth data only contains objects which yield at least one radar detection. This is done via associating the 3D bounding boxes with radar points. The fusion approach is expected to show its potential for those objects which are detectable in both modalities. Second, we apply a ground-truth filter to the radar data which removes all radar detections outside of the 3D ground-truth bounding boxes. Of course, this step cannot be performed if applied to a real scenario. It is employed here to show the general feasibility of the fusion concept with less clutter in the input signal. The radar data after the application of the filter is shown in Figure 2b. Note, that the ground-truth radar filter (GRF) does not output perfect radar data and partly filters out relevant detections from the data for four reasons. First, we do not compensate the motion of other objects when we concatenate the past radar detections in the input. As the nuScenes dataset is labeled at $2\,\mathrm{Hz}$, no ground-truth is available for intermediate radar detection cycles, radar object detections only present in intermediate cycles are possibly filtered out. Second, slight spatial miscalibrations between the radar and camera sensors result in a misalignment of the radar detection locations and

the ground-truth bounding boxes at greater distances. Third, the data from the radar and the camera are not recorded at the exact same time. This leads to a spatial misalignment for moving objects. As we jointly operate on the last 13 detections of the radar, this effect is increased. Fourth, while the radar distance measurement is very reliable, its measurements are not perfect and slight inaccuracies can cause the detections to lie outside of the ground-truth bouding boxes. The unintended filtering of relevant data can partly be seen in Figure 2b. In Section V-C, we compare the results for the network using raw radar data and ground-truth filtered radar data. For the training and evaluation step, the 3D ground-truth bounding boxes are projected onto the 2D image plane.

IV. Network Fusion Architecture

Our neural network architecture builds on RetinaNet [30] as implemented in [31] with a VGG backbone [11]. The network is extended to deal with the additional radar channels of the augmented image. The output of the network is a 2D regression of bounding box coordinates and a classification score for the bounding box. The network is trained using focal loss, as proposed in [30]. Our baseline method uses a VGG feature extractor during the first convolutional layers.

The amount of information of one radar return is different from the information of a single pixel. The distance of an object to the ego-vehicle, as measured by the radar, can be considered more relevant to the driving task than a simple color value of a pixel of a camera. If both sensors are fused by concatenation in an early fusion, we should assume that the different data are semantically similar [32]. As we cannot strongly motivate this assumption, the fusion of the first layer of the network might not be optimal. In deeper layers of the neural network, the input data is compressed into a denser representation which ideally contains all the relevant input information. As it is hard to quantify the abstraction level of the information provided by each of the two sensor types, we design the network in a way that it learns itself at which depth level the fusion of the data is most beneficial to the overall loss minimization. The high-level structure of the network is shown in Figure 3. The main pipeline of the fusion network is shown in the center branch of the graph, composed of the VGG blocks. The camera and radar data is concatenated and fed into the network in the top row. This branch of the network is processed via the VGG layers for both the camera and radar data. In the left branch, the raw radar data is additionally fed into the network at deeper layers of the network at accordingly scaled input sizes through max-pooling. The radar data is concatenated to the output of the previous fused network layers of the main branch of the network. The Feature Pyramid Network (FPN) as introduced in [33] is represented by the blocks P3 through P7; herein, the radar channels are additionally fused by concatenation at each level. The outputs of the FPN blocks are finally processed by the bounding box regression and the classification blocks [30]. The optimizer implicitly teaches the network at which depth levels the radar data is fused with the greatest impact, by adapting the weights to the radar features at the different layers. A similar technique has been applied by [16].

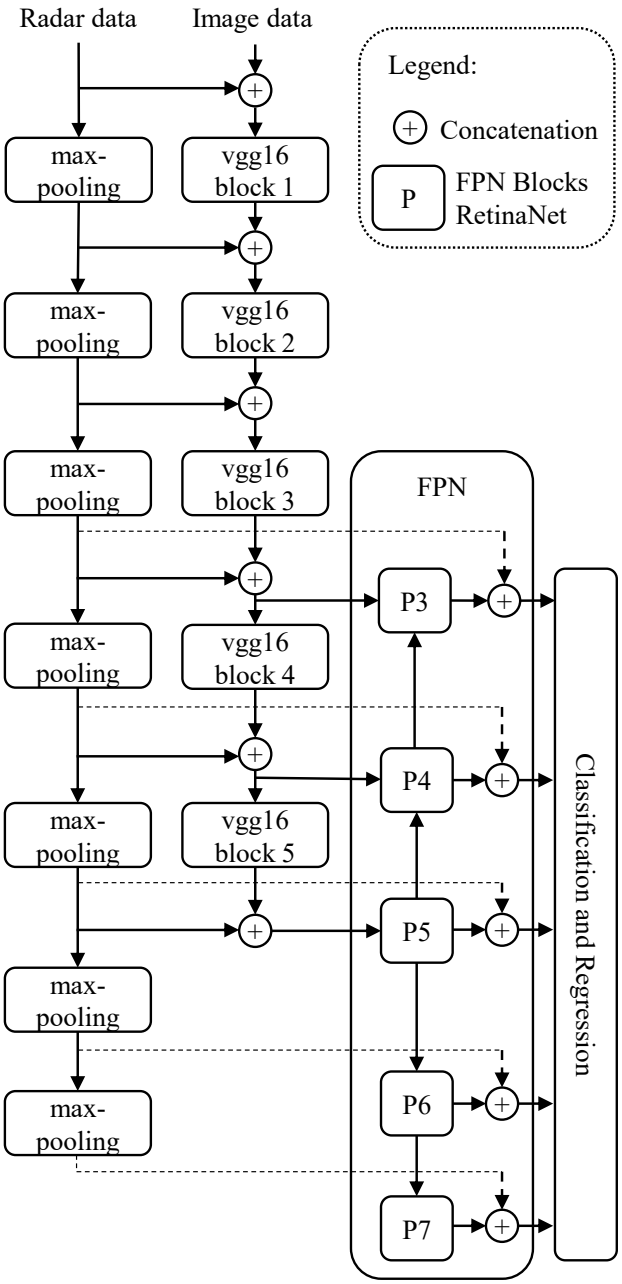


Fig. 3: High-level structure of CameraRadarFusionNet (CRF-Net)

We introduce a new training strategy to multi-modal sensor fusion for camera and radar data. The strategy is inspired by the technique Dropout [34]. Instead of single neurons, we

simultaneously deactivate *all* input neurons for the camera image data, for random training steps. This is done at a rate of 0.2 of all training images. We call this technique *BlackIn*. [35] introduced BlackOut which is inspired by dropout on the final layer of the network. The absence of camera input data pushes the network to rely more on the radar data. The goal is to teach the network the information value of the sparse radar data independently of the much denser camera representation. We begin the training with weights that are pretrained on images for the feature extractor. The training focus towards the radar, additionally intends to overcome this bias.

V. Experiments and Results

In this section, we evaluate the network on the nuScenes dataset and a second dataset collected during the work for this paper. We compare our CameraRadarFusionNet (CRF-Net) with the baseline network, which is our adapted implementation of RetinaNet [30].

A. Datasets

a) nuScenes dataset: The nuScenes dataset is extensively described in [3]. It is recorded in various locations and conditions in Boston and in Singapore. We condense the original 23 object classes into the classes shown in Table I for our detection evaluation. The nuScenes results are evaluated with and without the application of ground-truth filters.

TABLE I: Number of objects per object class in nuScenes and our dataset

Object classes	nuScenes	TUM
Car	22591	4020
Bus	1332	109
Motorcycle	729	10
Truck	5015	14
Trailer	1783	45
Bicycle	616	438
Human	10026	678

b) Our data (TUM): We utilize the same classes for evaluation as for the nuScenes dataset. Our dataset is annotated with 2D bounding boxes using the Computer Vision Annotation Tool (CVAT) [36]. As we lack 3D ground-truth data, no additional ground-truth filter can be applied to this dataset during the training and validation step. We reduce the default anchor sizes of RetinaNet by a factor two for our dataset, as the objects appear smaller on the fish-eye images.

B. Training

We create an 60:20:20 split from the raw data of nuScenes to balance the amount of day, rain and night scenes in the training, validation and test set. We use the nuScenes images at an input size of 360 x 640 pixels. The fish-eye images of our dataset are processed at 720 x 1280 pixel resolution. Objects generally appear smaller in the fish-eye images which we want to compensate with the augmentation of the resolution. We weight the object classes according to the number of appearances in the respective datasets for the mean Average Precision (mAP) calculation.

The weights of the VGG feature extractor are pretrained on the Imagenet dataset [37]. During preprocessing, the camera image channels are min-max scaled to the interval [-127.5,127.5], the radar channels remain unscaled. We perform data augmentation on our dataset because the amount of labeled data is relatively small. The number of objects per class for each dataset is shown in Table I.

Training and evaluation are performed with an Intel Xeon Silver 4112 CPU, 96GB RAM and a NVIDIA Titan XP GPU. On the nuScenes dataset, the networks are trained for 25 epochs and a batch size of 1 in a period of about 22 hours for the baseline network and about 24 hours for the CRF-Net. On our dataset, the networks are trained for 50 epochs and a batch size of 1 over a period of about 18 hours.

C. Evaluation

Table II shows the mean average precision for different configurations of our proposed network. The first block shows the results on the nuScenes dataset. The fusion network is achieving comparable but slightly higher detection results than the image network for the raw data inputs. The CRF-Net trained with BlackIn achieves a mAP of $0.35\,\%$-points more than without BlackIn. In the next step, we apply the annotation filter (AF) which considers only objects which are detected by at least one radar point. When the network additionally learns on ground-truth filtered radar data (AF, GRF), the mAP advantage of the CRF-Net rises to $12.96\,\%$-points compared to the image baseline (AF). The last line of the nuScenes block shows an additional comparison study. The radar channels are reduced to one channel which indicates solely the existence or non-existence of a radar detection in the image plane. The drop in the mAP score shows that the radar meta data, e.g. distance and RCS, are important for the detection result.

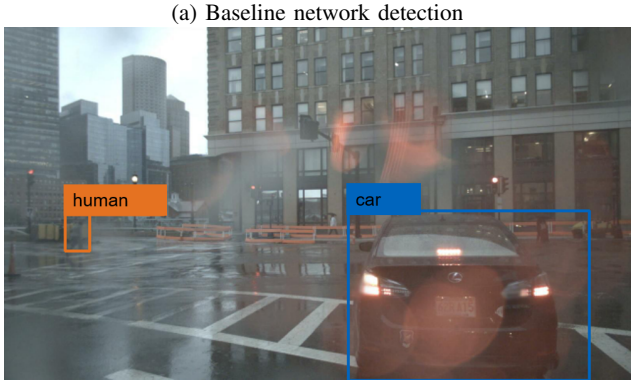
The second block of Table II shows the data for our own dataset. The performance gain of the fusion network compared to the baseline ($1.4\,\%$-points) is greater for our data than for nuScenes. This could be due to the use of three partly overlapping radars in our data and due to the use of a more advanced radar sensor. In addition, we labeled objects that appear small in the images in our dataset; in the nuScenes dataset, objects at a distance greater than $80\,\mathrm{m}$ are mostly not labeled. As suggested in [27], it is possible that the radar is beneficial especially for objects at a greater distance from the ego vehicle. The camera data differs in both datasets due to the different lens characteristics and the different input resolutions, so that a definite reason cannot be given here.

Figure 4 qualitatively illustrates the superiority of the object detection with the CRF-Net for an example scene.

TABLE II: mAP scores of the baseline network and our CameraRadarFusionNet. Configurations: (AF) - Annotation filter, (GRF) - ground-truth radar filter, (NRM) - No radar meta data

Data	Network	mAP
nuScenes	Baseline image network	43.47 %
	CRF-Net w/o BlackIn	43.6 %
	CRF-Net	**43.95** %
	Baseline image network (AF)	43.03 %
	CRF-Net (AF)	44.85 %
	CRF-Net (AF, GRF)	**55.99 %**
	CRF-Net (AF, GRF, NRM)	53.23 %
TUM	Baseline image network	56.12 %
	CRF-Net	**57.50** %

(a) Baseline network detection



(b) CRF-Net detection

Fig. 4: Detection comparison of the baseline network (a) and the CRF-Net (b). The baseline network does not detect the pedestrian on the left.

The overall higher mAPs for the fusion network compared to the baseline presented in Table II show the potential of the fusion approach. This potential motivates further research towards an ideal network architecture for this type of fusion. The performance gain for ground-truth filtered radar data motivates the development of a non-ground-truth based filtering method for the radar data during preprocessing, or inside the neural network. In future work, we will continue research into filtering out noisy radar detections before feeding them into the fusion network, to improve the results for the application under real-world conditions.

The baseline network needs 33 ms for the processing of one image at a size of 360 x 640 pixels. The CRF-Net needs 43 ms for the processing of the corresponding fused data. Additionally the data processing for the radar projection and channel generation amounts to 56 ms of CPU time. The time needed for the processing of the ground-truth filters is negligible. In our TUM dataset, we input the data at a higher resolution, which results in increased execution times. The baseline network processing takes 92 ms, the CRF-Net needs 103 ms, the data generation takes 333 ms. In this dataset more radar data is used and the projection is done with a fish-eye projection method which adds to the data generation time. However, the data generation is not optimized and the values are given as a reference to present the current status of the implementation.

VI. Conclusions and Outlook

This paper proposes the CameraRadarFusion-Net (CRF-Net) architecture to fuse camera and radar sensor data of road vehicles. The research adapts ideas from lidar and camera data processing and shows a new direction for fusion with radar data. Difficulties and solutions to process the radar data are discussed. The BlackIn training strategy is introduced for the fusion of radar and camera data. We show that the fusion of radar and camera data in a neural network can augment the detection score of a state-of-the-art object detection network. This paper lends justification to a variety of areas for further research. As neural fusion for radar and camera data has only recently been studied in literature, finding optimized network architectures needs to be explored further.

In the future, we plan research to design network layers to process the radar data prior to the fusion, so as to filter out noise in the radar data. The fusion with additional sensor modalities such as lidar data could further increase the detection accuracy, while at the same time adding complexity by augmenting the layers or through the need to introduce new design concepts. The study of the robustness of neural fusion approaches against spatial and temporal miscalibration of the sensors needs to be evaluated. We see an increased potential for multi-modal neural fusion for driving in adverse weather conditions. Additional datasets modeling these conditions need to be created to study this assumption. Lastly, as the radar sensor introduces distance information into the detection scheme, the applicability of the fusion concept to 3D object detection is a direction we want to explore.

On the hardware side, high-resolution or imaging radars [38] are expected to increase the information density of radar data and reduce the amount of clutter. The hardware advancement is expected to enable an increase in the detection results of our approach.

Contributions and Acknowledgments

Felix Nobis initiated the idea of this paper and contributed essentially to its conception and content. Maximilian Geisslinger and Markus Weber wrote their master theses in the research project and contributed to the conception, implementation and experimental results of this research. Johannes Betz revised the paper critically. Markus Lienkamp made an essential contribution to the conception of the research project. He revised the paper critically for important intellectual content. He gave final approval of the version to be published and agrees to all aspects of the work. As a guarantor, he accepts the responsibility for the overall integrity of the paper. We express gratitude to Continental Engineering Service for funding for the underlying research project and for providing the sensor hardware and guidance for this research.

References

[1] T.-Y. Lin, G. Patterson, M. R. Ronchi, and Y. Cui, "Common objects in context," 2019. [Online]. Available: http://cocodataset.org/#detection-leaderboard

[2] F. de Ponte Müller, "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles," *Sensors (Basel, Switzerland)*, vol. 17, no. 2, 2017.

[3] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[4] 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 - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc, 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[5] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors." [Online]. Available: http://arxiv.org/pdf/1611.10012v3

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: single shot multibox detector," *European Conference on Computer Vision*, vol. 9905, pp. 21–37, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *29th IEEE Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2016, pp. 770–778.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc, 2015, pp. 91–99.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Seventh International Conference on Learning Representations*, 2015. [Online]. Available: https://arxiv.org/pdf/1409.1556.pdf

[12] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," *ECCV'18*, 2018.

[13] M. Tan and Q. Le V, "Efficientnet: Rethinking model scaling for convolutional neural networks," *International Conference on Machine Learning*, 2019.

[14] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," *IROS*, 2018.

[15] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving."

[16] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde, "Lidar-camera fusion for road detection using fully convolutional neural networks," *Robotics and Autonomous Systems*, no. 111, pp. 125–131, 2018.

[17] D. Yu, H. Xiong, Q. Xu, J. Wang, and K. Li, "Multi-stage residual fusion network for lidar-camera road detection," *IEEE Intelligent Vehicles Symposium*, 2019.

[18] Z. Ji and D. V. Prokhorov, "Radar-vision fusion for object classification," *11th International Conference on Information Fusion*, 2008.

[19] J. Kocic, N. Jovicic, and V. Drndarevic, "Sensors and sensor fusion in autonomous vehicles," in *TELFOR 2018*. Belgrade: Telecommunications Society and Academic Mind, 2018, pp. 420–425.

[20] S. Han, X. Wang, L. Xu, H. Sun, and N. Zheng, "Frontal object perception for intelligent vehicles based on radar and camera fusion," in *Proceedings of the 35th Chinese Control Conference*, J. Chen and Q. Zhao, Eds. Piscataway, NJ: IEEE, 2016, pp. 4003–4008.

[21] X. Zhang, M. Zhou, P. Qiu, Y. Huang, and J. Li, "Radar and vision fusion for the real-time obstacle detection and identification," *Industrial Robot: the international journal of robotics research and application*, vol. 2007, no. 2, p. 233, 2019.

[22] S. Zeng, W. Zhang, and B. B. Litkouhi, "Fusion of obstacle detection using radar and camera," Patent US 9.429,650 B2, 2016.

[23] H. Jha, V. Lodhi, and D. Chakravarty, "Object detection and identification using vision and radar data fusion system for ground-based navigation," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2019, pp. 590–593.

[24] K.-E. Kim, C.-L. Lee, D.-S. Pac, and M.-T. Lim, *Sensor Fusion for Vehicle Tracking with Camera and Radar Sensor: 2017 17th International Conference on Control, Automation and Systems : proceedings : October 18-21, 2017, Ramada Plaza, Jeju, Korea*. Piscataway, NJ: IEEE, 2017.

[25] V. Lekic and Z. Babic, "Automotive radar and camera fusion using generative adversarial networks," *Computer Vision and Image Understanding*, 2019. [Online]. Available: https://www-sciencedirect-com.eaccess.ub.tum.de/science/article/pii/S1077314219300530?via%3Dihub

[26] Di Feng, C. Haase-Schuetz, L. Rosenbaum, H. Hertlein, F. Duffhauss, C. Glaser, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *CoRR*, vol. abs/1902.07830, 2019.

[27] S. Chadwick, W. Maddern, and P. Newman, "Distant vehicle detection using radar and vision," *CoRR*, vol. abs/1901.10951, 2019.

[28] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006*. Piscataway, NJ: IEEE Operations Center, 2006, pp. 5695–5701.

[29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[31] Fizyr, "Keras retinanet," 2015. [Online]. Available: https://github.com/fizyr/keras-retinanet

[32] K. Liu, Y. Li, N. Xu, and P. Natarajan, "Learn to combine modalities in multimodal deep learning," *arXiv e-prints*, p. arXiv:1805.11730, 2018.

[33] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[35] S. Ji, S. V. N. Vishwanathan, N. Satish, M. J. Anderson, and P. Dubey, "Blackout: Speeding up recurrent neural network language models with very large vocabularies."

[36] Intel, "Computer vision annotation tool (cvat)," 2018. [Online]. Available: https://github.com/opencv/cvat

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR09*, 2009.

[38] S. Brisken, F. Ruf, and F. Höhne, "Recent evolution of automotive imaging radar and its information content," *IET Radar, Sonar & Navigation*, vol. 12, no. 10, pp. 1078–1081, 2018.

4.3 Radar, Camera, and Lidar Fusion for Object Detection

The previous section presented a fusion of camera and radar data. As state of the art 3D mAP scores for object detection are achieved with lidar input data, this section performs a fusion of all three data modalities to increase the achievable performance of 3D object detection. The content of this section has been published in [232]. The developed 3D object detection network for radar, camera, and lidar data is available on GitHub [236].

Summary
This paper aims to set a baseline for further fusion developments by adapting the effective lidar VoxelNet architecture [154] to use multi-modal sensor data input. The sensor data are fused in the input layer in an early fusion approach. The input space of the radar and lidar points are discretized to a 3D grid. Camera pixel information is projected onto the lidar points. In this way, a depth estimation from the image data itself becomes obsolete for a fusion in 3D. Radar, camera, and lidar features are concatenated at the respective input grid points. The joint input tensor is processed with sparse convolutions to enable efficient processing of the 3D voxel grid. The resulting network is named Radar Voxel Fusion-Net (RVF-Net).

In this study only car objects are detected with the network to reduce the output complexity and as this category is the most frequent object of interest in the used data set and potentially real-world scenarios. Example detections which show the benefit of the addition of radar data are visualized in Figure 4.3.

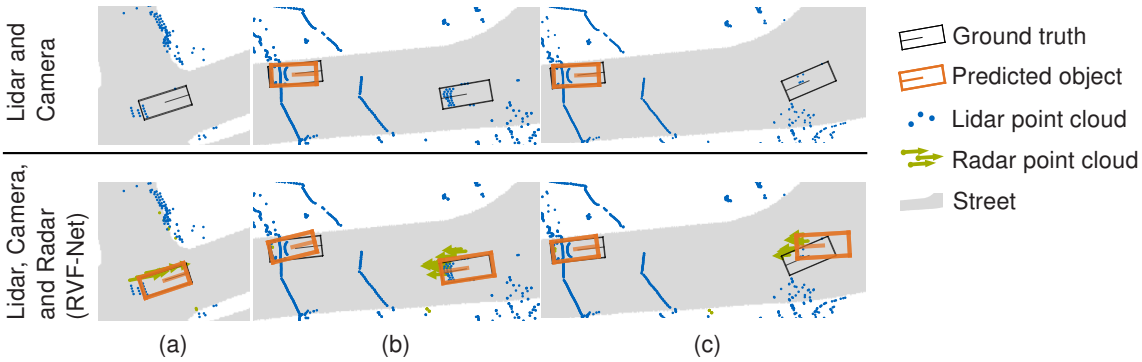


Figure 4.3: Object detection result comparison using different input data, visualized from a BEV. The fusion of radar data enhances the detection performance. (a) Only RVF-Net is able to detect the vehicle from the point cloud. (b) RVF-Net detects both bounding boxes. (c) RVF-Net detects both bounding boxes. However, the detected box on the right has a high translation and rotation error towards the ground truth bounding box. Adapted from [232].

The fusion approach increases the object detection performance to the lidar baseline by about 5 % with a 3D AP score of 54.9 %. The 3D fusion is especially effective for samples with rain or night conditions where it outperforms the lidar baseline by 10 % and 6 %, respectively. The absolute 3D AP for night scenes of the RVF-Net is 67.4 % which is significantly higher than the score for the overall data set. The absolute AP for rain scenes is 48.3 %. The difference in absolute AP scores for the different data sets shows the strong data dependence of deep learning methods.

The network is evaluated against a high-level fusion approach. For this, lidar and projected camera data are used as a shared input in the proposed network architecture, with re-trained network parameters, to produce the first group of object proposals to the high-level fusion pipeline. The radar data are processed with the same DNN architecture, with radar specific

re-trained network parameters, to produce the second group of object proposals independently. Both, camera and lidar data are used in the first pipeline to enable an overall comparison to the low-level fusion approach of all three modalities in RVF-Net, as no camera-only processing in the 3D space is possible with the proposed approach. Both groups of object proposals are fused with a Kalman filter which is parameterized using three randomly selected scenes of the nuScenes data set. As the tracking performed by the Kalman filter for the high-level fusion has an impact on the resulting object detection performance, the low-level fusion detections are also tracked with the same filter to enable a comparison. The tracked low-level fusion reaches an 3D AP score of 47.1 % which is relatively 41 % higher than that of the high-level fusion. The resulting high-level fusion performance is deteriorated through the use of the radar data in the fusion. A pipeline using only the camera and lidar input data reaches an AP score which is 20 % higher than that of the high-level fusion. The independent object detection results of the weak radar features are not precise enough to contribute positively to the detection result. The joint processing in the low-level fusion approach, however, can leverage the weak information in the radar data in conjunction with further data modalities to improve the detection result. A summary of the results is given in Table 4.3.

Table 4.3: 3D object detection and tracking results. The low-level fusion outperforms the lidar and high-level fusion approaches.

Network	3D mAP	Relative Difference
High-level fusion	33.3 %	0 %
Lidar	40.0 %	20 %
Low-level fusion (RVF-Net)	47.1 %	41 %

The paper furthermore introduces a combined classification and regression loss for the optimization of the bounding box orientation estimation. The novel loss increases the 3D AP by 4 %, while the Average Orientation Error (AOE) is reduced by about 40 % in comparison to the same network architecture with a standard regression loss. The proposed loss is similar to the one used in SECOND [151], with the enhancement that it solves an edge case where two distinct bounding box configurations would generate the same loss value in their formulation.

When projecting camera pixels to lidar data, camera data not associated with lidar locations get lost. To mitigate this problem the lidar point cloud is depth-completed with the IP-Basic algorithm [96] prior to the camera data fusion in an ablation study. The addition of this data, however, did not lead to an increased performance.

A combined IoU- and distance-based matching threshold is used for the proposed network. In contrast to results in the literature, the combined matching threshold did not result in a higher object detection score in comparison to the commonly used IoU-threshold.

Potential for further improvements of the network are discussed in the paper. The classification loss overfits before the bounding box regression loss reaches its minimum. A further tuning of the network training process to balance the results could improve the network performance. Furthermore, the field of view of the fusion could be extended to enable a full 360° FOV for the object detection which is required to participate in the official nuScenes evaluation.

Research Questions

This publication is related to the second sub research question. The developed low-level fusion DNN increases the 3D mAP in comparison to a high-level fusion baseline for object detection. The proposed low-level fusion approach is determined as a suitable processing method for both input data. The answer to the research question will be discussed in more detail in Section 5.1.

Contributions

Felix Nobis (F.N.), as the first author, initiated the idea of this paper and contributed essentially to its concept and content. Conceptualization, F.N.; methodology, F.N. and Ehsan Shafiei (E.S.); software, E.S., F.N. and Phillip Karle (P.K.).; data curation, E.S. and F.N.; writing—original draft preparation, F.N.; writing—review and editing, E.S., P.K., Johannes Betz (J.B.) and Markus Lienkamp (M.L.); visualization, F.N. and E.S.; project administration, J.B. and M.L. All authors have read and agreed to the published version of the manuscript.

Imprint of the Paper

applied sciences

MDPI

Article

Radar Voxel Fusion for 3D Object Detection

Felix Nobis [1,*], Ehsan Shafiei [1], Phillip Karle [1], Johannes Betz [2] and Markus Lienkamp [1]

[1] Institute of Automotive Technology, Technical University of Munich, 85748 Garching, Germany; ehsan.shafiei@tum.de (E.S.); karle@ftm.mw.tum.de (P.K.); lienkamp@ftm.mw.tum.de (M.L.)
[2] mLab:Real-Time and Embedded Systems Lab, University of Pennsylvania, Philadelphia, PA 19104, USA; joebetz@seas.upenn.edu
* Correspondence: nobis@ftm.mw.tum.de

Abstract: Automotive traffic scenes are complex due to the variety of possible scenarios, objects, and weather conditions that need to be handled. In contrast to more constrained environments, such as automated underground trains, automotive perception systems cannot be tailored to a narrow field of specific tasks but must handle an ever-changing environment with unforeseen events. As currently no single sensor is able to reliably perceive all relevant activity in the surroundings, sensor data fusion is applied to perceive as much information as possible. Data fusion of different sensors and sensor modalities on a low abstraction level enables the compensation of sensor weaknesses and misdetections among the sensors before the information-rich sensor data are compressed and thereby information is lost after a sensor-individual object detection. This paper develops a low-level sensor fusion network for 3D object detection, which fuses lidar, camera, and radar data. The fusion network is trained and evaluated on the nuScenes data set. On the test set, fusion of radar data increases the resulting AP (Average Precision) detection score by about 5.1% in comparison to the baseline lidar network. The radar sensor fusion proves especially beneficial in inclement conditions such as rain and night scenes. Fusing additional camera data contributes positively only in conjunction with the radar fusion, which shows that interdependencies of the sensors are important for the detection result. Additionally, the paper proposes a novel loss to handle the discontinuity of a simple yaw representation for object detection. Our updated loss increases the detection and orientation estimation performance for all sensor input configurations. The code for this research has been made available on GitHub.

Keywords: perception; deep learning; sensor fusion; radar point cloud; object detection; sensor; camera; radar; lidar

Citation: Nobis, F.; Shafiei, E.; Karle P.; Betz, J.; Lienkamp, M. Radar Voxel Fusion for 3D Object Detection. *Appl. Sci.* **2021**, *11*, 5598. https://doi.org/10.3390/app11125598

Academic Editor: Chris G. Tzanis

Received: 29 April 2021
Accepted: 11 June 2021
Published: 17 June 2021

1. Introduction

In the current state of the art, researchers focus on 3D object detection in the field of perception. Three-dimensional object detection is most reliably performed with lidar sensor data [1–3] as its higher resolution—when compared to radar sensors—and direct depth measurement—when compared to camera sensors—provide the most relevant features for object detection algorithms. However, for redundancy and safety reasons in autonomous driving applications, additional sensor modalities are required because lidar sensors cannot detect all relevant objects at all times. Cameras are well-understood, cheap and reliable sensors for applications such as traffic-sign recognition. Despite their high resolution, their capabilities for 3D perception are limited as only 2D information is provided by the sensor. Furthermore, the sensor data quality deteriorates strongly in bad weather conditions such as snow or heavy rain. Radar sensors are least affected by inclement weather, e.g., fog, and are therefore a vital asset to make autonomous driving more reliable. However, due to their low resolution and clutter noise for static vehicles, current radar sensors cannot perform general object detection without the addition of further modalities. This work therefore combines the advantages of camera, lidar, and radar sensor modalities to produce an improved detection result.

Several strategies exist to fuse the information of different sensors. These systems can be categorized as early fusion if all input data are first combined and then processed, or late fusion if all data is first processed independently and the output of the data-specific algorithms are fused after the processing. Partly independent and joint processing is called middle or feature fusion.

Late fusion schemes based on a Bayes filter, e.g., the Unscented Kalman Filter (UKF) [4], in combination with a matching algorithm for object tracking, are the current state of the art, due to their simplicity and their effectiveness during operation in constrained environments and good weather.

Early and feature fusion networks possess the advantage of using all available sensor information at once and are therefore able to learn from interdependencies of the sensor data and compensate imperfect sensor data for a robust detection result similar to gradient boosting [5].

This paper presents an approach to fuse the sensors in an early fusion scheme. Similar to Wang et al. [6], we color the lidar point cloud with camera RGB information. These colored lidar points are then fused with the radar points and their radar cross-section (RCS) and velocity features. The network processes the points jointly in a voxel structure and outputs the predicted bounding boxes. The paper evaluates several parameterizations and presents the RadarVoxelFusionNet (RVF-Net), which proved most reliable in our studies.

The contribution of the paper is threefold:

- The paper develops an early fusion network for radar, lidar, and camera data for 3D object detection. The network outperforms the lidar baseline and a Kalman Filter late fusion approach.
- The paper provides a novel loss function to replace the simple discontinuous yaw parameterization during network training.
- The code for this research has been released to the public to make it adaptable to further use cases.

Section 2 discusses related work for object detection and sensor fusion networks. The proposed model is described in Section 3. The results are shown in Section 4 and discussed in Section 5. Section 6 presents our conclusions from the work.

2. Related Work

Firstly, this section gives a short overview of the state of the art of lidar object detection for autonomous driving. Secondly, a more detailed review of fusion methods for object detection is given. We refer to [7] for a more detailed overview of radar object detection methods.

2.1. 3D Lidar Object Detection

The seminal work of Qi et al. [8] introduces a method to directly process sparse, irregular point cloud data with neural networks for semantic segmentation tasks. Their continued work [9] uses a similar backbone to perform 3D object detection from point cloud frustums. Their so-called pointnet backbone has been adapted in numerous works to advance lidar object detection.

VoxelNet [10] processes lidar points in a voxel grid structure. The network aggregates a feature for each voxel from the associated points. These voxel grid cells are processed in a convolutional fashion to generate object detection results with an anchor-based region proposal network (RPN) [11].

The network that achieves the highest object detection score [3] on the KITTI 3D benchmark [2] uses both a voxel-based and pointnet-based processing to create their detection results. The processing of the voxel data is performed with submanifold sparse convolutions as introduced in [12,13]. The advantage of these sparse implementation of convolutions lies in the fact that they do not process empty parts of the grid that contain no information. This is especially advantageous for point cloud processing, as most of the 3D space does not contain any sensor returns. The network that achieves the highest object

detection score on the nuScenes data set [1] is a lidar-only approach as well [14]. Similarly, it uses a sparse VoxelNet backbone with a second stage for bounding box refinement.

2.2. 2D Sensor Fusion for Object Detection

This section reviews 2D fusion methods. The focus is on methods that fuse radar data as part of the input data.

Chadwick [15] is the first to use a neural network to fuse low level radar and camera for 2D object detection. The network fuses the data on a feature level after projecting radar data to the 2D image plane. The object detection scores of the fusion are higher than the ones of a camera-only network, especially for distant objects.

CRF-Net [16] develops a similar fusion approach. As an automotive radar does not measure any height information, the network assumes an extended height of the radar returns to account for the uncertainty in the radar returns origin. The approach shows a slight increase in object detection performance both on a private and the public nuScenes data set [1]. The paper shows further potential for the fusion scheme once less noisy radar data are available.

YOdar [17] uses a similar projection fusion method. The approach creates two detection probabilities of separate radar and image processing pipelines and generates their final detection output by gradient boosting.

2.3. 3D Sensor Fusion for Object Detection

This section reviews 3D fusion methods. The focus is on methods that fuse radar data as part of the input data.

2.3.1. Camera Radar Fusion

For 3D object detection, the authors of [18] propose GRIF-Net to fuse radar and camera data. After individual processing, the feature fusion is performed by a gated region of interest fusion (GRIF). In contrast to concatenation or addition as the fusion operation, the weight for each sensor in the fusion is learned in the GRIF module. The camera and radar fusion method outperforms the radar baseline by a great margin on the nuScenes data set.

The CenterFusion architecture [19] first detects objects in the 3D space via image-based object detection. Radar points inside a frustum around these detections are fused by concatenation to the image features. The radar features are extended to pillars similar to [16] in the 2D case. The object detection head operates on these joint features to refine the detection accuracy. The mean Average Precision (mAP) score of the detection output increases by 4% for the camera radar fusion compared to their baseline on the nuScenes validation data set.

While the methods above operate with point cloud-based input data, Lim [20] fuses azimuth range images and camera images. The camera data are projected to a bird's-eye view (BEV) with an Inverse Projection Mapping (IPM). The individually processed branches are concatenated to generate the object detection results. The fusion approach achieves a higher detection score than the individual modalities. The IPM limits the detection range to close objects and an assumed flat road surface.

Kim [21] similarly fuses radar azimuth-range images with camera images. The data are fused after initial individual processing, and the detection output is generated adopting the detection head of [22]. The fusion approach outperforms both their image and radar baselines on their private data set. Their RPN uses a distance threshold in contrast to standard Intersection over Union (IoU) matching for anchor association. The paper argues that the IoU metric prefers to associate distant bounding boxes over closer bounding boxes under certain conditions. Using a distance threshold instead increases the resulting AP by 4–5 points over the IoU threshold matching.

The overall detection accuracy of camera radar fusion networks is significantly lower than that of lidar-based detection methods.

2.3.2. Lidar Camera Fusion

MV3D [23] projects lidar data both to a BEV perspective and the camera perspective. The lidar representations are fused with the camera input after some initial processing in a feature fusion scheme.

AVOD [22] uses a BEV projection of the lidar data and camera data as their input data. The detection results are calculated with an anchor grid and an RPN as a detection head.

PointPainting [24] first calculates a semantic segmentation mask for an input image. The detected classes are then projected onto the lidar point cloud via a color-coding for the different classes. The work expands several lidar 3D object detection networks and shows that enriching the lidar data with class information augments the detection score.

2.3.3. Lidar Radar Fusion

RadarNet [25] fuses radar and lidar point clouds for object detection. The point clouds are transformed into a grid representation and then concatenated. After this feature fusion, the data are processed jointly to propose an object detection. An additional late fusion of radar features is performed to predict a velocity estimate separate to the object detection task.

2.3.4. Lidar Radar Camera Fusion

Wang [6] projects RGB values of camera images directly onto the lidar point cloud. This early fusion camera-lidar point cloud is used to create object detection outputs in a pointnet architecture. Parallel to the object detection, the radar point cloud is processed to predict velocity estimates of the input point cloud. The velocity estimates are then associated with the final detection output. The paper experimented with concatenating different amounts of past data sweeps for the radar network. Concatenating six consecutive time steps of the radar data for a single processing shows the best results in their study. The addition of the radar data increases their baseline detection score slightly on the public nuScenes data set.

3. Methodology

In the following, we list the main conclusions from the state of the art for our work:

- *Input representation*: The input representation of the sensor data dictates which subsequent processing techniques can be applied. Pointnet-based methods are beneficial when dealing with sparse unordered point cloud data. For more dense—but still sparse—point clouds, such as the fusion of several lidar or radar sensors, sparse voxel grid structures achieve more favorable results in the object detection literature. Therefore, we adopt a voxel-based input structure for our data. As many of the voxels remain empty in the 3D grid, we apply sparse convolutional operations [12] for greater efficiency.
- *Distance Threshold*: Anchor-based detection heads predominately use an IoU-based matching algorithm to identify positive anchors. However, Kim [21] has shown that this choice might lead to association of distant anchors for certain bounding box configurations. We argue that both IoU- and distance-based matching thresholds should be considered to facilitate the learning process. The distance-based threshold alone might not be a good metric when considering rotated bounding boxes with a small overlapping area. Our network therefore considers both thresholds to match the anchor boxes.
- *Fusion Level*: The data from different sensors and modalities can be fused at different abstraction levels. Recently, a rising number of papers perform early or feature fusion to be able to facilitate all available data for object detection simultaneously. Nonetheless, the state of the art in object detection is still achieved by considering only lidar data. Due to its resolution and precision advantage from a hardware perspective, software processing methods cannot compensate for the missing information in the input data of the additional sensors. Still, there are use cases where the lidar sensor alone is not sufficient. Inclement weather, such as fog, decreases the lidar

and camera data quality [26] significantly. The radar data, however, is only slightly affected by the change in environmental conditions. Furthermore, interference effects of different lidar modules might decrease the detection performance under certain conditions [27,28]. A drawback of early fusion algorithms is that temporal synchronized data recording for all sensors needs to be available. However, none of the publicly available data sets provide such data for all three sensor modalities. The authors of [7] discuss the publicly available data quality for radar sensors in more detail. Despite the lack of synchronized data, this study uses an early fusion scheme, as in similar works, spatio-temporal synchronization errors are treated as noise and compensated during the learning process of the fusion network. In contrast to recent papers, where some initial processing is applied before fusing the data, we present a direct early fusion to enable the network to learn optimal combined features for the input data. The early fusion can make use of the complementary sensor information provided by radar, camera and lidar sensors—before any data compression by sensor-individual processing is performed.

3.1. Input Data

The input data to the network consists of the lidar data with its three spatial coordinates x, y, z, and intensity value i. Similar to [6], colorization from projected camera images is added to the lidar data with r, g, b features. Additionally, the radar data contributes its spatial coordinates, intensity value RCS—and the radial velocity with its Cartesian components v_x and v_y. Local offsets for the points in the voxels dx, dy, dz complete the input space. The raw data are fused and processed jointly by the network itself. Due to the early fusion of the input data, any lidar network can easily be adapted to our fusion approach by adjusting the input dimensions.

3.2. Network Architecture

This paper proposes the RadarVoxelFusionNet (RVF-Net) whose architecture is based on VoxelNet [10] due to its empirically proven performance and straightforward network architecture. While other architectures in the state of the art provide higher detection scores, the application to a non-overengineered network from the literature is preferable for investigating the effect of a new data fusion method. Recently, A. Ng [29] proposed a shift from model-centric to data-centric approaches for machine learning development.

An overview of the network architecture is shown in Figure 1. The input point cloud is partitioned into a 3D voxel grid. Non-empty voxel cells are used as the input data to the network. The data are split into the features of the input points and the corresponding coordinates. The input features are processed by voxel feature encoding (VFE) layers composed of fully connected and max-pooling operations for the points inside each voxel. The pooling is used to aggregate one single feature per voxel. In the global feature generation, the voxel features are processed by sparse 3D submanifold convolutions to efficiently handle the sparse voxel grid input. The z dimension is merged with the feature dimension to create a sparse feature tensor in the form of a 2D grid. The sparse tensor is converted to a dense 2D grid and processed with standard 2D convolutions to generate features in a BEV representation. These features are the basis for the detection output heads.

The detection head consists of three parts: The classification head, which outputs a class score for each anchor box; the regression head with seven regression values for the bounding box position (x, y, z), dimensions (w, l, h) and the yaw angle e_θ; the direction head, which outputs a complementary classification value for the yaw angle estimation c_{dir}. For more details on the network architecture, we refer to the work of [10] and our open source implementation. The next section focuses on our proposed yaw loss, which is conceptually different from the original VoxelNet implementation.

Appl. Sci. **2021**, *11*, 5598

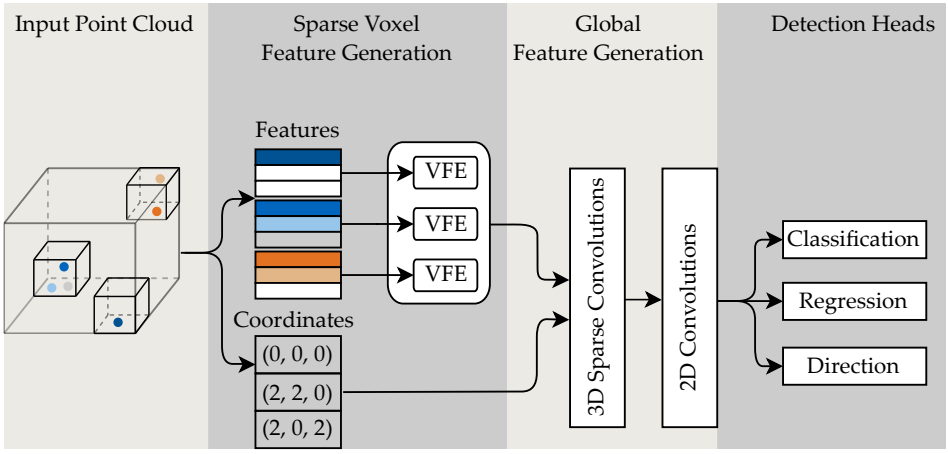
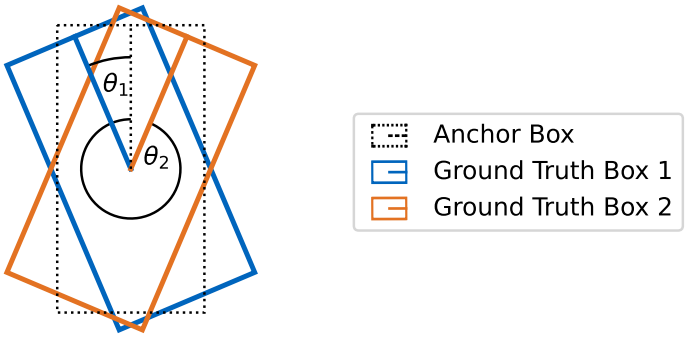


Figure 1. Network architecture of the proposed RVF-Net.

3.3. Yaw Loss Parameterization

While the original VoxelNet paper uses a simple yaw regression, we use a more complex parameterization to facilitate the learning process. Zhou [30] argues that a simple yaw representation is disadvantageous, as the optimizer needs to regress a smooth function over a discontinuity, e.g., from $-\pi$ rad to $+\pi$ rad. Furthermore, the loss value for small positive angle differences is much lower than that of greater positive angle differences, while the absolute angle difference from the anchor orientation might be the same. Figure 2 visualizes this problem of an exemplary simple yaw regression.



$$\theta_2 \gg \theta_1$$
$$L(\theta_2) \gg L(\theta_1)$$

Figure 2. Vehicle bounding boxes are visualized in a BEV. The heading of the vehicles is visualized with a line from the middle of the bounding box to the front. The relative angular deviations from the orange and blue ground truth boxes to the anchor box are equal. However, the resulting loss value of the orange bounding box is significantly higher than that of the blue one.

To account for this problem, the network estimates the yaw angle with a combination of a classification and a regression head. The classifier is inherently designed to deal with a discontinuous domain, enabling the regression of a continuous target. The regression head regresses the actual angle difference in the interval $[-\pi, \pi)$ with a smooth sine function, which is continuous even at the limits of the interval. The regression output of the yaw angle of a bounding box is

$$\begin{aligned} \theta_d &= \theta_{GT} - \theta_A \\ e_\theta &= sin(\theta_d), \end{aligned} \tag{1}$$

Appl. Sci. **2021**, *11*, 5598

where θ_{GT} is the ground truth box yaw angle and θ_A is the associated anchor box yaw angle.

The classification head determines whether or not the angle difference between the predicted bounding box and the associated anchor lies inside or outside of the interval $[-\pi/2, \pi/2)$. The classification value of the yaw is modeled as

$$c_{dir} = \begin{cases} 1, & \text{if } -\frac{\pi}{2} \leq (\theta_d + \pi) \bmod 2\pi - \pi < \frac{\pi}{2} \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

As seen above, the directional bin classification head splits the angle space into two equally spaced regions. The network uses two anchor angle parameterizations at 0 and $\pi/2$. A vehicle driving towards the sensor vehicle matches with the anchor at 0 rad. A vehicle driving in front of the vehicle would match with the same anchor. The angle classification head intuitively distinguishes between these cases. Therefore, there is no need to compute additional anchors at π and $-\pi/2$.

Due to the subdivision of the angular space by the classification head, the yaw regression needs to regress smaller angle differences, which leads to a fast learning progress. A simple yaw regression would instead need to learn a rotation of 180 degrees to match the ground truth bounding box. It has been shown that high regression values and discontinuities negatively impact the network performance [30]. The regression and classification losses used to estimate the yaw angle are visualized in Figure 3.

The SECOND architecture [31] introduces a sine loss as well. Their subdivision of the positive and negative half-space, however, comes with the drawback that both bounding box configurations shown in Figure 3 would result in the same regression and classification loss values. Our loss is able to distinguish these bounding box configurations.

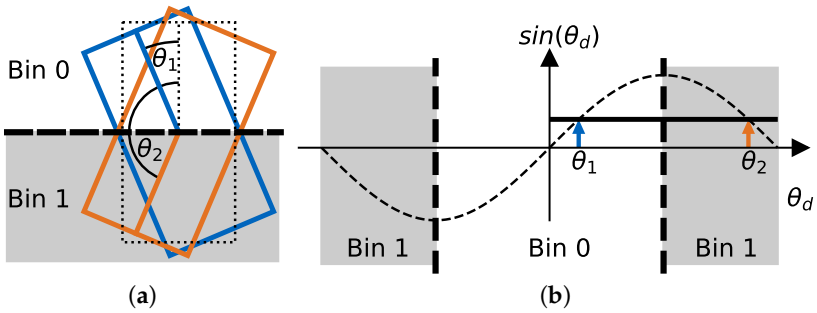


Figure 3. Visualization of our yaw loss. (**a**) Bin classification. (**b**) Sine regression. The bounding boxes in (**a**) are not distinguishable by the sine loss. The bin classification distinguishes these bounding boxes as visualized by the bold dotted line, which splits the angular space in two parts.

As the training does not learn the angle parameter directly, the regression difference is added to the anchor angle under consideration of the classification interval output to get the final value of the yaw angle during inference.

3.4. Data Augmentation

Data augmentation techniques [32] manipulate the input features of a machine learning method to create a greater variance in the data set. Popular augmentation methods translate or rotate the input data to generate *new* input data from the existing data set.

More complex data augmentation techniques include the use of General Adversarial Networks [33] to generate artificial data frames in the style of the existing data. Complex data augmentation schemes are beneficial for small data sets. The used nuScenes data set comprises about 34,000 labeled frames. Due to the relatively large data set, we limit the use of data augmentation to rotation, translation, and scaling of the input point cloud.

3.5. Inclement Weather

We expect the fusion of different sensor modalities to be most beneficial in inclement weather, which deteriorates the quality of the output of lidar and camera sensors. We analyze the nuScenes data set for frames captured in such environment conditions. At the same time, we make sure that enough input data, in conjunction with data augmentation, are available for the selected environment conditions to realize a good generalization for the trained networks. We filter the official nuScenes training and validation sets for samples recorded in rain or night conditions. Further subsampling for challenging conditions such as fog is not possible for the currently available data sets. The amount of samples for each split is shown in Table 1. We expect the lidar quality to deteriorate in the rain scenes, whereas the camera quality should deteriorate in both rain and night scenes. The radar detection quality should be unaffected by the environment conditions.

Table 1. Training and validation splits for different environment conditions. The table only considers samples in which at least one car is present in the field of view of the front camera.

Data Set Split	Training Samples	Validation Samples
nuScenes	19,659	4278
Rain	2289	415
Night	4460	788

3.6. Distance Threshold

Similar to [21], we argue that an IoU-based threshold is not the optimal choice for 3D object detection. We use both an IoU-based and a distance-based threshold to distinguish between the positive, negative, and ignore bounding box anchors. For our proposed network, the positive IoU-threshold is empirically set to 35% and the negative threshold is set to 30%. The distance threshold is set to 0.5 m.

3.7. Simulated Depth Camera

To simplify the software fusion scheme and to lower the cost of the sensor setup, lidar and camera sensor could be replaced by a depth or stereo camera setup. Even though the detection performance of stereo vision does not match the one of lidar, recent developments show promising progress in this field [34]. The relative accuracy of stereo methods is higher for close range objects, where high accuracy is of greater importance for the planning of the driving task. The nuScenes data set was chosen for evaluation since it is the only feasible public data set that contains labeled point cloud radar data. However, stereo camera data are not included in the nuScenes data set, which we use for evaluation.

In comparison to lidar data, stereo camera data are more dense and contain the color of objects in its data. To simulate a stereo camera, we use the IP-Basic algorithm [35] to approximate a denser depth image from the sparser lidar point cloud. The IP-Basic algorithm estimates additional depth measurements from lidar pixels, so that additional camera data can be used for the detection. The depth of these estimated pixels is less accurate than that of the lidar sensor, which is in compliance with the fact that stereo camera depth estimation is also more error-prone than that of lidar [36,37].

Our detection pipeline looks for objects in the surroundings of up to 50 m from the ego vehicle so that the stereo camera simulation by the lidar is justified as production stereo cameras can provide reasonable accuracy in this sensor range [38,39]. An alternative approach would be to learn the depth of the monocular camera images directly. An additional study [40] showed that the state of the art algorithms in this field [41] are not robust enough to create an accurate depth estimation for the whole scene for a subsequent fusion. Although the visual impression of monocular depth images seems promising, the disparity measurement of stereo cameras results in a better depth estimation.

3.8. Sensor Fusion

By simulating depth information for the camera, we can investigate the influence of four different sensors for the overall detection score: radar, camera, simulated depth camera, and lidar. In addition to the different sensors, consecutive time steps of radar and lidar sensors are concatenated to increase the data density. While the nuScenes data set allows to concatenate up to 10 lidar sweeps on the official score board, we limit our network to use the past 3 radar and lidar sweep data. While using more sweeps may be beneficial for the overall detection score through the higher data density for static objects, more sweeps add significant inaccuracies for the position estimate of moving vehicles, which are of greater interest for a practical use case.

As discussed in our main conclusions from the state of the art in Section 3, we fuse the different sensor modalities in an early fusion scheme. In particular, we fuse lidar and camera data by projecting the lidar data into the image space, where the lidar points serve as a mask to associate the color of the camera image with the 3D points.

To implement the simulated depth camera, we first apply the IP-Basic algorithm to the lidar input point cloud to approximate the depth of the neighborhood area of the lidar points to generate a more dense point cloud. The second step is the same as in the lidar and camera fusion, where the newly created point cloud serves as a mask to create the dense depth color image.

The radar, lidar, and simulated depth camera data all originate from a continuous 3D space. The data are then fused together in a discrete voxel representation before they are processed with the network presented in Section 3.2. The first layers of the network compress the input data to discrete voxel features. The maximum number of points per voxel is limited to 40 for computational efficiency. As the radar data are much sparser than lidar data, it is preferred in the otherwise random downsampling process to make sure that the radar data contributes to the fusion result and its data density is not further reduced.

After the initial fusion step, the data are processed in the RadarVoxelFusionNet in the same fashion, independent of which data type was used. This modularity is used to compare the detection result of different sensor configurations.

3.9. Training

The network is trained with an input voxel size of 0.2 m for the dimensions parallel to the ground. The voxel size in height direction is 0.4 m.

Similar to the nuScenes split, we limit the sensor detection and evaluation range to 50 m in front of the vehicle and further to 20 m on either side to cover the principal area of interest for driving. The sensor fusion is performed for the front camera, front radar, and the lidar sensor of the nuScenes data set.

The classification outputs are learned via a binary cross entropy loss. The regression values are learned via a smooth L1 loss [42]. The training is performed on the official nuScenes split. We further filter for samples that include at least one vehicle in the sensor area to save training resources for samples where no object of interest is present. Training and evaluation are performed for the nuScenes car class. Each network is trained on an NVIDIA Titan Xp graphics card for 50 epochs or until overfitting can be deduced from the validation loss curves.

4. Results

The model performance is evaluated with the average precision (AP) metric as defined by the nuScenes object detection challenge [1]. Our baseline is a VoxelNet-style network with lidar data as the input source. All networks are trained with our novel yaw loss and training strategies, as described in Section 3.

4.1. Sensor Fusion

Table 2 shows the results of the proposed model with different input sensor data. The networks have been trained several times to rule out that the different AP scores are caused

Appl. Sci. **2021**, *11*, 5598

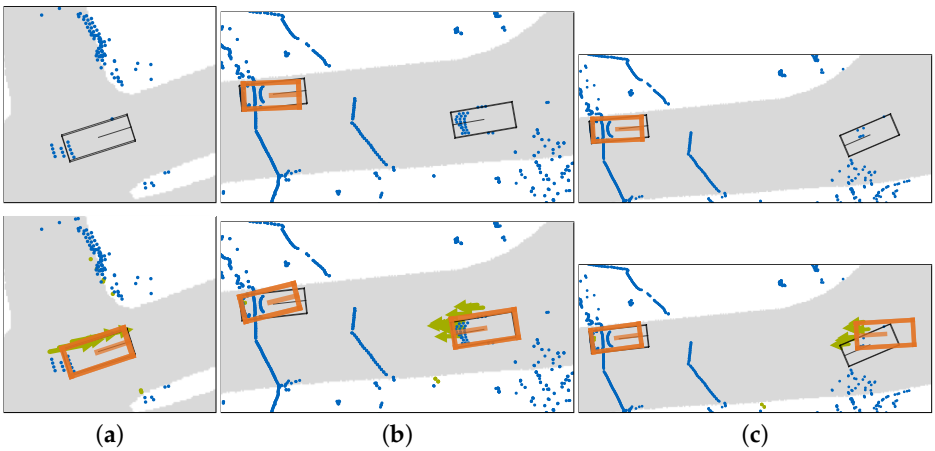
by random effects. The lidar baseline outperforms the radar baseline by a great margin. This is expected as the data density and accuracy of the lidar input data are higher than that of the radar data.

The fusion of camera RGB and lidar data does not result in an increased detection accuracy for the proposed network. We assume that this is due to the increased complexity that the additional image data brings into the optimization process. At the same time, the additional color feature does not distinguish vehicles from the background, as the same colors are also widely found in the environment.

The early fusion of radar and lidar data increases the network performance against the baseline. The fusion of all three modalities increases the detection performance by a greater margin for most of the evaluated data sets. Only for night scenes, where the camera data deteriorates most, does the fusion of lidar and radar outperform the RVF-Net. Example detection results in the BEV perspective from the lidar, RGB input, and the RVF-Net input are compared in Figure 4.

Table 2. AP scores for different environment (data) and network configurations on the respective validation data set.

Network Input	nuScenes	Rain and Night	Rain	Night
Lidar	52.18%	50.09%	43.94%	63.56%
Radar	17.43%	16.00%	16.42%	22.46%
Lidar, RGB	49.96%	46.59%	42.72%	61.66%
Lidar, Radar	54.18%	53.10%	47.51%	**68.01%**
Lidar, RGB, Radar **(RVF-Net)**	**54.86%**	**53.12%**	**48.32%**	67.39%
Simulated Depth Cam	48.02%	46.07%	39.07%	57.33%
Simulated Depth Cam, Radar	52.06%	48.31%	41.65%	61.04%



(**a**) (**b**) (**c**)

Figure 4. BEV of the detection results: Lidar and RGB fusion in the top row. RVF-Net fusion in the bottom row. Detected bounding boxes in orange. Ground truth bounding boxes in black. Lidar point cloud in blue. Radar point cloud and measured velocity in green. The street is shown in gray. (**a**) Only RVF-Net is able to detect the vehicle from the point cloud. (**b**) RVF-Net detects both bounding boxes. (**c**) RVF-Net detects both bounding boxes. However, the detected box on the right has a high translation and rotation error towards the ground truth bounding box.

The simulated depth camera approach does not increase the detection performance. The approach adds additional input data by depth-completing the lidar points. However, the informativeness in this data cannot compensate for the increased complexity introduced by its addition.

The absolute AP scores between the different columns of Table 2 cannot be compared since the underlying data varies between the columns. The data source has the greatest

influence for the performance of machine learning models. All models have a significantly higher scores for the night scenes split than for the other splits. This is most likely due to the lower complexity of the night scenes present in the data set.

The relative performance gain of different input data within each column shows a valid comparison of the fusion methods since they are trained and evaluated on the same data. The radar data fusion of the RVF-Net outperforms the lidar baseline by 5.1% on the nuScenes split, while it outperforms the baseline on the rain split by 10.0% and on the night split by 6.0%. The increased performance of the radar fusion is especially notable for the rain split where lidar and camera data quality is limited. The fusion of lidar and radar is also especially beneficial for night scenes, even though the lidar data quality should not be affected by these conditions.

4.2. Ablation Studies

This section evaluates additional training configurations of our proposed RVF network to measure the influence of the proposed training strategies. Table 3 shows an overview of the results.

To study the effect of the introduced yaw loss, we measure the Average Orientation Error (AOE) as introduced by nuScenes. The novel loss reduces the orientation error by about 40% from an AOE of 0.5716 with the old loss to an AOE of 0.3468 for the RVF-Net. At the same time, our novel yaw loss increases the AP score of RVF-Net by 4.1 percent. Even though the orientation of the predicted bounding boxes does not directly impact the AP calculation, the simpler regression for the novel loss also implicitly increases the performance for the additional regression targets.

Data augmentation has a significant positive impact on the AP score.

Contrary to the literature results, the combined IoU and distance threshold decreases the network performance in comparison to a simple IoU threshold configuration. It is up to further studies to find the reason for this empirical finding.

We have performed additional experiments with 10 lidar sweeps as the input data. While the sweep accumulation for static objects is not problematic since we compensate for ego-motion, the point clouds of moving objects are heavily blurred when considering 10 sweeps of data, as the motion of other vehicles cannot be compensated. Nonetheless, the detection performance increases slightly for the RVF-Net sensor input.

For a speed comparison, we have also started a training with non-sparse convolutions. However, this configuration could not be trained on our machine since the non-sparse network is too large and triggers an out-of-memory (OOM) error.

Table 3. AP scores for different training configurations on the validation data set.

Network	nuScenes
RVF-Net	54.86%
RVF-Net, simple yaw loss	52.69%
RVF-Net, without augmentation	50.68%
RVF-Net, IoU threshold only	55.93%
RVF-Net, 10 sweeps	55.25%
RVF-Net, standard convolutions	OOM error

4.3. Inference Time

The inference time of the network for different input data configurations is shown in Table 4. The GPU processing time per sample is averaged over all samples of the validation split. In comparison to the lidar baseline, the RVF-Net fusion increases the processing time only slightly. The different configurations are suitable for a real-time application with input data rates of up to 20 Hz. The processing time increases for the simulated depth camera input data configuration as the number of points is drastically increased by the depth completion.

Table 4. Inference times of different sensor input configurations on the NVIDIA Titan Xp GPU.

Network Input	Inference Time
Lidar	0.042 s
Radar	0.02 s
Lidar, RGB	0.045 s
Lidar, Radar	0.044 s
RVF-Net	0.044 s
Simulated Depth Cam, Radar	0.061 s
RVF-Net, 10 sweeps	0.063 s

4.4. Early Fusion vs. Late Fusion

The effectiveness of the neural network early fusion approach is further evaluated against a late fusion scheme for the respective sensors. For the lidar, RGB, and radar input configurations are fused with an UKF and an Euclidean-distance-based matching algorithm to generate the final detection output. This late fusion output is compared against the early fusion RVF-Net and lidar detection results, which are individually tracked with the UKF to enable comparability. The late fusion tracks objects over consecutive time steps and requires temporal coherence for the processed samples, which is only given for the samples within a scene but not over the whole data set. Table 5 shows the resulting AP score for 10 randomly sampled scenes to which the late fusion is applied. The sampling is done to lower the computational and implementation effort, and no manual scene selection in favor or against the fusion method was performed. The evaluation shows that the late fusion detection leads to a worse result than the early fusion. Notably, the tracked lidar detection outperforms the late fusion approach as well. As the radar-only detection accuracy is relatively poor and its measurement noise does not comply with the zero-mean assumption of the Kalman filter, a fusion of this data to the lidar data leads to worse results. In contrast to the early fusion where the radar features increased the detection score, the late fusion scheme processes the two input sources independently and the detection results cannot profit from the complementary features of the different sensors. In this paper, the UKF tracking serves as a fusion method to obtain detection metrics for the late fusion approach. It is important to note that for an application in autonomous driving, object detections need to be tracked independent of the data source, for example with a Kalman Filter, to create a continuous detection output. The evaluation of further tracking metrics will be performed in a future paper.

Table 5. AP scores of the tracked sensor inputs. The early fusion RVF-Net outperforms the late fusion by a great margin.

Network	nuScenes
Tracked Lidar	40.01%
Tracked Late Fusion	33.29%
Tracked Early Fusion (RVF-Net)	47.09%

5. Discussion

The RVF-Net early fusion approach proves its effectiveness by outperforming the lidar baseline by 5.1%. Additional measures have been taken to increase the overall detection score. Data augmentation especially increased the AP score for all networks. The novel loss, introduced in Section 3.3, improves both the AP score and notably the orientation error of the networks. Empirically, the additional classification loss mitigates the discontinuity problem in the yaw regression, even though classifications are discontinuous decisions on their own.

Furthermore, the paper shows that the early fusion approach is especially beneficial in inclement weather conditions. The radar features, while not being dense enough for an accurate object detection on their own, contribute positively to the detection result when

processed with an additional sensor input. It is interesting to note that the addition of RGB data increases the performance of the lidar, radar, and camera fusion approach, while it does not increase the performance of the lidar and RGB fusion. We assume that the early fusion performs most reliably when more different input data and interdependencies are present. In addition to increasing robustness and enabling autonomous driving in inclement weather scenarios, we assume that early fusion schemes can be advantageous for special use cases such as mining applications, where dust oftentimes limits lidar and camera detection ranges.

When comparing our network to the official detection scores on the nuScenes data set, we have to take into account that our approach is evaluated on the validation split and not on the official test split. The hyperparameters of the network, however, were not optimized on the validation split, so that it serves as a valid test set. We assume that the complexity of the data in the frontal field of view does not differ significantly from the full 360 degree view. We therefore assume that the detection AP of our approach scales with the scores provided by other authors on the validation split. To benchmark our network on the test split, a 360 degree coverage of the input data would be needed. Though there are no conceptual obstacles in the way, we decided against the additional implementation overhead due to the general shortcomings of the radar data provided in the nuScenes data set [7,43] and no expected new insights from the additional sensor coverage. The validation split suffices to evaluate the applicability of the proposed early fusion network.

On the validation split, our approach outperforms several single sensor or fusion object detection algorithms. For example, the CenterFusion approach [19], which achieves 48.4% AP for the car class on the nuScenes validation split. In the literature, only Wang [6] fuses all three sensor modalities. Our fusion approach surpasses their score of 45% AP on the validation split and 48% AP on the test split.

On the other hand, further object detection methods, such as the leading lidar-only method CenterPoint [14], outperform even our best network in the ablation studies by a great margin. The two stage network uses center points to match detection candidates and performs an additional bounding box refinement to achieve an AP score of 87% on the test split.

When analyzing the errors in our predictions, we see that the regressed parameters of the predicted bounding boxes are not as precise as the ones of state-of-the-art networks. The validation loss curves for our network are shown in Figure 5. The classification loss overfits before the regression loss converges. Further studies need to be performed in order to further balance the losses. One approach could be to first only train the regression and direction loss. The classification loss is then trained in a second stage. Additionally, further experiments will be performed to fine tune the anchor matching thresholds to the data set to get a better detection result. The tuning of this outer optimization loop requires access to extensive GPU power to find optimal hyperparameters. For future work, we expect the hyperparameters to influence the absolute detection accuracy greatly as simple strategies such as data augmentation could already improve the overall performance. The focus of this work lies in the evaluation of different data fusion inputs relative to a potent baseline network. For this evaluation, we showed a vast amount of evidence to motivate our fusion scheme and network parameterization.

The simulated depth camera did not provide a better detection result than the lidar-only detection. This and the late fusion approach show that a simple fusion assumption in the manner of "more sensor data, better detection result" does not hold true. The complexity introduced by the additional data decreased the overall detection result. The decision for an early fusion system is therefore dependent on the sensors and the data quality available in the sensors. For all investigated sub data sets, we found that early fusion of radar and lidar data is beneficial for the overall detection result. Interestingly, the usage of 10 lidar sweeps increased the detection performance of the fusion network over the proposed baseline. This result occurred despite the fact that the accumulated lidar data leads to blurry contours for moving objects in the input data. This is especially disadvantageous for objects moving

Appl. Sci. **2021**, *11*, 5598

at a high absolute speed. For practical applications, we therefore use only three sweeps in our network, as the positions of moving objects are of special interest for autonomous driving. The established metrics for object detection do not account for the importance of surrounding objects. We assume that the network trained with 10 sweeps performs worse in practice, despite its higher AP score. Further research needs to be performed to establish a detection metric tailored for autonomous driving applications.

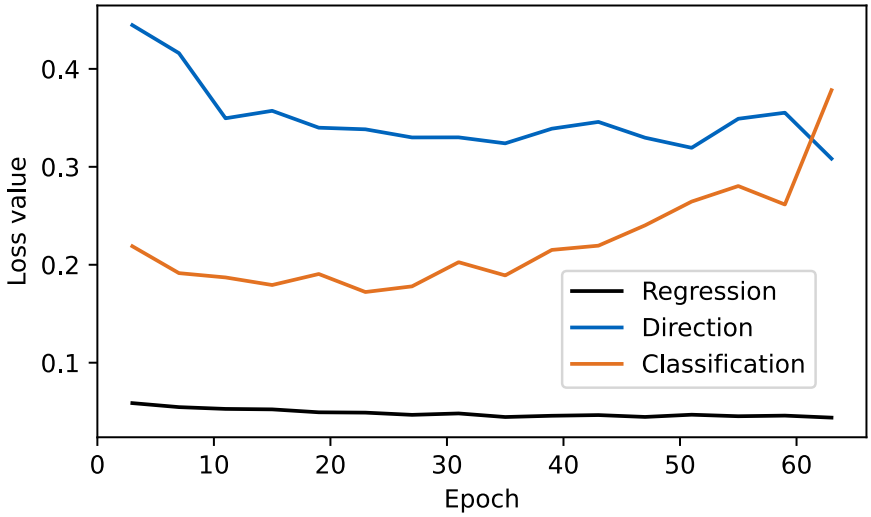


Figure 5. Loss values of the RVF-Net. The classification loss starts to overfit around epoch 30 while regression and direction loss continue to converge.

The sensors used in the data set do not record the data synchronously. This creates an additional ambiguity in the input data between the position information inferred from the lidar and from the radar data. The network training should compensate for this effect partially; however, we expect the precision of the fusion to increase when synchronized sensors are available.

This paper focuses on an approach for object detection. Tracking/prediction is applied as a late fusion scheme or as a subsequent processing step to the early fusion. In contrast, LiRaNet [44] performs a combined detection and prediction of objects from the sensor data. We argue that condensed scene information, such as object and lane positions, traffic rules, etc., are more suitable for the prediction task in practice. A decoupled detection, tracking, and prediction pipeline increases the interpretability of all modules to facilitate validation for real-world application in autonomous driving.

6. Conclusions and Outlook

In this paper, we have developed an early fusion network for lidar, camera, and radar data for 3D object detection. This early fusion network outperforms both the lidar baseline and the late fusion of lidar, camera, and radar data on a public autonomous driving data set. In addition, we integrated a novel loss for the yaw angle regression to mitigate the effect of the discontinuity of a simple yaw regression target. We provide a discussion about the advantages and disadvantages of the proposed network architecture. Future steps include the amplification of the fusion scheme to a full 360 degree view and the optimization of hyperparameters to balance the losses for further convergence of the regression losses.

We have made the code for the proposed network architectures and the interface to the nuScenes data set available to the public. The repository can be found online at https://github.com/TUMFTM/RadarVoxelFusionNet (accessed on 16 June 2021).

Author Contributions: F.N., as the first author, initiated the idea of this paper and contributed essentially to its concept and content. Conceptualization, F.N.; methodology, F.N. and E.S.; software,

E.S., F.N. and P.K.; data curation, E.S. and F.N.; writing—original draft preparation, F.N.; writing—review and editing, E.S., P.K., J.B. and M.L.; visualization, F.N. and E.S.; project administration, J.B. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: We express gratitude to Continental Engineering Services for funding for the underlying research project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. *arXiv* **2019**, arXiv:1903.11027.
2. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets Robotics: The KITTI Dataset. *Int. J. Robot. Res. IJRR* **2013**. [CrossRef]
3. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. *arXiv* **2019**, arXiv:1912.13192.
4. Julier, S.J.; Uhlmann, J.K. New extension of the Kalman filter to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*; SPIE Proceedings; Kadar, I., Ed.; SPIE: Bellingham, WA, USA, 1997; p. 182. [CrossRef]
5. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*. [CrossRef]
6. Wang, L.; Chen, T.; Anklam, C.; Goldluecke, B. High Dimensional Frustum PointNet for 3D Object Detection from Camera, LiDAR, and Radar. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1621–1628. [CrossRef]
7. Nobis, F.; Fent, F.; Betz, J.; Lienkamp, M. Kernel Point Convolution LSTM Networks for Radar Point Cloud Segmentation. *Appl. Sci.* **2021**, *11*, 2599. [CrossRef]
8. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
9. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum PointNets for 3D Object Detection from RGB-D Data. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
10. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018 .
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems 28*; Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2015; pp. 91–99.
12. Graham, B.; van der Maaten, L. Submanifold Sparse Convolutional Networks. *arXiv* **2017**, arXiv:1706.01307.
13. Graham, B.; Engelcke, M.; van der Maaten, L. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. *arXiv* **2017**, arXiv:1711.10275.
14. Yin, T.; Zhou, X.; Krähenbühl, P. Center-based 3D Object Detection and Tracking. *arXiv* **2020**, arXiv:2006.11275.
15. Chadwick, S.; Maddern, W.; Newman, P. Distant Vehicle Detection Using Radar and Vision. *arXiv* **2019**, arXiv:1901.10951.
16. Nobis, F.; Geisslinger, M.; Weber, M.; Betz, J.; Lienkamp, M. A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection. In Proceedings of the 2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 15–17 October 2019; pp. 1–7. [CrossRef]
17. Kowol, K.; Rottmann, M.; Bracke, S.; Gottschalk, H. YOdar: Uncertainty-Based Sensor Fusion for Vehicle Detection with Camera and Radar Sensors. *arXiv* **2020**, arXiv:2010.03320.
18. Kim, J.; Kim, Y.; Kum, D. Low-level Sensor Fusion Network for 3D Vehicle Detection using Radar Range-Azimuth Heatmap and Monocular Image. In Proceedings of the Asian Conference on Computer Vision (ACCV) 2020, Kyoto, Japan, 30 November–4 December 2020.
19. Nabati, R.; Qi, H. CenterFusion: Center-based Radar and Camera Fusion for 3D Object Detection. *arXiv* **2020**, arXiv:2011.04841.
20. Lim, T.Y.; Ansari, A.; Major, B.; Daniel, F.; Hamilton, M.; Gowaikar, R.; Subramanian, S. Radar and Camera Early Fusion for Vehicle Detection in Advanced Driver Assistance Systems. In Proceedings of the Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
21. Kim, Y.; Choi, J.W.; Kum, D. GRIF Net: Gated Region of Interest Fusion Network for Robust 3D Object Detection from Radar Point Cloud and Monocular Image. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2021.
22. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S. Joint 3D Proposal Generation and Object Detection from View Aggregation. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.
23. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-View 3D Object Detection Network for Autonomous Driving. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
24. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. PointPainting: Sequential Fusion for 3D Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.

25. Yang, B.; Guo, R.; Liang, M.; Casas, S.; Urtasun, R. RadarNet: Exploiting Radar for Robust Perception of Dynamic Objects. *arXiv* **2020**, arXiv:2007.14366.
26. Daniel, L.; Phippen, D.; Hoare, E.; Stove, A.; Cherniakov, M.; Gashinova, M. Low-THz Radar, Lidar and Optical Imaging through Artificially Generated Fog. In Proceedings of the International Conference on Radar Systems (Radar 2017), Belfast, Ireland, 23–26 October 2017; The Institution of Engineering and Technology: Stevenage, UK, 2017. [CrossRef]
27. Hebel, M.; Hammer, M.; Arens, M.; Diehm, A.L. Mitigation of crosstalk effects in multi-LiDAR configurations. In Proceedings of the Electro-Optical Remote Sensing XII, Berlin, Germany, 12–13 September 2018; Kamerman, G., Steinvall, O., Eds.; SPIE: Bellingham, WA, USA, 2018; p. 3. [CrossRef]
28. Kim, G.; Eom, J.; Park, Y. Investigation on the occurrence of mutual interference between pulsed terrestrial LIDAR scanners. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; pp. 437–442. [CrossRef]
29. Ng, A. A Chat with Andrew on MLOps: From Model-Centric to Data-Centric AI. Available online: https://www.youtube.com/watch?v=06-AZXmwHjo (accessed on 16 June 2021).
30. Zhou, Y.; Barnes, C.; Lu, J.; Yang, J.; Li, H. On the Continuity of Rotation Representations in Neural Networks. *arXiv* **2018**, arXiv:1812.07035.
31. Yan, Y.; Mao, Y.; Li, B. SECOND: Sparsely Embedded Convolutional Detection. *Sensors* **2018**, *18*, 3337. [CrossRef]
32. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*. [CrossRef]
33. Sandfort, V.; Yan, K.; Pickhardt, P.J.; Summers, R.M. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. *Sci. Rep.* **2019**, *9*, 16884. [CrossRef]
34. You, Y.; Wang, Y.; Chao, W.L.; Garg, D.; Pleiss, G.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. *arXiv* **2019**, arXiv:1906.06310.
35. Ku, J.; Harakeh, A.; Waslander, S.L. In Defense of Classical Image Processing: Fast Depth Completion on the CPU. In Proceedings of the 2018 15th Conference on Computer and Robot Vision (CRV), Toronto, ON, Canada, 8–10 May 2018.
36. Chen, Y.; Cai, W.L.; Zou, X.J.; Xu, D.F.; Liu, T.H. A Research of Stereo Vision Positioning under Vibration. *Appl. Mech. Mater.* **2010**, *44–47*, 1315–1319. [CrossRef]
37. Fan, R.; Wang, L.; Bocus, M.J.; Pitas, I. Computer Stereo Vision for Autonomous Driving. *arXiv* **2020**, arXiv:2012.03194.
38. Instruments, T. Stereo Vision-Facing the Challenges and Seeing the Opportunities for ADAS (Rev. A). Available online: https://www.ti.com/lit/wp/spry300a/spry300a.pdf?ts=1623899849893&ref_url=https%253A%252F%252Fwww.google.com%252F (accessed on 16 June 2021)
39. Wang, Y.; Chao, W.L.; Garg, D.; Hariharan, B.; Campbell, M.; Weinberger, K.Q. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. *arXiv* **2018**, arXiv:1812.07179.
40. Nobis, F.; Brunhuber, F.; Janssen, S.; Betz, J.; Lienkamp, M. Exploring the Capabilities and Limits of 3D Monocular Object Detection—A Study on Simulation and Real World Data. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–8. [CrossRef]
41. Zhao, C.; Sun, Q.; Zhang, C.; Tang, Y.; Qian, F. Monocular depth estimation based on deep learning: An overview. *Sci. China Technol. Sci.* **2020**, *63*, 1612–1627. [CrossRef]
42. Girshick, R. Fast R-CNN. In Proceedings of the ICCV 2015, Santiago, Chile, 7–13 December 2015.
43. Scheiner, N.; Schumann, O.; Kraus, F.; Appenrodt, N.; Dickmann, J.; Sick, B. Off-the-shelf sensor vs. experimental radar—How much resolution is necessary in automotive radar classification? *arXiv* **2020**, arXiv:2006.05485.
44. Shah, M.; Huang, Z.; Laddha, A.; Langford, M.; Barber, B.; Zhang, S.; Vallespi-Gonzalez, C.; Urtasun, R. LiRaNet: End-to-End Trajectory Prediction using Spatio-Temporal Radar Fusion. *arXiv* **2020**, arXiv:2010.00731.

5 Discussion

Section 5.1 answers the research questions. The relevance of the results of the thesis is discussed in Section 5.2. The chapter concludes by giving an outlook to future work and related object detection topics in Section 5.3.

5.1 Research Questions

This section revisits the sub research questions and discusses the advantages and disadvantages of the developed methods, before giving an answer to the main research question.

How to process radar data with deep learning methods?
For the limited quality radar data used in this research, PointNet++-based methods [149] provide enough capabilities to perform semantic segmentation. The limited data quality impedes higher classification scores with more complex models such as the proposed KPConv or LSTM architectures. The thesis has shown that the proposed methods perform on par with methods from literature. However, these proposed models increased the computational load without improving the overall performance.

Additional studies regarding the nuScenes radar data characteristics were performed [237]. Even when limiting the area of interest to the front radar and to a maximum distance of 50 m, 51 % of cars are not detected by a single radar point target. Even when the last three radar measurement cycles are accumulated, the same holds true for 37 % of the objects [237, p. 36]. Each radar used in the nuScenes data set records up to 125 point cloud targets per measurement cycle. The number of points is limited by the Controller Area Network (CAN) bus data rate. Point targets with a RCS value of under $-5\,\mathrm{dB\,m^2}$ are filtered from the detection results [238, p. 13]. While an increased segmentation performance could not be shown on the data set, further research with the proposed KPConv model architecture is recommended due to reasons for the limited performance found in the data.

The literature suggests that even the baseline PointNet++ approach is able to achieve higher segmentation scores on denser radar data [192]. As of August 2021, the recently released RadarScenes data set [70] can be used to evaluate semantic segmentation of moving objects in radar point cloud data. Once such data for both, moving and non-moving object detection are available to the public, the proposed KPConv-based architectures should be evaluated again to investigate if they provide better results than the current PointNet++ state of the art.

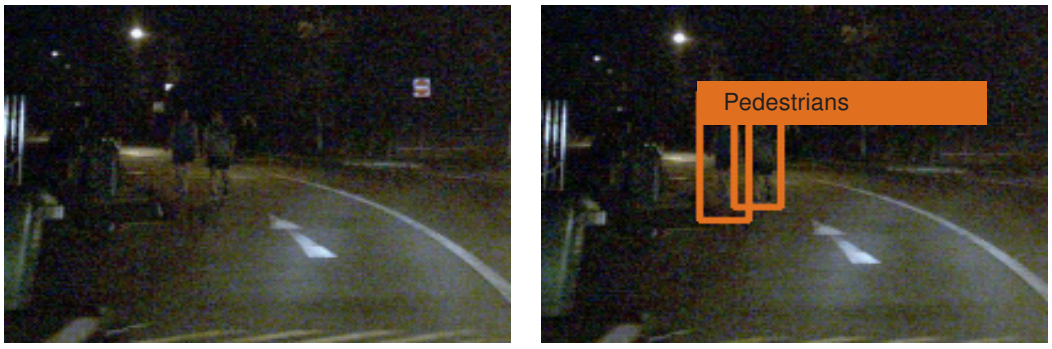
Incorporating the time dimension in unstructured point processing is explored as an additional measure for classification of radar data. The additional time processing produced results on par with the state of the art. At the same time, the proposed point-based LSTM architecture is computationally expensive. With the currently achieved performance, the point-based LSTM processing is in a too early stage to be recommended for a practical application.

DNN architectures like VoxelNet [154] that show quality results on higher resolution lidar data have not been explored for processing radar data in this thesis. A detection performance comparison between those modelling strategies cannot be given by this thesis. With higher resolution data both modelling approaches should be further explored. For lidar processing a combination of both architectures has resulted in favorable results [158].

To conclusively answer the research question, a PointNet++ network architecture [149] is empirically found as the most effective and efficient way for semantic segmentation of the radar data of the nuScenes data set. Rule-based approaches to process such limited quality radar data are not known to the author for comparison. More complex DNN approaches, do not increase the overall performance. Due to the limited performance of all methods on the preliminary semantic segmentation, a radar-only object detection method is not proposed.

Should radar and camera data be fused with low-level or high-level fusion methods? Section 4.2 has shown that the projection of radar data onto the image plane is an effective measure to fuse the two data sources originating from perpendicular planes in the 3D space. The distance information and additional radar features are preserved in the projection. The combined early and feature fusion of radar and camera data outperforms the camera baseline on the whole data set.

Additional studies with CRF-Net have shown that the data fusion-based object detection is more robust against noise in the image data than the baseline [75, p. 81]. While the fusion method provides favorable results for the whole data set, the fusion shows an even greater performance increase against the baseline for rain and night scenes where the camera data quality is deteriorated [239]. Figure 5.1 exemplary shows the benefit of the low-level fusion in comparison to the camera baseline for scenes recorded during the night.



(a) without radar fusion (b) with radar fusion

Figure 5.1: Object detection results at night. (a) The pedestrians in front of the vehicle are not detected from camera input data. (b) The pedestrians are detected by fusing additional radar data in CRF-Net.

An additional classification head is added to the network to estimate the distance to objects [75, p. 75] so that the 2D fusion enables a 3D position estimation. This research direction can be further explored to generate a full 3D pose estimation from the 2D projection. However, the evidence from concurrent literature suggests that 2D projection-based object detection methods are not as precise as 3D methods. This thesis therefore recommends a 2D fusion when a), no lidar data are available or b), when a precise 3D localization performance for all objects is not required. This could be in applications with guaranteed slow relative velocities, such as production environments, as the performance of the 2D fusion is more precise for close vehicles.

The motivation for the low-level fusion in the neural network lies in the joint processing of the features of both sensors. In this case, even weak features from the radar, when complemented by weak features from the camera at night time, can generate a strong detection result. In a classical high-level fusion the potential of those weak features is lost. A sole projection of the radar data to the image plane, followed by a DNN object detection, does not result in accurate object proposals. A probabilistic high-level fusion of these inaccurate object proposals with object proposals from camera processing, deteriorates the overall object detection result against the camera baseline so that a high-level fusion is not recommendable in this case. Both, the proposed low-level fusion method in the 2D space and the camera-only method prove to be more effective. The combination of weak features in the processing allows to recognize certain similarities to gradient boosting machine learning methods where weak classifier models are combined to produce one strong classifier output [240]. It is yet to see if the further development of low-level fusion models can augment to similar popularity and performance advantages.

In recent years, further research works have proposed low-level camera and radar data fusion approaches for object detection. Though concurrently developed, the 2D fusion approach developed in this thesis can be seen as an extension of the fusion work of Chadwick et al. [215] by incorporating multi-level fusion and height extension for the radar projections in the processing. Since then, other works have proposed extensions building on similar low-level fusion concepts [219, 226, 241]. The first two works [219, 241] process radar and camera data in two separate input branches before performing a feature fusion by projecting the radar data into the image space. Nabati and Qi [226] perform a mixed early and feature fusion by fusing pre-processed image and raw radar features in the image space. The work of Kowol et al. [242] also uses a radar projection but uses separate neural networks for both data sources. The features generated by these networks are combined in a gradient boosting fusion of both data.

Despite the rising number of publications proposing 2D low-level fusion approaches, there are also downsides to the fusion by 2D projection. Pair-wise distant radar detections in the 3D space might be close in the projected 2D space which may limit the resulting object detection performance. This downside is explicitly addressed in the recent work of Nabati and Qi [226]. The paper mitigates the problem by generating a camera depth estimate in a pre-processing step to filter out radar detections not belonging to an object of interest before the fusion. As a downside, this approach lowers the potential of the complementary processing of the fusion, since the camera depth estimation needs to be reliable on its own in the first step.

A further drawback of the fusion in the 2D space is that the relative image grid resolution decreases for distant objects. While a projection error of one pixel on the ground close to the vehicle might correspond to a 3D displacement error of a few centimeters, one pixel error at far distances might correspond to a displacement error of several meters. The association in the projection space is thereby more susceptible to errors for objects at far distances. Objects that are in direct line of sight of each other from the camera point-of-view might project to the same position in the image location. In this case, the projection discards the more distant object, while both objects may be easily distinguishable in the original radar data representation. As physically close objects are in general of greater importance to the autonomous driving use-case, this drawback does not impede a successful application.

Competing fusion concepts which do not possess theses drawbacks, e.g. a high-level fusion or a low-level fusion by projecting camera data into the 3D space, are not suitable for the use-case presented in this thesis. A fusion by projecting the camera data to the 3D space is not feasible due to the insufficient accuracy of monocular depth estimation methods [90]. Therefore, a fusion

of camera and radar data in the 3D space cannot be performed with a reasonable spatial coherence.

For a high-level fusion, each sensor modality needs to provide sufficiently discriminant information to produce reasonable object detection results on their own which can subsequently be fused. For example, the Kalman filter uses a zero mean assumption for the measurement noise to guarantee optimality [198]. This is not given by the trained radar-only DNN which produces object detection proposals with no reasonable relation to the distribution of ground truth objects due to the insufficient information content in the radar data. For higher performance sensor-individual object detection pipelines, a high-level fusion can be a reasonable alternative due to the maturity of such approaches in practice.

In conclusion, this thesis empirically determines a low-level fusion of camera and radar data in the 2D space to be the most effective approach for the presented object detection use-case. Strengths and weaknesses of the proposed and alternative approaches were discussed to guide the selection of an adequate approach for future object detection use-cases with respect to their individual boundary conditions.

Should radar, camera, and lidar data be fused with low-level or high-level fusion methods? The thesis proposes an early fusion method for radar, camera, and lidar data. The camera data are projected to the lidar point cloud. The augmented lidar point cloud is then fused in 3D voxels with the radar data. This combined data tensor is the input for a voxel-based neural network. The fusion approach is also tested without the additional camera input which leads to an increased performance against the lidar baseline as well. The fusion of all modalities, however, resulted in the highest detection score. The presented 3D fusion paper [232] is the first to process all three sensor modalities jointly in a DNN for object detection.

Similar to the radar and camera fusion in Section 4.2, the joint processing of the weak radar features with the strong lidar features proves more effective than the high-level fusion of both data. A competitive 3D object detection performance is not achieved in the current literature from both production camera-only or radar-only data for object detection in the 3D space, as seen in the respective nuScenes and KITTI leaderboards [39, 40]. Current radar processing is only mature enough for production applications for moving object detection. Camera detection is mature for 2D applications such as traffic sign recognition or the assisted detection of objects where a precise distance estimate for the objects is not required. Despite the low radar data quality, the weak information can still be leveraged in a low-level fusion, whereas it deteriorates the result of a high-level fusion.

From a practical point of view, a high-level fusion needs reasonable detection results of all independent measurement sources to be beneficial. Such reasonable results could not be obtained from the radar-only data of nuScenes, negatively impacting the overall result of the high-level fusion object detection. There is no proof that a low-level fusion will generate better object detection results for all cases. However, this performance advantage could empirically be demonstrated for 2D and 3D object detection use-cases on the nuScenes data set in this thesis.

The relative benefit of the fusion increased for night and rain scenes, demonstrating the robustness of the approach. While the deterioration of the lidar detection in rain conditions is expected due to increased noise, the reason for the relatively weaker performance of the lidar object detection in night conditions is not found by the author. The relative benefit of the fusion input over the lidar input for night scenes in comparison to the fusion input over the lidar input for all conditions in the data set is consistently observed for different network configurations. Due to

the small relative performance gain, however, it cannot be ruled out that the higher performance is a non-significant observation and should be investigated further.

The relative performance gain towards lidar-only and high-level fusion baselines motivates further research for the proposed fusion method. Point cloud-based fusion methods are generally feasible as well. However, the performance advantages of the voxel-based methods found in literature and the additional effort for the network structure to combine the feature dimensions of the different modalities in a point-based fusion favor the usage of a voxel-based fusion. As with all research focusing on data-based methods, the reader should keep in mind that there is no guarantee that this empirical advantage will permanently hold true for further advancements in the respective areas.

A feature fusion has not been developed in this work, but due to the non-strict border between early and feature fusion approaches, both approaches possess further potential for a fusion of radar, camera, and lidar data. More studies with additional data about low-level fusion methods are needed to explore the potential in this field.

Wrong predictions of the orientation of objects by the current implementation are a main reason for the inferior performance of the model derived in this thesis to the lidar state of the art [131]. Even though the newly proposed loss function in Section 4.3 improved the overall result by a large margin, there is further potential in tuning the networks' overall performance. Access to additional compute power would help to parallelize the exploration of the model parameter space further, by reducing the overall needed training time for additional experiments.

To conclude the research question, the evidence generated on the nuScenes data set leads to the proposal of a voxel-based low-level fusion method of radar, camera, and lidar data for 3D object detection. The developed DNN reaches higher 3D mAP scores than a state of the art high-level fusion approach. The advantage of the low-level fusion could be demonstrated for challenging conditions such as rain and night time.

Can the usage of radar data make camera and lidar object detection more robust?
The robustness of object detection with automotive radar data has been studied with three different approaches in this thesis. The used radar data alone do not enable a robust object detection due to the limited data quality. Both developed fusion approaches prove that an increased detection performance and robustness can be achieved by the low-level fusion of radar data with additional sensor data in comparison to high-level fusion approaches.

The robustness of such fusion approaches is demonstrated by the relative performance increases to the baselines for rain and night conditions. The performance increase of the fusion approaches is especially notable since the radar data quality in the nuScenes data set is limited and other authors forgo working on the nuScenes radar data due to the limited prospect of developing a successful model on the data [71, 73]. Both, projected and voxel low-level fusion methods, show further potential for more precise and robust object detection.

At the same time, the development of deep learning-based sensor fusion methods is still in its early stages. The performances achieved in this thesis are still far from enabling safe and robust autonomous driving in unconstrained environments. While the input data are arguably the greatest challenge for successful detection strategies (Section 5.3.2), the algorithm also needs to be developed further. At the beginning of this research, deep learning-based radar fusion was not covered in literature at all. The publication of additional fusion approaches in parallel to this thesis, presented in Section 2.3.4, shows a rising interest in this area. With the release of more

labeled radar data sets, it is expected that more radar fusion methods will be developed [35, p. 1354].

A sole thesis cannot determine which sensor modality or fusion approach will be the most beneficial approach for future applications of object detection. The influence of sensor technology advancements, increased computational resources or newly proposed algorithms can have a major impact for the future development of sensor fusion methods. It can, however, provide a recommendation based on the empirical results and the discussion of related fields. In this thesis, the usage of radar data alone did not lead to a robust object detection result. In contrast, the fusion of radar and camera data, as well as the fusion of radar, camera, and lidar data have proven beneficial in comparison to their respective baseline approaches. The usage of radar data in the form of a low-level deep learning fusion with additional sensor modalities is therefore recommended as a processing approach for robust object detection.

5.2 Practical Relevance

Finding the answers to the open research questions alone does not enable a final assessment of the proposed systems. For the adaption and further development in this field, scientific, commercial, and societal aspects need to be considered to determine the relevance of the development. This section discusses topics which are explicit and implicit conclusions from the presented research work.

Scientific Relevance
All proposed solutions of this thesis are released under open source licences on GitHub [234–236]. The practice of making algorithms publicly available has led to a speed up of the development of camera- and lidar-based algorithms. This thesis contributes its part to the development of radar-centric algorithms to serve as a source of reference and inspiration for future work of more advanced object detection algorithms.

The publications that comprise the core of this thesis are among the first works in their respective fields. Though far from solving the underlying problem, they serve as references for further developments of low-level fusion methods using radar data. Recent publications of human activity recognition [214] and multi-modal classification [233] have demonstrated similar advantages of low-level fusion methods in terms of performance and robustness in adjacent fields. The proposed radar and camera fusion method of this thesis [231] is already adapted and extended in several works in literature [219, 226, 241]. The code is widely adapted in the open source community as indicated by the statistics on GitHub [235].

The proposed object detection systems are validated on the public nuScenes data set. This data set comprises real-world sensor data. The validation on realistic sensor data information from practical use-cases could therefore be shown. The validation of a closed-loop autonomous driving software is still outstanding and should be postponed until the relevant points for future work in Section 5.3.1 are solved.

Commercial Relevance
Due to the lower price of radar and camera in comparison to lidar sensors, the fusion of camera and radar data might be a beneficial approach for commercial applications which have cost-effective requirements. Private companies take inspiration of the ideas and code published in this work in their current developments [243, 244].

Enabling autonomous driving in an increased number of environment conditions is an important factor to bringing autonomous driving into a more mature state outside of restrictively defined ODDs [245]. The fusion of radar, camera, and lidar data can help to enlarge those ODDs for a broader application of autonomous driving.

The time-consuming and uncertain outcome of the training of deep learning algorithms, as developed in this thesis, has shown that an adaption of the used workflow might be difficult to implement in current commercial organizations. For engineering companies, the switch from rule-based to data-based development [246] might also require the adaption of organizational structures to new development workflows. In the case that an object detection algorithm does not detect an object, a classical algorithm would be analyzed and an issue would be created that addresses the identified problem in the code. With a data-based algorithm, such an exact analysis to find the root cause might not be possible. Solving the problem could furthermore involve the collection of new data from fleet vehicles, labeling of new data, and retraining of the network. An explicit involvement of a software engineer would not be required, but rather a team working on a shared task at different stages. The further advancement of machine learning methods could impact the work style, not just of developers, but also of entire companies in the radar sector.

The effectiveness of a deep learning system is not only dependent on the quantity but more importantly on the quality of the training data. This thesis has proven this fact once more for the application of deep learning methods on radar data. A data set comprising 100,000 samples of a static environment might be less effective than a data set comprising 100 distinct samples with relevant variance. Adequate data sets need to be tailored to the used sensors and intended use-cases. The amount of samples needed to train a robust system needs to be inferred iteratively by measuring relevant metrics and explicitly targeting related weaknesses in the data distribution. Developing such a system requires a mature data collection, labelling, and data selection process that needs to be steered coordinately for a successful development.

Even if well-founded adaptions of the training data, process, or software are applied, there is no guarantee that the initial problem of the algorithm will be solved by the proposed adjustment and new iterations may be required. This newly introduced source of uncertainty in the development increases the effort needed to make a reliable system available to the market. This is also visible in the number of mergers and cooperations between big automotive and software companies for the development of autonomous systems [247–249]. Even big companies do not have all monetary resources and talent at hand to solve the challenges on their own.

It is argued, e.g. by Tesla, that camera-based systems suffice for autonomous driving applications, since humans use a similar sensory system to perceive their environment. However, deep learning-based 3D depth estimation from camera data is a critical point as safety requirements need to be met for an autonomous software stack [250]. Current planning algorithms [251] require precise knowledge of the environment for adequately handling dynamic traffic scenarios, rough estimates are not sufficient. Radar and lidar sensors provide such precise spatial information. It is questionable whether legislators will allow the use of camera-based estimated depth information for safety critical tasks, further motivating the use of a diverse sensor suite and data fusion methods.

Societal Relevance

The low-level fusion of different modalities has proven to be effective for non-ideal conditions such as rain and night time. Due to the frequent occurrence of such conditions in everyday driving, low-level fusion methods could become an important building block to enable autonomous driving

in a variety of non-ideal everyday conditions. The author expects that a low-level fusion can also be beneficial for complex scenarios where objects are only partly visible in each individual sensor or environment conditions limiting the sensors, such as rain, fog and direct sunlight blinding. The additional robustness, paired with a visualization for the passengers, can increase the trust of the society in such a technology and enable a wider adoption of autonomous driving.

The work on robust object detection algorithms contributes to the wide diffusion of autonomous vehicles for everyday driving. Autonomous driving itself, if widely applied, will cause a revolution in the field of transportation. The number of parking spaces, which take as much as 24 % of the surface of American cities, could be reduced [8]. Owning a car might no longer be a necessity even in more rural areas, reducing overall costs further. Vehicle crashes could be reduced by 90 % [8]. A recent report projects that using a driverless taxi will cost less than half of using a human driven taxi [252]. Analysts of Morgan Stanley project the annual benefit generated by autonomous vehicles at 1.3 trillion dollars annually in the United States of America alone. Less traffic congestion and productivity gains from working on the road contribute to benefits of 571 billion dollars [253]. However, the future in which autonomous driving is a reality is still distant due to a variety of challenges that need to be solved. In 2021, Litman predicts that autonomous vehicles might be available within 5 to 25 years [252, p. 29]. This high uncertainty in the estimation alone shows that the road of research is still long until robust systems will be a reality.

5.3 Outlook

This work presented radar-centric low-level fusion methods for robust object detection in inclement weather conditions. Directions for future work based on this thesis are given in Section 5.3.1. Due to its importance, future work in the field of radar hardware and data sets is discussed in Section 5.3.2. In addition, a broad variety of open research questions in the field of object detection exist which are briefly discussed in Section 5.3.3 to give a conclusive overview of the field.

5.3.1 Future Work

The radar only object detection approach could not be conclusively evaluated through the lack of adequate data. With the release of the RadarScenes [70] data set, the network should be re-evaluated for the task of moving object detection to investigate the performance on a more dense radar-centric data set.

For 2D sensor data fusion, benefits of the low-level data fusion over a high-level fusion are shown. However, the explicit knowledge of the distance information of point targets could potentially get lost in the deep learning processing during the DNN processing for the 3D position estimation. One could explore an additional rule-based fusion of the radar measured distance values to the estimations of the neural network, to see if the 3D detection performance can be augmented in this way.

The fusion of lidar data with additional modalities in a 2D projection was not developed in this thesis due to the better performance of 3D fusion methods in literature. For a use-case which is constrained by computational resources, the 2D fusion of all modalities could be explored if limitations in the resulting performance can be tolerated for the intended application.

A visualization of the detected objects has shown that wrong bounding box orientation estimations have a negative influence on the overall 3D object detection performance of the developed algorithm. Further tuning of the network parameters and a balancing of the partial losses should be explored to increase the performance of the orientation estimation and the overall algorithm.

The methods developed in this thesis are evaluated on the official nuScenes validation data set. However, the fusion algorithms are only applied to the front-facing sensors of the vehicle. For an evaluation on the official test data set, the fusion approaches need to be expanded to a 360° object detection by incorporating all sensors around the vehicle. Even though literature suggests that the results on the validation and test set of nuScenes do not differ significantly, this additional sensor coverage would increase the validity of the results further.

The highest positive impact on the results of the radar-centric approaches is expected if the official object labels are revised and adapted to the needs of the radar sensors. This includes adding new bounding boxes for objects which are only visible in the radar and camera data but not in the lidar data, as well as adapting the labeled bounding boxes to the calibration of the radar sensors.

5.3.2 Radar Hardware and Data

Regardless of whether classical algorithms or deep learning methods are used, the software processing is limited by the data it operates on. This section therefore gives a brief overview of current radar hardware and data. It gives an outlook to future requirements for data sets for radar-centric and low-level fusion object detection methods.

With respect to a single radar sensor, the resolution and resulting data density of the provided point cloud is a challenge for object detection. Scheiner et al. [71] perform clustering and classification of point cloud data of two different sensor hardware generations. The increased resolution of the newer sensor hardware enabled a relative performance increase of the clustering result by 23 % and of the classification result by 13 %. Engels et al. [73] specify a current high-resolution radar with a distance resolution of 0.2 m and angular resolution of under 2.0° which is similar to the radar used in the aforementioned work [71]. In comparison, the resolution of the nuScenes data set is specified sparser with a distance resolution of 0.4 m–1.8 m and angular resolution of 3.2°–12.3°. On the other hand, current lidar sensors [254] achieve even higher resolutions at accuracies of 0.01 m and 0.07°, respectively. The expected availability of higher resolution radar input data in the coming years will have a continued positive effect on the achievable detection performance [73, 255, 256]. The additional measurement of the elevation component of high resolution radar sensors is expected to further increase the object detection performance [69, 223]. The radar-centric methods presented in this thesis have already proven their ability to handle denser 3D data through their application to high resolution lidar data in both, literature and in the thesis itself. The robustness and range rate measurement of the radar combined with a higher resolution could create an increased potential of radar sensors for autonomous driving in the future.

This thesis argues that in the foreseeable future, a multi-modal sensor system of radar, camera, and lidar will be necessary to enable safe autonomous driving. Temporal synchronized simultaneous recording of the input sensors is an important factor for the final obtainable performance of an object detection system that is not considered in public data sets. Without a simultaneous recording, contradicting information about the poses of moving objects is present in the input data when used in a low-level fusion approach. While Precision Time Protocol (PTP) synchro-

nization is already available in a variety of current systems, external recording trigger possibilities are not widely employed [257]. The agreement on a shared specification for the interface to synchronization and data recording triggering amongst vendors would facilitate the fusion of different sensor models and generations for a range of use-cases and precision requirements.

Spatial calibration is of equal importance for unimodal and multi-modal object detection. Several approaches are proposed to calibrate sensors with respect to artificial targets. These include calibration for omni-directional cameras [258], and the extrinsic calibration of radar, camera, and lidar sensors [259].

Both the extrinsic and intrinsic sensor parameters can change over time due to mechanical and thermal influences. Online re-calibration is therefore a pre-requisite for the fleet application of autonomous vehicles. Neural networks have been proposed for target-less online calibration of camera and lidar sensors [260], and camera and radar sensors [261]. Online calibration of the intrinsic camera parameters can be performed by using the known shape of traffic signs [262].

Static re-calibration of sensors is done for the recording of the nuScenes data set twice a week [40]. Miscalibrations between radar sensor data and the ground truth data, generated from lidar data, are nonetheless clearly visible, especially for distant objects throughout the data set. If miscalibrated input data are fed into a neural network, the optimization might be able to average out some outlier effects but will not be able to mitigate systematic errors present in the data. Providing exact spatial calibration information at all times is therefore an important open challenge for future data sets and real world applications of autonomous driving.

The selection of appropriate samples for a data set is of further importance. A variety of environment conditions and traffic scenarios need to be covered in a data set to get an accurate performance estimation for real scenarios. The selection of the right data for such a data set might be of greater importance than the choice of the actual algorithm as suggested by A. Ng [246]. From a practical application point of view, data selection, monitoring, and management are important organizational challenges to generate a suitable data set in an iterative way. Tools to annotate 3D sensor data are both available open-source [263–265] and from commercial vendors.

Once reliable data processing methods are developed, the computational and memory resources required for the processing methods need to be further optimized [266, 267]. Hardware and software components for the sensor data processing can be considered jointly to increase the real-time capabilities of the system [268, 269]. With the expected sensor resolution increase, the demand for computational hardware resources for the data processing will increase further.

Section 2.1 has shown that no universal data set with quality radar data is available to the public. This section has discussed a variety of challenges for the use of multi-modal sensor data for object detection. A data set, addressing these challenges, could facilitate further research in the field of radar-centric data fusion for object detection.

5.3.3 Object Detection

Section 5.3.2 showed that the creation of radar-centric data sets is an important step to facilitate further radar research. This section discusses software processing related open questions for object detection which are, in contrast to the topics in Section 5.3.1, only implicitly derived from the work of this thesis.

Recording realistic data from simulation would drastically decrease the effort needed for data-related tasks when developing an object detection system. However, current simulation environments are not able to produce such high-fidelity sensor data so that scenario mining from real data remains obligatory [270, 271]. The simulation of radar data with its complex interactions with the environment is a notable challenge in this field [272, 273].

Irrespective of the data themselves, there are further open research directions for radar-centric object detection model development. Operating on radar cube data brings the benefit of introducing even more dense data to the neural network processing, generating additional potential for the object detection performance. Some first works have been released to process the full 3D radar cube data with deep learning methods [191]. At the same time, the amount of data that needs to be handled by such methods increases compared to point-based radar processing, leading to additional computational resource requirements.

This work fuses perception sensors for object detection. Information from additional sources such as the outline of a prior map can be added to the fusion approach to increase the amount of input data further [274]. Map features indicating a wall should make the prediction of a vehicle at the same location unlikely.

Standard object detection approaches can only detect objects of known classes from the training set. In the practical application, an autonomous vehicle might encounter objects that do not belong to any of the pre-trained classes. These open-set conditions can occur at any time on the road and need to be handled [275, 276].

While some form of probability is estimated for the classification part of object detection, the bounding box is estimated with a fixed pose in the 3D space. The network does not provide probabilistic certainty information about the spatial pose estimation. Probabilistic object detection deals with this problem by not only predicting a location of an object, but also an uncertainty estimate for the correctness of this predicted location [277, 278].

Using current metrics for object detection, one compares different methods depending on the proportions of correct and incorrect detections of objects in the data. For the autonomous driving task, however, some objects are of greater important than others. This may lead to the selection of a lower performance method for the intended task by relying on current metrics. Appropriate test methods [250] and respective metrics [279] need to be conceived for the safe deployment of autonomous vehicles.

In a scenario where vehicles are no longer controlled by humans, the abuse of the computer driving the vehicle can have severe impacts. In addition to manipulating the code on the system itself, deep learning-based systems are also susceptible to special visual patterns that can be maliciously displayed to the object detection system to cause a failure of the system. Certain textures of materials can lead to a detection of an object in empty space or prevent the network from identifying an object in plain sight. More research is needed to avoid such adversarial attacks on sensors of all modalities [280–282].

6 Summary

The thesis developed methods for object detection in an automotive context. Object detection is identified as one of the important technical challenges for autonomous driving. The performance of lidar- and camera-based object detection systems is limited in inclement environment conditions. Radar sensor data are less affected by such conditions, making it a preferred modality for robust object detection. Its low resolution, however, limits the number of use-cases that can be solved by radar data processing alone. With the advent of higher resolution radar sensors, this thesis develops radar-centric processing and multi-modal fusion methods for robust object detection.

The introductory chapter highlights the economic and societal benefits of autonomous driving applications, for which object detection is a prerequisite. It continues to define the terminology of automotive perception and gives an overview of the object detection task.

After the introduction, the thesis reviewed related work for object detection in Chapter 2. As deep learning-based methods are the most effective in the field of object detection, publicly available data sets for training object detection DNNs are reviewed. While there is no radar-centric data set for the training of radar-based object detection systems, the nuScenes data set is identified as the only adequate data set to enable current developments. A short overview of performance metrics to measure the performance of machine learning-based object detection models is given before reviewing the state of the art of object detection in an automotive context. Camera, lidar, radar, and sensor fusion methods are reviewed separately. The focus of the review is on radar and radar-centric fusion methods with additional modalities.

Conclusions from the related work are drawn in Chapter 3. While a lot of work covers deep learning-based object detection with camera and lidar data, radar and radar-centric deep learning fusion methods were not available before writing this thesis. Motivated by the success of the application to lidar and camera data, this research further developed radar-centric deep learning approaches with the goal to leverage the robustness of radar sensors for the final detection result. The research is structured into three sub fields with respective research goals: Exploring new methods for radar data processing, radar and camera fusion, as well as radar, camera, and lidar fusion.

The results of the thesis are presented in the form of prior publications in Chapter 4. A DNN for the direct processing of irregular point cloud data is adapted to the radar domain. A semantic segmentation is performed as an intermediate step towards object detection. Despite leveraging the range rate and signal intensity features of the radar, the presented KPConv approach reaches similar performance scores as the previous deep learning-based state of the art. The additional adaption of recurrent network structures to point cloud data, motivated by classical radar processing, could not create an additional performance gain. An additional investigation showed shortcomings of the radar data and labels in the nuScenes data set as the main source for the limited performance. Another evaluation of the method is recommended once more dense data with correct labels are available.

To further evaluate the potential of radar-centric object detection, a fusion with camera data is employed to generate more dense input data. The radar data are projected onto the camera image plane and fused at different abstractions levels with the camera data. It was shown that the shared processing of the data in a combined early and feature fusion in a DNN outperforms the camera-only and high-level fusion baselines. The performance gain is especially visible for rain and night scenes where the robustness of the radar sensor can be leveraged. The low-level fusion approach enables the utilization of weak features which otherwise would be discarded due to the data abstraction in a high-level fusion approach.

Due to the importance of lidar data in the state of the art of object detection, a third development is presented which fuses radar, camera, and lidar data for object detection. The data are fused in an early fusion approach in a regular voxel grid and then processed with sparse convolutions. A novel loss is proposed which increases the detection performance compared to the baseline regression loss. The low-level fusion outperforms the high-level fusion baseline. The robustness of the proposed method is shown by outperforming the baseline by an even greater margin for rain and night scenes.

The presented methods are discussed within their respective publications. The research questions are answered in more detail in Chapter 5. The usage of low-level fusion methods, as used in the publications of this thesis, is proposed as a promising research direction for robust object detection. Directions for future research building on this thesis are given. The impact of the developments for the research community and its potential for the industrial adoption and application in society are discussed.

While the thesis provides promising directions for robust object detection, there is still a number of obstacles in the way of enabling safe autonomous driving in all conditions. The outlook in Section 5.3 identifies data selection and hardware development as important factors to increase the object detection performance of radar-centric object detection methods. Radar cube processing, probabilistic object detection, and sensor simulation are identified as important open challenges for future radar based software research. To support future work in this field, the code developed for this thesis is made available open source on GitHub.

List of Figures

Figure 1.1: Subdivision of the perception task for autonomous vehicles. 2
Figure 2.1: Confusion matrix example. 9
Figure 2.2: Precision-recall curve example. 11
Figure 2.3: Comparison of IoU and distance metric for bounding box matching. Inspired by Kim et al. [84]. 11
Figure 2.4: Comparison of camera-based object detection pipelines. 14
Figure 2.5: Comparison of point-based and grid-based processing pipelines for lidar object detection.. 17
Figure 2.6: Exemplary radar signal processing chain, adapted from Engels et al. [73]. 19
Figure 2.7: Comparison of consecutive, probabilistic, and deep learning fusion pipelines. 23
Figure 3.1: Structure of the remainder of the thesis. 31
Figure 4.1: BEV of radar point cloud data and missing labels in the nuScenes data set. 34
Figure 4.2: Fusion of radar point cloud and camera data in a DNN. 55
Figure 4.3: Object detection result comparison using different input data in a DNN. ... 64
Figure 5.1: Object detection result comparison using different input data in a DNN at night. 84

List of Tables

Table 2.1: Overview of publicly available autonomous driving perception data sets. The table focuses on data sets which contain some form of radar data. 6
Table 2.2: Bounding box detection ordering for AP calculation. Inspired by Hui [82]..... 10
Table 2.3: Overview of related work most relevant to this thesis. 27
Table 4.1: Radar point cloud segmentation results. 34
Table 4.2: 2D object detection results with different input data. 56
Table 4.3: 3D object detection and tracking results..... 65

Bibliography

[1] M. Ryan, „The Future of Transportation: Ethical, Legal, Social and Economic Impacts of Self-driving Vehicles in the Year 2025," *Science and engineering ethics*, vol. 26, no. 3, pp. 1185–1208, 2020, DOI: 10.1007/s11948-019-00130-2.

[2] D. Bissell, T. Birtchnell, A. Elliott and E. L. Hsu, „Autonomous automobilities: The social impacts of driverless vehicles," *Current Sociology*, vol. 68, no. 1, pp. 116–134, 2020, DOI: 10.1177/0011392118816743.

[3] Y.-C. Lee and J. H. Mirman, „Parents' perspectives on using autonomous vehicles to enhance children's mobility," *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 415–431, 2018, DOI: 10.1016/j.trc.2018.10.001.

[4] K. Faber and D. van Lierop, „How will older adults use automated vehicles? Assessing the role of AVs in overcoming perceived mobility barriers," *Transportation Research Part A: Policy and Practice*, vol. 133, pp. 353–363, 2020, DOI: 10.1016/j.tra.2020.01.022.

[5] M. Ostrovsky and M. Schwarz, „Carpooling and the Economics of Self-Driving Cars," in *2019 ACM Conference on Economics and Computation*, 2019, pp. 581–582, DOI: 10.3386/w24349.

[6] L. M. Clements and K. M. Kockelman, „Economic Effects of Automated Vehicles," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2606, no. 1, pp. 106–114, 2017, DOI: 10.3141/2606-14.

[7] R. Lanctot, C. Ambrosio, H. Cohen and I. Riches. „Accelerating The Future: The Economic Impact Of The Emerging Passenger Economy," 2021. [Online]. Available: https://newsroom.intel.com/newsroom/wp-content/uploads/sites/11/2017/05/passenger-economy.pdf [visited on 05/17/2021].

[8] V. Haydin. „How Will Self Driving Cars Save Money?," 2021. [Online]. Available: https://www.intellias.com/self-driving-car-save-money/ [visited on 07/16/2021].

[9] N. A. Greenblatt, „Self-driving cars and the law," *IEEE Spectrum*, vol. 53, no. 2, pp. 46–51, 2016, DOI: 10.1109/MSPEC.2016.7419800.

[10] P. Lin, „Why Ethics Matters for Autonomous Cars," in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, J. C. Gerdes, B. Lenz and H. Winner, ed. Berlin, Heidelberg: Springer, 2016, pp. 69–85, DOI: 10.1007/978-3-662-48847-8_4.

[11] B. Templeton. „Waymo Peforms Embarrassingly In Construction Cone Situation," 2021. [Online]. Available: https://www.forbes.com/sites/bradtempleton/2021/05/14/waymo-peforms-embarrassingly-in-construction-cone-situation/ [visited on 05/17/2021].

[12] R. Stumpf. „Autopilot Blamed for Tesla's Crash Into Overturned Truck," 2021. [Online]. Available: https://www.thedrive.com/news/33789/autopilot-blamed-for-teslas-crash-into-overturned-truck [visited on 05/17/2021].

[13] B. H. Frank. „Uber parks its self-driving cars after fatal pedestrian crash in Tempe,“ 2021. [Online]. Available: https://venturebeat.com/2018/03/19/uber-parks-its-self-driving-cars-after-fatal-pedestrian-crash-in-tempe/ [visited on 05/17/2021].

[14] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus and M. Ang, „Perception, Planning, Control, and Coordination for Autonomous Vehicles,“ *Machines*, vol. 5, no. 1, p. 6, 2017.

[15] J. van Brummelen, M. O'Brien, D. Gruyer and H. Najjaran, „Autonomous vehicle perception: The technology of today and tomorrow,“ *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 384–406, 2018, DOI: 10.1016/j.trc.2018.02.012.

[16] S. Campbell, N. O'Mahony, L. Krpalcova, D. Riordan, J. Walsh, A. Murphy and C. Ryan, „Sensor Technology in Autonomous Vehicles : A review,“ *29th Irish Signals and Systems Conference*, pp. 1–4, 2018, DOI: 10.1109/ISSC.2018.8585340.

[17] T. Dahlström. „The road to everywhere: are HD maps for autonomous driving sustainable?,“ 2021. [Online]. Available: https://www.autonomousvehicleinternational.com/features/the-road-to-everywhere-are-hd-maps-for-autonomous-driving-sustainable.html [visited on 07/23/2021].

[18] R. P. D. Vivacqua, M. Bertozzi, P. Cerri, F. N. Martins and R. F. Vassallo, „Self-Localization Based on Visual Lane Marking Maps: An Accurate Low-Cost Approach for Autonomous Driving,“ *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 582–597, 2018, DOI: 10.1109/TITS.2017.2752461.

[19] M. R. U. Saputra, A. Markham and N. Trigoni, „Visual SLAM and Structure from Motion in Dynamic Environments: A Survey,“ *ACM Computing Surveys*, vol. 51, no. 2, pp. 1–36, 2018, DOI: 10.1145/3177853.

[20] F. Nobis, O. Papanikolaou, J. Betz and M. Lienkamp, „Persistent Map Saving for Visual Localization for Autonomous Vehicles: An ORB-SLAM 2 Extension,“ in *Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, Monaco, 2020, pp. 1–9, DOI: 10.1109/EVER48776.2020.9243094.

[21] G. Younes, D. Asmar, E. Shammas and J. Zelek, „Keyframe-based monocular SLAM: design, survey, and future directions,“ *Robotics and Autonomous Systems*, vol. 98, pp. 67–88, 2017, DOI: 10.1016/j.robot.2017.09.010.

[22] R. Mur-Artal and J. D. Tardos, „ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,“ *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017, DOI: 10.1109/TRO.2017.2705103.

[23] OXTS. „RT3000 v3,“ 2021. [Online]. Available: https://www.oxts.com/products/rt3000-v3/ [visited on 06/21/2021].

[24] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer and T. Koehler, „A system for traffic sign detection, tracking, and recognition using color, shape, and motion information,“ in *2005 IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, USA, 2005, pp. 255–260, DOI: 10.1109/IVS.2005.1505111.

[25] J. Cao, C. Song, S. Peng, F. Xiao and S. Song, „Improved Traffic Sign Detection and Recognition Algorithm for Intelligent Vehicles,“ *Sensors*, vol. 19, no. 18, 2019, DOI: 10.3390/s19184021.

[26] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing and C. Igel, „Detection of traffic signs in real-world images: The German traffic sign detection benchmark," in *2013 International Joint Conference on Neural Networks*, Dallas, TX, USA, 2013, pp. 1–8, DOI: 10.1109/IJCNN.2013.6706807.

[27] Z.-Q. Zhao, P. Zheng, S.-t. Xu and X. Wu, „Object Detection With Deep Learning: A Review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019, DOI: 10.1109/TNNLS.2018.2876865.

[28] P. N. Druzhkov and v.d. Kustikova, „A survey of deep learning methods and software tools for image classification and object detection," *Pattern Recognition and Image Analysis*, vol. 26, no. 1, pp. 9–15, 2016, DOI: 10.1134/S1054661816010065.

[29] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu and M. Pietikäinen, „Deep Learning for Generic Object Detection: A Survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020, DOI: 10.1007/s11263-019-01247-4.

[30] K. Riedl, S. Huber, M. Bomer, J. Kreibich, F. Nobis and J. Betz, „Importance of Contextual Information for the Detection of Road Damages," in *Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, Monaco, 2020, pp. 1–7, DOI: 10.1109/EVER48776.2020.9242954.

[31] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez and J. Garcia-Rodriguez, „A Review on Deep Learning Techniques Applied to Semantic Segmentation," *arXiv*, 2017. Available: https://arxiv.org/pdf/1704.06857.

[32] P. R Palafox, J. Betz, F. Nobis, K. Riedl and M. Lienkamp, „SemanticDepth: Fusing Semantic Segmentation and Monocular Depth Estimation for Enabling Autonomous Driving in Roads without Lane Lines," *Sensors*, vol. 19, no. 14, p. 3224, 2019, DOI: 10.3390/s19143224.

[33] E. Grilli, F. Menna and F. Remondino, „A review of point clouds segmentation and classification algorithms," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W3, pp. 339–344, 2017, DOI: 10.5194/isprs-archives-xlii-2-w3-339-2017.

[34] Y. He, H. Yu, X. Liu, Z. Yang, W. Sun, Y. Wang, Q. Fu, Y. Zou and A. Mian, „Deep Learning based 3D Segmentation: A Survey," *arXiv*, 2021. Available: https://arxiv.org/pdf/2103.05423.

[35] D. Feng, C. Haase-Schuetz, L. Rosenbaum, H. Hertlein, F. Duffhauss, C. Glaser, W. Wiesbeck and K. Dietmayer, „Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2021.

[36] A. Gupta, A. Anpalagan, L. Guan and A. S. Khwaja, „Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, 2021.

[37] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby and A. Mouzakitis, „A Survey on 3D Object Detection Methods for Autonomous Driving Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019, DOI: 10.1109/TITS.2019.2892405.

[38] A. Ennajar, N. Khouja, R. Boutteau and F. Tlili, „Deep Multi-modal Object Detection for Autonomous Driving," in *18th International Multi-Conference on Systems, Signals & Devices (SSD)*, Monastir, Tunisia, 2021, pp. 7–11, DOI: 10.1109/SSD52085.2021.9429355.

[39] A. Geiger, P. Lenz and R. Urtasun, „Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. Available: http://www.cvlibs.net/datasets/kitti/.

[40] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan and O. Beijbom, „nuScenes: A Multimodal Dataset for Autonomous Driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 11618–11628, DOI: 10.1109/CVPR42600.2020.01164.

[41] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt and S. Thrun, „Junior: The Stanford Entry in the Urban Challenge," in *The DARPA Urban Challenge: Autonomous vehicles in city traffic* (Springer Tracts in Advanced Robotics). vol. 56, M. Buehler, ed. Berlin: Springer, 2009, pp. 91–123, DOI: 10.1007/978-3-642-03991-1_3.

[42] F. Petit. „MEMS mirrors enable LiDAR sensors for the mass market," 2021. [Online]. Available: https://www.blickfeld.com/blog/lidar-for-the-mass-market/ [visited on 07/21/2021].

[43] C. Sammut and G. I. Webb, *Encyclopedia of machine learning: With 78 tables*, New York, NY, Springer, 2011, DOI: 10.1007/978-0-387-30164-8.

[44] T. M. Mitchell, *Machine Learning*, New York, McGraw-Hill, 1997, ISBN: 0070428077.

[45] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.

[46] F. Chollet, *Deep Learning with Python*, Manning, 2017, ISBN: 9781617294433.

[47] F. Tung, J. Chen, L. Meng and J. J. Little, „The Raincouver Scene Parsing Benchmark for Self-Driving in Adverse Weather and at Night," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2188–2193, 2017.

[48] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo and M. A. Kaafar, „The Impact of Adverse Weather Conditions on Autonomous Vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car," *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 103–111, 2019, DOI: 10.1109/MVT.2019.2892497.

[49] L. Daniel, D. Phippen, E. Hoare, A. Stove, M. Cherniakov and M. Gashinova, „Low-THz Radar, Lidar and Optical Imaging through Artificially Generated Fog," in *2017 International Conference on Radar Systems*, Belfast, UK, 2017, DOI: 10.1049/cp.2017.0369.

[50] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer and F. Heide, „Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020.

[51] Z. Liu, Y. Cai, H. Wang, L. Chen, H. Gao, Y. Jia and Y. Li, „Robust Target Recognition and Tracking of Self-Driving Cars With Radar and Camera Information Fusion Under Severe Weather Conditions," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2021, DOI: 10.1109/TITS.2021.3059674.

[52] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt and B. Sick, „An Application-Driven Conceptualization of Corner Cases for Perception in Highly Automated Driving," *arXiv*, 2021. Available: https://arxiv.org/pdf/2103.03678.

[53] E. R. Corral-Soto and L. Bingbing, „Understanding Strengths and Weaknesses of Complementary Sensor Modalities in Early Fusion for Object Detection," in *2020 IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, USA, 2020, pp. 1785–1792, DOI: 10.1109/IV47402.2020.9304558.

[54] D. Teich. „Good data quality for machine learning is an analytics must," 2021. [Online]. Available: https://searchdatamanagement.techtarget.com/tip/Good-data-quality-for-machine-learning-is-an-analytics-must [visited on 07/21/2021].

[55] J. Tan. „How to improve data quality for machine learning?," 2021. [Online]. Available: https://towardsdatascience.com/how-to-improve-data-preparation-for-machine-learning-dde107b60091 [visited on 07/21/2021].

[56] Google. „Dataset Search," 2021. [Online]. Available: https://datasetsearch.research.google.com/search?query=%20automotive%20object%20detection [visited on 07/23/2021].

[57] M.-F. Chang, D. Ramanan, J. Hays, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr and S. Lucey, „Argoverse: 3D Tracking and Forecasting With Rich Maps," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 8740–8749, DOI: 10.1109/CVPR.2019.00895.

[58] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang and V. Shet. „Lyft Level 5 AV Dataset 2019," 2021. [Online]. Available: https://level5.lyft.com/dataset/ [visited on 07/16/2021].

[59] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis and P. Schuberth, „A2D2: Audi Autonomous Driving Dataset," *arXiv*, 2020. Available: https://arxiv.org/pdf/2004.06320.

[60] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen and D. Anguelov, „Scalability in Perception for Autonomous Driving: Waymo Open Dataset," *arXiv*, 2020. Available: https://arxiv.org/abs/1912.04838.

[61] Z. Yan, L. Sun, T. Krajnik and Y. Ruichek, „EU Long-term Dataset with Multiple Sensors for Autonomous Driving," *arXiv*, 2019. Available: https://arxiv.org/pdf/1909.03330.

[62] D. Barnes, M. Gadd, P. Murcutt, P. Newman and I. Posner, „The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, DOI: 10.1109/ICRA40945.2020.

[63] G. Kim, Y. S. Park, Y. Cho, J. Jeong and A. Kim, „MulRan: Multimodal Range Dataset for Urban Place Recognition," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, DOI: 10.1109/ICRA40945.2020.

[64] A. Mimouna, I. Alouani, A. B. Khalifa, Y. El Hillali, A. Taleb-Ahmed, A. Menhaj, A. Ouahabi and N. Essoukri Ben Amara, „OLIMP: A Heterogeneous Multimodal Dataset for Advanced Environment Perception," *Electronics*, vol. 9, no. 4, 2020.

[65] A. Ouaknine, A. Newson, J. Rebut, F. Tupin and P. Pérez, „CARRADA Dataset: Camera and Automotive Radar with Range-Angle-Doppler Annotations," *arXiv*, 2020. Available: https://arxiv.org/pdf/2005.01456.

[66] M. Sheeny, E. D. Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang and A. Wallace, „RADIATE: A Radar Dataset for Automotive Perception in Bad Weather," *arXiv*, 2020. Available: http://arxiv.org/pdf/2010.09076v3.

[67] University of Washington. „CRUW Dataset," 2021. [Online]. Available: https://www.cruwdataset.org/home [visited on 07/16/2021].

[68] T. Y. Lim, S. Markowitz and M. N. Do, „RaDICaL: A Synchronized FMCW Radar, Depth, IMU and RGB Camera Data Dataset with Low-Level FMCW Radar Signals," *IEEE Journal of Selected Topics in Signal Processing*, 2021, DOI: 10.1109/JSTSP.2021.3061270.

[69] M. Meyer and G. Kuschk, „Automotive Radar Dataset for Deep Learning Based 3D Object Detection," in *2019 16th European Radar Conference (EuRAD)*, 2019, pp. 129–132.

[70] O. Schumann, M. Hahn, N. Scheiner, F. Weishaupt, J. F. Tilly, J. Dickmann and C. Wöhler, „RadarScenes: A Real-World Radar Point Cloud Data Set for Automotive Applications," *arXiv*, 2021. Available: http://arxiv.org/pdf/2104.02493v1.

[71] N. Scheiner, O. Schumann, F. Kraus, N. Appenrodt, J. Dickmann and B. Sick, „Off-the-shelf sensor vs. experimental radar – How much resolution is necessary in automotive radar classification?," *arXiv*, 2020. Available: https://arxiv.org/abs/2006.05485.

[72] F. Nobis, F. Fent, J. Betz and M. Lienkamp, „Kernel Point Convolution LSTM Networks for Radar Point Cloud Segmentation," *Applied Sciences*, vol. 11, no. 6, p. 2599, 2021, DOI: 10.3390/app11062599.

[73] F. Engels, P. Heidenreich, M. Wintermantel, L. Stacker, M. Al Kadi and A. M. Zoubir, „Automotive Radars Signal Processing: Research Directions and Practical Challenges," *IEEE Journal of Selected Topics in Signal Processing*, 2021, DOI: 10.1109/JSTSP.2021.3063666.

[74] Continental. „ARS 408-21 Long Range Radar Sensor 77 GHz," 2021. [Online]. Available: https://conti-engineering.com/wp-content/uploads/2020/02/ARS-408-21_EN_HS-1.pdf [visited on 07/16/2021].

[75] M. Geisslinger, „Autonomous Driving: Object Detection using Neural Networks for Radar and Camera Sensor Fusion," Master's Thesis, TU Munich, Munich, 2019.

[76] motional. „nuScenes: Data annotation," 2021. [Online]. Available: https://www.nuscenes.org/nuscenes#data-annotation [visited on 07/21/2021].

[77] Google. „Google scholar nuScenes citation count," 2021. [Online]. Available: https://scholar.google.de/citations?view_op=view_citation&hl=de&user=373LKEYAAAAJ&citation_for_view=373LKEYAAAAJ:roLk4NBRz8UC [visited on 08/05/2021].

[78] N. Tishby and N. Zaslavsky, „Deep learning and the information bottleneck principle," in *2015 IEEE Information Theory Workshop*, Jerusalem, Israel, 2015, pp. 1–5, DOI: 10.1109/ITW.2015.7133169.

[79] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf and G.-Z. Yang, „XAI-Explainable artificial intelligence," *Science Robotics*, vol. 4, no. 37, 2019, DOI: 10.1126/scirobotics.aay7120.

[80] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel and U. Muller, „Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car," *arXiv*, 2017. Available: https://arxiv.org/pdf/1704.07911.

[81] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller and W. Samek, „On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation," *PloS one*, vol. 10, no. 7, 2015, DOI: 10.1371/journal.pone.0130140.

[82] J. Hui. „mAP (mean Average Precision) for Object Detection," 2021. [Online]. Available: https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173 [visited on 06/21/2021].

[83] nuScenes. „mAP Definition," 2021. [Online]. Available: https://www.nuscenes.org/object-detection?externalData=all&mapData=all&modalities=Any#:~:text=number%20of%20points.-,Average%20Precision%20metric,-mean%20Average%20Precision [visited on 07/23/2021].

[84] J. Kim, Y. Kim and D. Kum, „Low-level Sensor Fusion Network for 3D Vehicle Detection using Radar Range-Azimuth Heatmap and Monocular Image," in *15th Asian Conference on Computer Vision (ACCV)*, Kyoto, Japan, 2020.

[85] J. Effertz, „Autonome Fahrzeugführung in urbaner Umgebung durch Kombination objekt- und kartenbasierter Umfeldmodelle," Dissertation, Technische Universität Braunschweig, Braunschweig, 2009.

[86] S. Sengupta, S. Basak, P. Saikia, S. Paul, V. Tsalavoutis, F. Atiah, V. Ravi and A. Peters, „A Review of Deep Learning with Special Emphasis on Architectures, Applications and Recent Trends," *arXiv*, 2019. Available: https://arxiv.org/abs/1905.13294.

[87] D. Griffiths and J. Boehm, „A review on deep learning techniques for 3D sensed data classification," *arXiv*, 2019. Available: https://arxiv.org/pdf/1907.04444.

[88] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. S. Hong, S.-H. Han and I. S. Kweon, „VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition," in *2017 IEEE International Conference on Computer Vision*, Venice, 2017, pp. 1965–1973, DOI: 10.1109/ICCV.2017.215.

[89] F. Schneemann and P. Heinemann, „Context-based Detection of Pedestrian Crossing Intention for Autonomous Driving in Urban Environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2243–2248, DOI: 10.1109/IROS.2016.7759351.

[90] F. Nobis, F. Brunhuber, S. Janssen, J. Betz and M. Lienkamp, „Exploring the Capabilities and Limits of 3D Monocular Object Detection - A Study on Simulation and Real World Data," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems*, Rhodes, 2020, pp. 1–8, DOI: 10.1109/ITSC45102.2020.9294625.

[91] R. Fan, L. Wang, M. J. Bocus and I. Pitas, „Computer Stereo Vision for Autonomous Driving," *arXiv*, 2020. Available: https://arxiv.org/pdf/2012.03194.

[92] Y. Chen, W. L. Cai, X. J. Zou, D. F. Xu and T. H. Liu, „A Research of Stereo Vision Positioning under Vibration," *Applied Mechanics and Materials*, vol. 44-47, pp. 1315–1319, 2010, DOI: 10.4028/www.scientific.net/AMM.44-47.1315.

[93] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, „ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, Barcelona, Spain, 2011, pp. 2564–2571, DOI: 10.1109/ICCV.2011.6126544.

[94] N. Dalal and B. Triggs, „Histograms of Oriented Gradients for Human Detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005, pp. 886–893, DOI: 10.1109/CVPR.2005.177.

[95] J. Canny, „A Computational Approach to Edge Detection," *IEEE transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986, DOI: 10.1109/TPAMI.1986.4767851.

[96] J. Ku, A. Harakeh and S. L. Waslander, „In Defense of Classical Image Processing: Fast Depth Completion on the CPU," *arXiv*, 2018. Available: https://arxiv.org/abs/1802.00036.

[97] A. Krizhevsky, I. Sutskever and G. E. Hinton, „ImageNet Classification with Deep Convolutional Neural Networks," in *25th International Conference on Neural Information Processing Systems - Volume 1*, Lake Tahoe, NV, USA, 2012, pp. 1097–1105.

[98] H. R. „Success story on Imagenet CNNs," 2021. [Online]. Available: https://medium.com/@harishr2301/sucess-story-on-imagenet-cnns-6e8ccc5f1d19 [visited on 07/23/2021].

[99] S. Hochreiter and J. Schmidhuber, „Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997, DOI: 10.1162/neco.1997.9.8.1735.

[100] K. Simonyan and A. Zisserman, „Very Deep Convolutional Networks for Large-Scale Image Recognition," *17th International Conference on Learning Representations*, 2015.

[101] K. He, X. Zhang, S. Ren and J. Sun, „Deep Residual Learning for Image Recognition," in *2016 29th IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778, DOI: 10.1109/CVPR.2016.90.

[102] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, „Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. Available: http://jmlr.org/papers/v15/srivastava14a.html.

[103] S. Ren, K. He, R. Girshick and J. Sun, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *28th International Conference on Neural Information Processing Systems - Volume 1*, Montreal, Canada, 2015, pp. 91–99.

[104] R. Girshick, J. Donahue, T. Darrell and J. Malik, „Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *arXiv*, 2013. Available: https://arxiv.org/pdf/1311.2524.

[105] R. Girshick, „Fast R-CNN," *arXiv*, 2015. Available: https://arxiv.org/pdf/1504.08083.

[106] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, „OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," *arXiv*, 2013. Available: https://arxiv.org/pdf/1312.6229.

[107] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, „You Only Look Once: Unified, Real-Time Object Detection," in *2016 29th IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016.

[108] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, „SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision*, vol. 9905, pp. 21–37, 2016.

[109] K. He, G. Gkioxari, P. Dollár and R. Girshick, „Mask R-CNN," *arXiv*, 2017. Available: https://arxiv.org/pdf/1703.06870.

[110] B. Hariharan, P. Arbeláez, R. Girshick and J. Malik, „Simultaneous Detection and Segmentation," *arXiv*, 2014. Available: http://arxiv.org/pdf/1407.1808v1.

[111] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan and S. J. Belongie, „Feature Pyramid Networks for Object Detection," *arXiv*, 2016. Available: https://arxiv.org/abs/1612.03144.

[112] T.-Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, „Focal Loss for Dense Object Detection," *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[113] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama and K. Murphy, „Speed/accuracy trade-offs for modern convolutional object detectors," *arXiv*, 2016. Available: http://arxiv.org/pdf/1611.10012v3.

[114] 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," *arXiv*, 2017. Available: https://arxiv.org/pdf/1704.04861.

[115] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, „Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, 2014, pp. 740–755.

[116] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan and L. Zhang, „Dynamic Head: Unifying Object Detection Heads With Attentions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7373–7382.

[117] C. R. Qi, W. Liu, C. Wu, H. Su and L. J. Guibas, „Frustum PointNets for 3D Object Detection from RGB-D Data," *arXiv*, 2017. Available: https://arxiv.org/pdf/1711.08488.

[118] Z. Qin, J. Wang and Y. Lu, „MonoGRNet: A Geometric Reasoning Network for Monocular 3D Object Localization," *arXiv*, 2018. Available: https://arxiv.org/pdf/1811.10247.

[119] E. Jörgensen, C. Zach and F. Kahl, „Monocular 3D Object Detection and Box Fitting Trained End-to-End Using Intersection-over-Union Loss," *arXiv*, 2019. Available: http://arxiv.org/pdf/1906.08070v2.

[120] A. Mousavian, D. Anguelov, J. Flynn and J. Kosecka, „3D Bounding Box Estimation Using Deep Learning and Geometry," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[121] A. Simonelli, S. Rota Bulò, L. Porzi, M. López-Antequera and P. Kontschieder, „Disentangling Monocular 3D Object Detection," *arXiv*, 2019. Available: https://arxiv.org/abs/1905.12365.

[122] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler and R. Urtasun, „Monocular 3D Object Detection for Autonomous Driving," in *2016 29th IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 2147–2156, DOI: 10.1109/CVPR.2016.236.

[123] C. Godard, O. M. Aodha and G. Brostow, „Digging Into Self-Supervised Monocular Depth Estimation," *arXiv*, 2018. Available: https://arxiv.org/abs/1806.01260.

[124] M. Ding, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu and P. Luo, „Learning Depth-Guided Convolutions for Monocular 3D Object Detection," *arXiv*, 2019. Available: https://arxiv.org/pdf/1912.04799.

[125] F. Manhardt, W. Kehl and A. Gaidon, „ROI-10D: Monocular Lifting of 2D Detection to 6D Pose and Metric Shape," *arXiv*, 2018. Available: https://arxiv.org/pdf/1812.02781.

[126] 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," *arXiv*, 2018. Available: https://arxiv.org/pdf/1812.07179.

[127] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell and K. Q. Weinberger, „Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving," *arXiv*, 2019. Available: https://arxiv.org/pdf/1906.06310.

[128] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang and X. Fan, „Accurate Monocular 3D Object Detection via Color-Embedded 3D Reconstruction for Autonomous Driving," in *2019 IEEE International Conference on Computer Vision*, 2019, pp. 6851–6860.

[129] J. Chang and G. Wetzstein, „Deep Optics for Monocular Depth Estimation and 3D Object Detection," *arXiv*, 2019. Available: https://arxiv.org/abs/1904.08601.

[130] Y. Zhang, J. Lu and J. Zhou, „Objects Are Different: Flexible Monocular 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3289–3298.

[131] W. Zheng, W. Tang, L. Jiang and C.-W. Fu, „SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14494–14503.

[132] J. M. U. Vianney, S. Aich and B. Liu, „RefinedMPL: Refined Monocular PseudoLiDAR for 3D Object Detection in Autonomous Driving," *arXiv*, 2019. Available: https://arxiv.org/pdf/1911.09712.

[133] Y. Chen, S. Liu, X. Shen and J. Jia, „DSGN: Deep Stereo Geometry Network for 3D Object Detection," *arXiv*, 2020. Available: https://arxiv.org/abs/2001.03398.

[134] D. Garg, Y. Wang, B. Hariharan, M. Campbell, K. Weinberger and W.-L. Chao, „Wasserstein Distances for Stereo Disparity Estimation," in *34th Conference on Neural Information Processing Systems*, 2020.

[135] G. L. Heritage and A. R. G. Large, *Laser Scanning for the Environmental Sciences*, Chichester, U.K and Hoboken, N.J, Wiley-Blackwell, 2009, DOI: 10.1002/9781444311952.

[136] Y. Li and J. Ibanez-Guzman, „Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020, DOI: 10.1109/msp.2020.2973615.

[137] S. Bianco, R. Cadène, L. Celona and P. Napoletano, „Benchmark Analysis of Representative Deep Neural Network Architectures," *arXiv*, 2018. Available: https://arxiv.org/abs/1810.00736.

[138] M. Hebel, M. Hammer, M. Arens and A. L. Diehm, „Mitigation of crosstalk effects in multi-LiDAR configurations," in *Electro-optical remote sensing XII: 12-13 September 2018*, Berlin, Germany, 2018, DOI: 10.1117/12.2324305.

[139] G. Kim, J. Eom and Y. Park, „Investigation on the occurrence of mutual interference between pulsed terrestrial LIDAR scanners," in *2015 IEEE Intelligent Vehicles Symposium*, Seoul, South Korea, 2015, pp. 437–442, DOI: 10.1109/IVS.2015.7225724.

[140] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, „A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Second International Conference on Knowledge Discovery and Data Mining*, Portland, OR, USA, 1996, pp. 226–231.

[141] B. Kim, B. Choi, M. Yoo, H. Kim and E. Kim, „Robust object segmentation using a multi-layer laser scanner," *Sensors*, vol. 14, no. 11, pp. 20400–20418, 2014.

[142] T. N. Johansson and O. Wellenstam, „LiDAR Clustering and Shape Extraction for Automotive Applications," Master's Thesis, Chalmers University of Technology, Gothenburg, Sweden, 2017.

[143] M. E. Bouzouraa and U. Hofmann, „Fusion of occupancy grid mapping and model based object tracking for driver assistance systems using laser and radar sensors," in *2010 IEEE Intelligent Vehicles Symposium*, La Jolla, CA, USA, 2010, pp. 294–300, DOI: 10.1109/IVS.2010.5548106.

[144] A. Asvadi, C. Premebida, P. Peixoto and U. Nunes, „3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes," *Robotics and Autonomous Systems*, vol. 83, pp. 299–311, 2018.

[145] C. Coué, C. Pradalier, C. Laugier, T. Fraichard and P. Bessière, „Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, 2016, DOI: 10.1177/0278364906061158.

[146] H. Najdataei, Y. Nikolakopoulos, V. Gulisano and M. Papatriantafilou, „Lisco: A Continuous Approach in LiDAR Point-cloud Clustering," *arXiv*, 2017. Available: https://arxiv.org/abs/1711.01853.

[147] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao and Y. Chao, „The connected-component labeling problem: A review of state-of-the-art algorithms," *Pattern Recognition*, vol. 70, pp. 25–43, 2017, DOI: 10.1016/j.patcog.2017.04.018.

[148] C. R. Qi, H. Su, K. Mo and L. J. Guibas, „PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[149] C. R. Qi, L. Yi, H. Su and L. J. Guibas, „PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," *arXiv*, 2017. Available: https://arxiv.org/pdf/1706.02413.

[150] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette and L. J. Guibas, „KPConv: Flexible and Deformable Convolution for Point Clouds," *arXiv*, 2019. Available: https://arxiv.org/pdf/1904.08889.

[151] Y. Yan, Y. Mao and B. Li, „SECOND: Sparsely Embedded Convolutional Detection," *Sensors*, vol. 18, no. 10, 2018.

[152] B. Graham, M. Engelcke and L. van der Maaten, „3D Semantic Segmentation with Submanifold Sparse Convolutional Networks," *arXiv*, 2017. Available: https://arxiv.org/pdf/1711.10275.

[153] Z. Liang, M. Zhang, Z. Zhang, X. Zhao and S. Pu, „RangeRCNN: Towards Fast and Accurate 3D Object Detection with Range Image Representation," *arXiv*, 2020. Available: https://arxiv.org/pdf/2009.00206.

[154] Y. Zhou and O. Tuzel, „VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4490–4499, 2018.

[155] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom, „PointPillars: Fast Encoders for Object Detection from Point Clouds," *arXiv*, 2019. Available: https://arxiv.org/abs/1812.05784.

[156] S. Shi, X. Wang and H. Li, „PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud," *arXiv*, 2018. Available: https://arxiv.org/abs/1812.04244.

[157] S. Shi, Z. Wang, J. Shi, X. Wang and H. Li, „From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network," *arXiv*, 2019. Available: https://arxiv.org/pdf/1907.03670.

[158] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang and H. Li, „PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," *arXiv*, 2019. Available: https://arxiv.org/pdf/1912.13192.

[159] T. Yin, X. Zhou and P. Krähenbühl, „Center-based 3D Object Detection and Tracking," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11784–11793, 2021.

[160] M. C. Potter, B. Wyble, C. E. Hagmann and E. S. McCourt, „Detecting meaning in RSVP at 13 ms per picture," *Attention, Perception, & Psychophysics*, vol. 76, no. 2, pp. 270–279, 2014, DOI: 10.3758/s13414-013-0605-z.

[161] H. Fan and Y. Yang, „PointRNN: Point Recurrent Neural Network for Moving Point Cloud Processing," vol. arXiv, 2019. Available: https://arxiv.org/pdf/1910.08287.

[162] A. Borst and M. Egelhaaf, „Principles of visual motion detection," *Trends in Neurosciences*, vol. 12, no. 8, pp. 297–306, 1989, DOI: 10.1016/0166-2236(89)90010-6.

[163] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao and L. Xiao, „A review of algorithms for filtering the 3D point cloud," *Signal Processing: Image Communication*, vol. 57, pp. 103–112, 2017, DOI: 10.1016/j.image.2017.05.009.

[164] N. Giordano, *College Physics: Reasoning and Relationships*, Brooks/Cole, 2010, ISBN: 978-0-534-42471-8.

[165] N. Yamada, Y. Tanaka and K. Nishikawa, „Radar cross section for pedestrian in 76GHz band," in *2005 European Microwave Conference*, Paris, France, 2005, DOI: 10.1109/EUMC.2005.1610101.

[166] M. Yasugi, Y. Cao, K. Kobayashi, T. Morita, T. Kishigami and Y. Nakagawa, „79GHz-band radar cross section measurement for pedestrian detection," in *2013 Asia-Pacific Microwave Conference*, Seoul, South Korea, 2013, pp. 576–578, DOI: 10.1109/APMC.2013.6694869.

[167] H. Winner, „Radarsensorik," in *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*, H. Winner, S. Hakuli and G. Wolf, ed. Wiesbaden: Vieweg+Teubner, 2009, pp. 123–171, DOI: 10.1007/978-3-8348-9977-4_13.

[168] S. Brisken, F. Ruf and F. Höhne, „Recent evolution of automotive imaging radar and its information content," *IET Radar, Sonar & Navigation*, vol. 12, no. 10, pp. 1078–1081, 2018, DOI: 10.1049/iet-rsn.2018.0026.

[169] S. J. Julier and J. K. Uhlmann, „A New Extension of the Kalman Filter to Nonlinear Systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, Orlando, FL, USA, 1997, p. 182, DOI: 10.1117/12.280797.

[170] J. Munkres, „Algorithms for the Assignment and Transportation Problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957, DOI: 10.1137/0105003.

[171] Y. Bar-Shalom, F. Daum and J. Huang, „The probabilistic data association filter," *IEEE Control Systems*, vol. 29, no. 6, pp. 82–100, 2009, DOI: 10.1109/MCS.2009.934469.

[172] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick and I. Reid, „Joint Probabilistic Data Association Revisited," in *2015 IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 3047–3055, DOI: 10.1109/ICCV.2015.349.

[173] A. Yilmaz, O. Javed and M. Shah, „Object tracking: A survey," *ACM Computing Surveys*, vol. 38, no. 4, p. 13, 2006, DOI: 10.1145/1177352.1177355.

[174] M. Lundgren, „Bayesian filtering for automotive applications," Dissertation, Chalmers University of Technology, Gothenburg, Sweden, 2015.

[175] E. Schubert, F. Meinl, M. Kunert and W. Menzel, „Clustering of High Resolution Automotive Radar Detections and Subsequent Feature Extraction for Classification of Road Users," *16th International Radar Symposium (IRS)*, pp. 174–179, 2015.

[176] M. Li, M. Stolz, Z. Feng, M. Kunert, R. Henze and F. Kucukay, „An Adaptive 3D Grid-Based Clustering Algorithm for Automotive High Resolution Radar Sensor," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES): 12-14 Sept. 2018*, Madrid, 2018, pp. 1–7, DOI: 10.1109/ICVES.2018.8519483.

[177] D. Kellner, J. Klappstein and K. Dietmayer, „Grid-based DBSCAN for clustering extended objects in radar data," in *2012 IEEE Intelligent Vehicles Symposium*, Madrid, Spain, 2012, pp. 365–370, DOI: 10.1109/IVS.2012.6232167.

[178] M. Stolz, M. Li, Z. Feng, M. Kunert and W. Menzel, „High Resolution Automotive Radar Data Clustering with Novel Cluster Method," *2018 IEEE Radar Conference*, 2018.

[179] Y. Xia, P. Wang, K. O. E. Berntorp, L. Svensson, K. Granstrom, H. Mansour, P. T. Boufounos and P. Orlik, „Learning-based Extended Object Tracking Using Hierarchical Truncation Measurement Model with Automotive Radar," *IEEE Journal of Selected Topics in Signal Processing*, p. 1, 2021, DOI: 10.1109/JSTSP.2021.3058062.

[180] J. Lombacher, M. Hahn, J. Dickmann and C. Wohler, „Potential of radar for static object classification using deep learning methods," in *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, San Diego, CA, USA, 2016, pp. 1–4, DOI: 10.1109/ICMIM.2016.7533931.

[181] J. Lombacher, M. Hahn, J. Dickmann and C. Wohler, „Object classification in radar using ensemble methods," in *2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, Nagoya, Japan, 2017, pp. 87–90, DOI: 10.1109/ICMIM.2017.7918863.

[182] O. Schumann, C. Wohler, M. Hahn and J. Dickmann, „Comparison of Random Forest and Long Short-Term Memory Network Performances in Classification Tasks Using Radar," in *2017 Symposium on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, 2017, pp. 1–6, DOI: 10.1109/SDF.2017.8126350.

[183] F. J. Abdu, Y. Zhang, M. Fu, Y. Li and Z. Deng, „Application of Deep Learning on Millimeter-Wave Radar Signals: A Review," *Sensors*, vol. 21, no. 6, 2021.

[184] P. Lang, X. Fu, M. Martorella, J. Dong, R. Qin, X. Meng and M. Xie, „A Comprehensive Survey of Machine Learning Applied to Radar Signal Processing," *arXiv*, 2020. Available: https://arxiv.org/pdf/2009.13702.

[185] Y. Cheng, J. Su, H. Chen and Y. Liu, „A New Automotive Radar 4D Point Clouds Detector by Using Deep Learning," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021, pp. 8398–8402, DOI: 10.1109/ICASSP39728.2021.9413682.

[186] H. Rohling, „Radar CFAR Thresholding in Clutter and Multiple Target Situations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-19, no. 4, pp. 608–621, 1983, DOI: 10.1109/TAES.1983.309350.

[187] A. Stroescu, L. Daniel, D. Phippen, M. Cherniakov and M. Gashinova, „Object Detection on Radar Imagery for Autonomous Driving Using Deep Neural Networks," *2020 17th European Radar Conference (EuRAD)*, 2021.

[188] B. Major, D. Fontijne, A. Ansari, R. Teja Sukhavasi, R. Gowaikar, M. Hamilton, S. Lee, S. Grzechnik and S. Subramanian, „Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors," in *2019 IEEE International Conference on Computer Vision*, 2019.

[189] S. Azam, F. Munir and M. Jeon, „Channel Boosting Feature Ensemble for Radar-based Object Detection," *arXiv*, 2021. Available: https://arxiv.org/pdf/2101.03531.

[190] M. Dimitrievski, I. Shopovska, D. van Hamme, P. Veelaert and W. Philips, „Weakly Supervised Deep Learning Method for Vulnerable Road User Detection in FMCW Radar," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems*, Rhodes, 2020, pp. 1–8, DOI: 10.1109/ITSC45102.2020.9294399.

[191] A. Palffy, J. Dong, J. F. P. Kooij and D. M. Gavrila, „CNN based Road User Detection using the 3D Radar Cube," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1263–1270, 2020, DOI: 10.1109/LRA.2020.2967272.

[192] O. Schumann, M. Hahn, J. Dickmann and C. Wohler, „Semantic Segmentation on Radar Point Clouds," in *2018 21st International Conference on Information Fusion (FUSION)*, Cambridge, 2018, pp. 2179–2186, DOI: 10.23919/ICIF.2018.8455344.

[193] A. Danzer, T. Griebel, M. Bach and K. Dietmayer, „2D Car Detection in Radar Data with PointNets," *arXiv*, 2019. Available: https://arxiv.org/abs/1904.08414.

[194] M. Dreher, E. Ercelik, T. Banziger and A. Knol, „Radar-based 2D Car Detection Using Deep Neural Networks," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems*, Rhodes, 2020, pp. 1–8, DOI: 10.1109/ITSC45102.2020.9294546.

[195] O. Schumann, J. Lombacher, M. Hahn, C. Wohler and J. Dickmann, „Scene Understanding With Automotive Radar," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 2, pp. 188–203, 2020, DOI: 10.1109/TIV.2019.2955853.

[196] P. Svenningsson, F. Fioranelli and A. Yarovoy, „Radar-PointGNN: Graph Based Object Recognition for Unstructured Radar Point-cloud Data," in *2021 IEEE Radar Conference*, Atlanta, GA, USA, 2021, pp. 1–6, DOI: 10.1109/RadarConf2147009.2021.9455172.

[197] H. P. Moravec, „Sensor Fusion in Certainty Grids for Mobile Robots," *AI Magazine*, vol. 9, no. 2, pp. 61–74, 1988.

[198] R. E. Kalman, „A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[199] S. P. Chaudhuri and S. Das, „Neural networks for data fusion," in *IEEE International Conference on Systems Engineering*, Pittsburgh, PA, USA, 1990, pp. 327–330, DOI: 10.1109/ICSYSE.1990.203163.

[200] K. Liu, Y. Li, N. Xu and P. Natarajan, „Learn to Combine Modalities in Multimodal Deep Learning," *arXiv*, 2018. Available: https://arxiv.org/abs/1805.11730.

[201] J. Fayyad, M. A. Jaradat, D. Gruyer and H. Najjaran, „Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review," *Sensors*, vol. 20, no. 15, 2020, DOI: 10.3390/s20154220.

[202] G. Velasco-Hernandez, D. J. Yeong, J. Barry and J. Walsh, „Autonomous Driving Architectures, Perception and Data Fusion: A Review," in *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, 2020, pp. 315–321, DOI: 10.1109/ICCP51029.2020.9266268.

[203] D. J. Yeong, G. Velasco-Hernandez, J. Barry and J. Walsh, „Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review," *Sensors*, vol. 21, no. 6, 2021, DOI: 10.3390/s21062140.

[204] Y. Cui, R. Chen, W. Chu, L. Chen, D. Tian, Y. Li and D. Cao, „Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–18, 2021, DOI: 10.1109/TITS.2020.3023541.

[205] S. Han, X. Wang, L. Xu, H. Sun and N. Zheng, „Frontal object perception for Intelligent Vehicles based on radar and camera fusion," in *35th Chinese Control Conference*, Chengdu, China, 2016, pp. 4003–4008, DOI: 10.1109/ChiCC.2016.7553978.

[206] R. Nabati and H. Qi, „RRPN: Radar Region Proposal Network for Object Detection in Autonomous Vehicles," *arXiv*, 2019. Available: https://arxiv.org/pdf/1905.00526.

[207] X. Zhang, M. Zhou, P. Qiu, Y. Huang and J. Li, „Radar and vision fusion for the real-time obstacle detection and identification," *Industrial Robot*, vol. 2007, no. 2, p. 233, 2019, DOI: 10.1108/IR-06-2018-0113.

[208] H. Jha, V. Lodhi and D. Chakravarty, „Object Detection and Identification Using Vision and Radar Data Fusion System for Ground-Based Navigation," in *6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 2019, pp. 590–593, DOI: 10.1109/SPIN.2019.8711717.

[209] S. Matzka and R. Altendorfer, „A comparison of track-to-track fusion algorithms for automotive sensor fusion," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, 2008, pp. 189–194, DOI: 10.1109/MFI.2008.4648063.

[210] S.-L. Sun and Z.-L. Deng, „Multi-sensor optimal information fusion Kalman filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, 2004, DOI: 10.1016/j.automatica.2004.01.014.

[211] T. Fortmann, Y. Bar-Shalom and M. Scheffe, „Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983, DOI: 10.1109/JOE.1983.1145560.

[212] K.-E. Kim, C.-L. Lee, D.-S. Pac and M.-T. Lim, „Sensor Fusion for Vehicle Tracking with Camera and Radar Sensor," *17th International Conference on Control, Automation and Systems (ICCAS)*, 2017.

[213] K.-H. Lee, Y. Kanzawa, M. Derry and M. R. James, „Multi-Target Track-to-Track Fusion Based on Permutation Matrix Track Association," in *2018 IEEE Intelligent Vehicles Symposium*, Changshu, 2018, pp. 465–470, ISBN: 978-1-5386-4452-2. DOI: 10.1109/IVS.2018.8500433.

[214] K. Gadzicki, R. Khamsehashari and C. Zetzsche, „Early vs Late Fusion in Multimodal Convolutional Neural Networks," in *2020 23rd International Conference on Information Fusion (FUSION)*, Rustenburg, South Africa, 2020, pp. 1–6, DOI: 10.23919/FUSION45008.2020.9190246.

[215] S. Chadwick, W. Maddern and P. Newman, „Distant Vehicle Detection Using Radar and Vision," *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[216] V. John and S. Mita, „Deep Feature-Level Sensor Fusion Using Skip Connections for Real-Time Object Detection in Autonomous Driving," *Electronics*, vol. 10, no. 4, p. 424, 2021, DOI: 10.3390/electronics10040424.

[217] V. John and S. Mita, „RVNet: Deep Sensor Fusion of Monocular Camera and Radar for Image-based Obstacle Detection in Challenging Environments," in *Pacific-Rim Symposium on Image and Video Technology*, 2019, pp. 351–364.

[218] V. John, M. K. Nithilan, S. Mita, H. Tehrani, R. S. Sudheesh and P. P. Lalu, „SO-Net: Joint Semantic Segmentation and Obstacle Detection Using Deep Fusion of Monocular Camera and Radar," in *Image and Video Technology*, 2020, pp. 138–148, ISBN: 978-3-030-39769-2. DOI: 10.1007/978-3-030-39770-8_11.

[219] R. Yadav, A. Vierling and K. Berns, „Radar + RGB Fusion For Robust Object Detection In Autonomous Vehicle," in *2020 IEEE International Conference on Image Processing*, Abu Dhabi, United Arab Emirates, 2020, pp. 1986–1990, ISBN: 978-1-7281-6395-6. DOI: 10.1109/ICIP40778.2020.9191046.

[220] Z. Wang, X. Miao, Z. Huang and H. Luo, „Research of Target Detection and Classification Techniques Using Millimeter-Wave Radar and Vision Sensors," *Remote Sensing*, vol. 13, no. 6, p. 1064, 2021, DOI: 10.3390/rs13061064.

[221] X. Chen, H. Ma, J. Wan, B. Li and T. Xia, „Multi-View 3D Object Detection Network for Autonomous Driving," in *2017 30th IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017.

[222] J. Ku, M. Mozifian, J. Lee, A. Harakeh and S. Waslander, „Joint 3D Proposal Generation and Object Detection from View Aggregation," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[223] M. Meyer and G. Kuschk, „Deep Learning Based 3D Object Detection for Automotive Radar and Camera," *16th European Radar Conference*, 2019.

[224] T.-Y. Lim, A. Ansari, B. Major, Daniel Fontijne, M. Hamilton, R. Gowaikar and S. Subramanian, „Radar and Camera Early Fusion for Vehicle Detection in Advanced Driver Assistance Systems," *33rd Conference on Neural Information Processing Systems*, 2019.

[225] Y. Kim, J.-W. Choi and D. Kum, „GRIF Net: Gated Region of Interest Fusion Network for Robust 3D Object Detection from Radar Point Cloud and Monocular Image," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[226] R. Nabati and H. Qi, „CenterFusion: Center-based Radar and Camera Fusion for 3D Object Detection," *arXiv*, 2020, DOI: 10.1109/WACV48630.2021.00157. Available: https://arxiv.org/pdf/2011.04841.

[227] B. Yang, R. Guo, M. Liang, S. Casas and R. Urtasun, „RadarNet: Exploiting Radar for Robust Perception of Dynamic Objects," *arXiv*, 2020. Available: https://arxiv.org/pdf/2007.14366.

[228] M. Shah, Z. Huang, A. Laddha, M. Langford, B. Barber, S. Zhang, C. Vallespi-Gonzalez and R. Urtasun, „LiRaNet: End-to-End Trajectory Prediction using Spatio-Temporal Radar Fusion," *arXiv*, 2020. Available: https://arxiv.org/pdf/2010.00731.

[229] L. Wang, T. Chen, C. Anklam and B. Goldluecke, „High Dimensional Frustum PointNet for 3D Object Detection from Camera, LiDAR, and Radar," in *2020 IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, USA, 2020, pp. 1621–1628, DOI: 10.1109/IV47402.2020.9304655.

[230] C. Wang, C. Ma, M. Zhu and X. Yang, „PointAugmenting: Cross-Modal Augmentation for 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11794–11803.

[231] F. Nobis, M. Geisslinger, M. Weber, J. Betz and M. Lienkamp, „A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection," in *2019 Symposium on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, 2019, pp. 1–7, DOI: 10.1109/SDF.2019.8916629.

[232] F. Nobis, E. Shafiei, P. Karle, J. Betz and M. Lienkamp, „Radar Voxel Fusion for 3D Object Detection," *Applied Sciences*, vol. 11, no. 12, p. 5598, 2021, DOI: 10.3390/app11125598.

[233] G. Barnum, S. Talukder and Y. Yue, „On the Benefits of Early Fusion in Multimodal Representation Learning," *arXiv*, 2020. Available: https://arxiv.org/pdf/2011.07191.

[234] F. Fent and F. Nobis. „RadarSeg," 2021. [Online]. Available: https://github.com/TUMFTM/RadarSeg [visited on 04/12/2021].

[235] M. Geisslinger, M. Weber and F. Nobis. „CameraRadarFusionNet," 2021. [Online]. Available: https://github.com/TUMFTM/CameraRadarFusionNet [visited on 04/12/2021].

[236] E. Shafiei and F. Nobis. „RadarVoxelFusionNet," 2021. [Online]. Available: https://github.com/TUMFTM/RadarVoxelFusionNet [visited on 04/12/2021].

[237] V. Sternlicht, „A Data-Centric Approach to Point Cloud Segmentation," Semester Thesis, TU Munich, Munich, 2021.

[238] F. Fent, „Machine Learning based Object Classification with Automotive Radar Sensors," Semester Thesis, TU Munich, Munich, 2020.

[239] F. Nobis. „A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection," 2021. [Online]. Available: https://www.youtube.com/watch?v=JhF90n0fOnU [visited on 04/12/2021].

[240] J. H. Friedman, „Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, 2001, DOI: 10.1214/aos/1013203451.

[241] S. Chang, Y. Zhang, F. Zhang, X. Zhao, S. Huang, Z. Feng and Z. Wei, „Spatial Attention Fusion for Obstacle Detection Using MmWave Radar and Vision Sensor," *Sensors*, vol. 20, no. 4, 2020, DOI: 10.3390/s20040956.

[242] K. Kowol, M. Rottmann, S. Bracke and H. Gottschalk, „YOdar: Uncertainty-based Sensor Fusion for Vehicle Detection with Camera and Radar Sensors," *arXiv*, 2020. Available: https://arxiv.org/pdf/2010.03320.

[243] K. Schmitt, „Fusion of Camera and Radar Data at Continental," Meeting, 2021.

[244] F. Mengele, „Fusion of Camera and Radar Data at Conti Temic microelectronic GmbH," Email, 2021.

[245] B. Berman. „The key to autonomous vehicle safety is ODD," 2021. [Online]. Available: https://www.sae.org/news/2019/11/odds-for-av-testing [visited on 07/23/2021].

[246] A. Ng. „A Chat with Andrew on MLOps: From Model-Centric to Data-Centric AI," 2021. [Online]. Available: https://fullstackfeed.com/a-chat-with-andrew-on-mlops-from-model-centric-to-data-centric-ai-video/ [visited on 04/12/2021].

[247] A. Davis. „Ford and VW Hitch Their Self-Driving Efforts Together," 2021. [Online]. Available: https://www.wired.com/story/ford-vw-hitch-self-driving-efforts-together/ [visited on 07/16/2021].

[248] S. Crow. „10 major mergers & acquisitions in autonomous vehicles," 2021. [Online]. Available: https://www.therobotreport.com/10-major-mergers-acquisitions-autonomous-vehicles/ [visited on 07/16/2021].

[249] A. J. Hawkins. „Toyota is buying Lyft's autonomous car division for $550 million," 2021. [Online]. Available: https://www.theverge.com/2021/4/26/22404406/toyota-lyft-autonomous-vehicle-acquisition-amount-deal [visited on 07/16/2021].

[250] P. Koopman and M. Wagner, „Challenges in Autonomous Vehicle Testing and Validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[251] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp and B. Lohmann, „Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, vol. 25, no. 8, pp. 1–31, 2019, DOI: 10.1080/00423114.2019.1631455.

[252] T. A. Litman. „Autonomous Vehicle Implementation Predictions: Implications for Transport Planning," 2021. [Online]. Available: https://www.vtpi.org/avip.pdf [visited on 07/16/2021].

[253] Morgan Stanley. „Autonomous Cars: Self-Driving the New Auto Industry Paradigm," 2021. [Online]. Available: https://robotonomics.wordpress.com/2014/02/26/morgan-stanley-the-economic-benefits-of-driverless-cars/ [visited on 07/16/2021].

[254] Luminar. „Hydra," 2021. [Online]. Available: https://www.luminartech.com/thank-you-hydra/ [visited on 07/16/2021].

[255] T. Visentin, „Polarimetric Radar for Automotive Applications," Dissertation, Karlsruhe Institute of Technology, Karlsruhe, 2019, DOI: 10.5445/KSP/1000090003.

[256] K. Qian, Z. He and X. Zhang, „3D Point Cloud Generation with Millimeter-Wave Radar," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 4, pp. 1–23, 2020, DOI: 10.1145/3432221.

[257] S. Liu, B. Yu, Y. Liu, K. Zhang, Y. Qiao, T. Y. Li, J. Tang and Y. Zhu, „The Matter of Time – A General and Efficient System for Precise Sensor Synchronization in Robotic Computing," *arXiv*, 2021. Available: https://arxiv.org/pdf/2103.16045.

[258] D. Scaramuzza, A. Martinelli and R. Siegwart, „A Toolbox for Easily Calibrating Omnidirectional Cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 5695–5701, DOI: 10.1109/IROS.2006.282372.

[259] J. Domhof, J. F. Kooij and D. M. Gavrila, „An Extrinsic Calibration Tool for Radar, Camera and Lidar," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, 2019, pp. 8107–8113, DOI: 10.1109/ICRA.2019.8794186.

[260] N. Schneider, F. Piewak, C. Stiller and U. Franke, „RegNet: Multimodal Sensor Registration Using Deep Neural Networks," *arXiv*, 2017. Available: https://arxiv.org/abs/1707.03167.

[261] C. Schöller, M. Schnettler, A. Krammer, G. Hinz, M. Bakovic, M. Guzet and A. Knoll, „Targetless Rotational Auto-Calibration of Radar and Camera for Intelligent Transportation Systems," in *2019 IEEE 22nd International Conference on Intelligent Transportation Systems*, Auckland, New Zealand, 2019, pp. 3934–3941, DOI: 10.1109/ITSC.2019.8917135.

[262] A. Hanel and U. Stilla, „Iterative Calibration of a Vehicle Camera using Traffic Signs Detected by a Convolutional Neural Network," in *4th International Conference on Vehicle Technology and Intelligent Transport Systems*, Funchal, Madeira, Portugal, 2018.

[263] B. Wang, V. Wu, B. Wu and K. Keutzer, „LATTE: Accelerating LiDAR Point Cloud Annotation via Sensor Fusion, One-Click Annotation, and Tracking," *arXiv*, 2019. Available: https://arxiv.org/pdf/1904.09085.

[264] W. Zimmer, A. Rangesh and M. Trivedi, „3D BAT: A Semi-Automatic, Web-based 3D Annotation Toolbox for Full-Surround, Multi-Modal Data Streams," *arXiv*, 2019. Available: https://arxiv.org/abs/1905.00525.

[265] Intel. „Computer Vision Annotation Tool (CVAT)," 2018. [Online]. Available: https://github.com/opencv/cvat [visited on 06/28/2018].

[266] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li and S. Han, „AMC: AutoML for Model Compression and Acceleration on Mobile Devices," *arXiv*, 2018. Available: https://arxiv.org/abs/1802.03494.

[267] 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," *arXiv*, 2017. Available: https://arxiv.org/abs/1602.073603.

[268] J. Stanisz, K. Lis, T. Kryjak and M. Gorgon, „Hardware-software implementation of the PointPillars network for 3D object detection in point clouds," in *Workshop on Design and Architectures for Signal and Image Processing*, Budapest Hungary, 2021, pp. 44–51, DOI: 10.1145/3441110.3441150.

[269] C. Hao and D. Chen, „Software/Hardware Co-design for Multi-modal Multi-task Learning in Autonomous Systems," *arXiv*, 2021. Available: https://arxiv.org/pdf/2104.04000.

[270] Z. Yang, Y. Chai, D. Anguelov, Y. Zhou, P. Sun, D. Erhan, S. Rafferty and H. Kretzschmar, „SurfelGAN: Synthesizing Realistic Sensor Data for Autonomous Driving," *arXiv*, 2020. Available: https://arxiv.org/pdf/2005.03844.

[271] S. Manivasagam, S. Wang, K. Wong, W. Zeng, M. Sazanovich, S. Tan, B. Yang, W.-C. Ma and R. Urtasun, „LiDARsim: Realistic LiDAR Simulation by Leveraging the Real World," *arXiv*, 2020. Available: https://arxiv.org/pdf/2006.09348.

[272] C. Schoffmann, B. Ubezio, C. Bohm, S. Muhlbacher-Karrer and H. Zangl, „Virtual Radar: Real-Time Millimeter-Wave Radar Sensor Simulation for Perception-Driven Robotics," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4704–4711, 2021, DOI: 10.1109/LRA.2021.3068916.

[273] M. F. Holder, C. Linnhoff, P. Rosenberger, C. Popp and H. Winner, „Modeling and Simulation of Radar Sensor Artifacts for Virtual Testing of Autonomous Driving," in *9. Tagung Automatisiertes Fahren*, 2019.

[274] J. Fang, D. Zhou, X. Song and L. Zhang, „MapFusion: A General Framework for 3D Object Detection with HDMaps," *arXiv*, 2021. Available: http://arxiv.org/pdf/2103.05929v1.

[275] K. Wong, S. Wang, M. Ren, M. Liang and R. Urtasun, „Identifying Unknown Instances for Autonomous Driving," *arXiv*, 2019. Available: https://arxiv.org/pdf/1910.11296.

[276] D. Miller, L. Nicholson, F. Dayoub and N. Sunderhauf, „Dropout Sampling for Robust Object Detection in Open-Set Conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA): 21-25 May 2018*, Brisbane, Australia, 2018, pp. 3243–3249, DOI: 10.1109/ICRA.2018.8460700.

[277] Di Feng, A. Harakeh, S. Waslander and K. Dietmayer, „A Review and Comparative Study on Probabilistic Object Detection in Autonomous Driving," *arXiv*, 2020. Available: https://arxiv.org/pdf/2011.10671.

[278] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova and N. Sünderhauf, „Probabilistic Object Detection: Definition and Evaluation," *arXiv*, 2018. Available: https://arxiv.org/abs/1811.10800.

[279] M. Hoss, M. Scholtes and L. Eckstein, „A Review of Testing Object-Based Environment Perception for Safe Automated Driving," *arXiv*, 2021. Available: https://arxiv.org/pdf/2102.08460.

[280] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow and R. Fergus, „Intriguing properties of neural networks," *arXiv*, 2013. Available: https://arxiv.org/pdf/1312.6199.

[281] R. Geirhos, D. H. J. Janssen, H. H. Schütt, J. Rauber, M. Bethge and F. A. Wichmann, „Comparing deep neural networks against humans: Object recognition when the signal gets weaker," *arXiv*, 2017. Available: https://arxiv.org/abs/1706.06969.

[282] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu and Z. M. Mao, „Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving," in *2019 ACM SIGSAC Conference on Computer and Communications Security*, London United Kingdom, 2019, pp. 2267–2281, DOI: 10.1145/3319535.3339815.

Prior Publications

During the development of this dissertation, publications and student theses were written in which partial aspects of this work were presented.

Journals; Scopus/Web of Science listed (peer-reviewed)

[32] P. R Palafox, J. Betz, F. Nobis, K. Riedl and M. Lienkamp, „SemanticDepth: Fusing Semantic Segmentation and Monocular Depth Estimation for Enabling Autonomous Driving in Roads without Lane Lines," *Sensors*, vol. 19, no. 14, p. 3224, 2019, DOI: 10.3390/s19143224.

[72] F. Nobis, F. Fent, J. Betz and M. Lienkamp, „Kernel Point Convolution LSTM Networks for Radar Point Cloud Segmentation," *Applied Sciences*, vol. 11, no. 6, p. 2599, 2021, DOI: 10.3390/app11062599.

[232] F. Nobis, E. Shafiei, P. Karle, J. Betz and M. Lienkamp, „Radar Voxel Fusion for 3D Object Detection," *Applied Sciences*, vol. 11, no. 12, p. 5598, 2021, DOI: 10.3390/app11125598.

Conferences, Periodicals; Scopus/Web of Science listed (peer-reviewed)

[20] F. Nobis, O. Papanikolaou, J. Betz and M. Lienkamp, „Persistent Map Saving for Visual Localization for Autonomous Vehicles: An ORB-SLAM 2 Extension," in *Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, Monaco, 2020, pp. 1–9, DOI: 10.1109/EVER48776.2020.9243094.

[30] K. Riedl, S. Huber, M. Bomer, J. Kreibich, F. Nobis and J. Betz, „Importance of Contextual Information for the Detection of Road Damages," in *Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*, Monaco, 2020, pp. 1–7, DOI: 10.1109/EVER48776.2020.9242954.

[90] F. Nobis, F. Brunhuber, S. Janssen, J. Betz and M. Lienkamp, „Exploring the Capabilities and Limits of 3D Monocular Object Detection - A Study on Simulation and Real World Data," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems*, Rhodes, 2020, pp. 1–8, DOI: 10.1109/ITSC45102.2020.9294625.

[231] F. Nobis, M. Geisslinger, M. Weber, J. Betz and M. Lienkamp, „A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection," in *2019 Symposium on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, 2019, pp. 1–7, DOI: 10.1109/SDF.2019.8916629.

Journals, Conferences, Periodicals, Reports, Conference Proceedings and Poster, etc.; not Scopus/Web of Science listed

J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, B. Lohmann and M. Lienkamp, „What can we learn from autonomous level-5 motorsport?," in *9th International Munich Chassis Symposium*, P. Pfeffer, ed. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, pp. 123–146, DOI: 10.1007/978-3-658-22050-1_12.

Non-thesis-relevant publications; Scopus/Web of Science listed (peer-reviewed)

J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer and M. Lienkamp, „A Software Architecture for an Autonomous Racecar," in *2019 IEEE 89th Vehicular Technology Conference*, Kuala Lumpur, Malaysia, 2019, pp. 1–6, DOI: 10.1109/VTCSpring.2019.8746367.

J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, L. Hermansdorfer, T. Stahl, T. Herrmann and M. Lienkamp, „A Software Architecture for the Dynamic Path Planning of an Autonomous Racecar at the Limits of Handling," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, Graz, Austria, 2019, pp. 1–8, DOI: 10.1109/ICCVE45908.2019.8965238.

F. Nobis, J. Betz, L. Hermansdorfer and M. Lienkamp, „Autonomous Racing: A Comparison of SLAM Algorithms for Large Scale Outdoor Environments," in *3rd International Conference on Virtual and Augmented Reality Simulations*, 2019, pp. 82–89, DOI: 10.1145/3332305.3332319.

Thesis-relevant open-source software

[234] F. Fent and F. Nobis. „RadarSeg," 2021. [Online]. Available: https://github.com/TUMFTM/RadarSeg [visited on 04/12/2021].

[235] M. Geisslinger, M. Weber and F. Nobis. „CameraRadarFusionNet," 2021. [Online]. Available: https://github.com/TUMFTM/CameraRadarFusionNet [visited on 04/12/2021].

[236] E. Shafiei and F. Nobis. „RadarVoxelFusionNet," 2021. [Online]. Available: https://github.com/TUMFTM/RadarVoxelFusionNet [visited on 04/12/2021].

O. Papanikolaou and F. Nobis. „ORSBSLAM Map Saving Extension," 2021. [Online]. Available: https://github.com/TUMFTM/orbslam-map-saving-extension [visited on 04/12/2021].

Supervised Students' Theses

The following student theses were written within the framework of the dissertation under the supervision of the author in terms of content, technical and scientific support as well as under relevant guidance of the author. In the following, the bachelor, semester and master theses relevant and related to this dissertation are listed. Many thanks to the authors of these theses for their extensive support within the framework of this research project.

[75] M. Geisslinger, „Autonomous Driving: Object Detection using Neural Networks for Radar and Camera Sensor Fusion," Master's Thesis, TU Munich, Munich, 2019.

[237] V. Sternlicht, „A Data-Centric Approach to Point Cloud Segmentation," Semester Thesis, TU Munich, Munich, 2021.

[238] F. Fent, „Machine Learning based Object Classification with Automotive Radar Sensors," Semester Thesis, TU Munich, Munich, 2020.

F. Spiegel, „Radar-Based Object Localization using Clustering and Tracking," Master's Thesis, TU Munich, Munich, 2019.

T. Woehrmueller, „Single Image Super Resolution for Automotive Applications," Master's Thesis, TU Munich, Munich, 2019.

M. Weber, „Autonomous Driving: Radar Sensor Noise Filtering and Multimodal Sensor Fusion for Object Detection with Artificial Neural Networks," Master's Thesis, TU Munich, Munich, 2019.

F. Pfab, „Autonomes Rennfahrzeug - LIDAR Objekttracking," Semester Thesis, TU Munich, Munich, 2019.

F. Schramm, „Autonomer Rennsport - Visual Stereo SLAM auf einem RC-Fahrzeug," Bachelor's Thesis, TU Munich, Munich, 2018.

F. Fent, „Machine Learning-Based Radar Point Cloud Segmentation," Master's Thesis, TU Munich, Munich, 2021.

M. Fortkord, „Autonomous Driving: Exploring Training Strategies for a Camera and Radar Fusion Network for Object Detection," Semester Thesis, TU Munich, Munich, 2020.

F. Brunhuber, „3D-Kamera-Objektdetektion mittels künstlicher neuronaler Netze: Erweiterung einer Simulationsumgebung zur Generierung synthetischer Trainingsdaten für das autonome Fahren," Semester Thesis, TU Munich, Munich, 2020.

F. Asanger, „Selbstfahrende Rennfahrzeuge-Aufbau einer Simulationsumgebung," Bachelor's Thesis, TU Munich, Munich, 2018.

A. Huefner, „Development of a Deep Learning based Mapping Pipeline using Fisheye Cameras," Master's Thesis, TU Munich, Munich, 2019.

S. Janssen, „3D Object Detection for Autonomous Racecars based on Monocular Depth Estimation," Semester Thesis, TU Munich, Munich, 2020.

M. von Krogh, „Physics-Aware Machine Learning: Vehicle Movement Prediction,“ Master’s Thesis, TU Munich, Munich, 2019.