AGH University of Krakow

FIELD OF SCIENCE ENGINEERING AND TECHNOLOGY

SCIENTIFIC DISCIPLINE AUTOMATION, ELECTRONIC, ELECTRICAL ENGINEERING AND SPACE TECHNOLOGIES

DOCTORAL THESIS

Low-level sensor data fusion for Object Detection in an Autonomous Vehicle perception system based on a Machine Learning approach.

Author: Mgr inż. Daniel Dworak

First supervisor: Dr hab. inż. Jerzy Baranowski, prof. AGH
Assisting supervisor: Dr inż. Mateusz Komorkiewicz

Completed in: Faculty of Electrical Engineering, Automatics, Computer Science, and Biomedical Engineering

Kraków, 2023

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

DZIEDZINA NAUK INŻYNIERYJNO-TECHNICZNYCH

DYSCYPLINA AUTOMATYKA, ELEKTRONIKA, ELEKTROTECHNIKA I TECHNOLOGIE KOSMICZNE

ROZPRAWA DOKTORSKA

Niskopoziomowa fuzja danych sensorycznych do detekcji obiektów w systemie percepcji pojazdu autonomicznego bazująca na technikach uczenia maszynowego.

Autor: Mgr inż. Daniel Dworak

Promotor rozprawy: Dr hab. inż. Jerzy Baranowski, prof. AGH
Promotor pomocniczy: Dr inż. Mateusz Komorkiewicz

Praca wykonana: Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Kraków, 2023

I extend my gratitude to both of my supervisors for their invaluable assistance and guidance throughout the course of this research and the crafting of this thesis. Additionally, I would like to express my sincere appreciation to my wife, parents, and all my friends whose support played an essential role in helping me achieve this goal.

Abstract

Autonomous Driving is a major research topic in the automotive domain. The promise of fully automating the driving process holds the potential to deliver substantial advantages, encompassing heightened user comfort and a considerable enhancement in overall safety on the roads. New tools and technological advances enable gradually more sophisticated systems, which try to closely reassemble the entirety of Autonomous Vehicle capabilities. Within this transformative area, sensors such as cameras, LiDAR and Radar play an essential role in perception systems, which corresponds to the cognitive functions of an AV. These sensors serve as the eyes and ears of autonomous systems, capturing crucial environmental data in the form of images or pointcloud readings. Throughout this thesis, a thorough exploration of automotive sensors is presented, focusing on both their hardware design and provided data formats, as this data constitutes the input to dedicated perception algorithms, which create a digital model of the surroundings.

Among those algorithms, Machine Learning methods in particular have recently gained significant recognition within the scope of perception systems. Especially in Object Detection problems, through the analysis of sensor data, those solutions tend to dominate in the current industrial applications. They achieve outstanding performance and offer perspectives for further improvements. This thesis research topic is centred exactly around AV perception systems, which are based on ML methods, and more precisely on Convolutional Neural Networks and Deep Learning approaches. These techniques have advanced the field of perception, enabling vehicles to sense their surroundings with remarkable accuracy. By utilizing modern architecture designs, reviewed in this research, such networks can decode intricate patterns and representations from sensor data, resulting in a high-level understanding of the environment.

Moreover, a comprehensive perception may benefit significantly from complementary information provided by various sources. Each sensor has its advantages and disadvantages for perception purposes and the combination of different devices in the AV sensor suite could mitigate their undesirable traits, consequently improving the whole system. Addressing that claim, this research focuses primarily on the utilization of sensor data fusion and the exploration of its benefits for such systems. The concept of data fusion is discussed in detail and different methods of fusing sensor data are presented. Among those, the special type of low-level data fusion is particularly interesting, as it naturally pairs well with the Neural Networks processing approach. The main goal of this thesis is to determine whether the ML-based Low-Level Fusion solution could prove to be beneficial for an OD task in an AV perception system. The target fusion is performed on camera images and pointcloud data from either LiDAR or Radar.

The key innovation in the pursuit of that goal is a novel approach to Low-Level Fusion, called the Cross-Domain Spatial Matching method. This method offers an alternative methodology, not yet seen in this research domain. It comprises two main elements: sensor data domain alignment and fusion methods. The former component addresses the challenge associated with disparate orientations of data samples in relation to the shared host vehicle coordinate system. Once the data is aligned, it facilitates the integration of samples from various sensors without the need for additional explicit projections. It also allows for the latter fusion strategies to be applied directly to the domain alignment output. In this research, three unique fusion strategies are proposed to be further verified, each built upon a different approach, posing distinct benefits. Both CDSM elements are seamlessly integrated into the Neural Network architecture. This integration not only facilitates end-to-end training but also contributes to efficient inference times during operational deployment.

In order to implement a complete OD network architecture, several single-sensor models are also created, based on State-Of-The-Art solutions from corresponding domains. These models serve a dual purpose: firstly, to evaluate the efficiency of the perception system when reliant solely on one sensor, and secondly, as integral submodules within the fusion architecture, responsible for extracting feature maps from each input sample. Through extensive experimentation and proper evaluation techniques, further exploration of fusion validity is conducted, using two open-source automotive datasets - KITTI and NuScenes. The thesis contains a detailed description of those datasets together with a training process overview. Furthermore, the results of single-sensor and fusion models are shown and compared to each other. The in-depth analysis of models' predictions is performed in terms of both visual and KPI metrics performance. For the fusion solution, the efficiency gain is highlighted and examples of corner cases are presented, offering insight into scenarios where the fusion model diverges from the predictions of single-sensor counterparts. The comparison to SOTA fusion solutions is also provided, facilitating the positioning of CDSM among currently leading techniques. Finally, these results analysis allows for drawing conclusions regarding such a fusion methodology in AV perception systems and whether LLF could be beneficial for it.

Streszczenie

Jazda autonomiczna to jeden z głównych tematów badawczych w branży motoryzacyjnej. Pełna automatyzacji procesu prowadzenia pojazdu może przynieść znaczne korzyści, obejmujące wyższy komfort użytkowania takiego pojazdu i poprawę ogólnego bezpieczeństwa na drogach. Nowe narzędzia i postęp technologiczny umożliwiają tworzenie coraz bardziej zaawansowanych systemów, które starają się w pełni zrealizować możliwości pojazdów autonomicznych. W tym nieustannie zmieniającym się obszarze, czujniki takie jak kamery, LiDAR i Radar odgrywają istotną rolę w systemach percepcji, odpowiedzialnych za funkcje poznawcze takich pojazdów. Czujniki te pełnią rolę oczu i uszu systemów percepcji, rejestrując kluczowe dane z otoczenia w postaci obrazów czy chmur punktów. W niniejszej rozprawie przedstawiona jest dogłębna analiza takich czujników, koncentrująca się zarówno na ich budowie, jak i formacie dostarczanych przez nie danych. Jest to szczególnie istotne z uwagi na to, że stanowią one dane wejściowe dla dedykowanych algorytmów percepcji, które tworzą cyfrowy model otoczenia.

Wśród tych algorytmów, metody uczenia maszynowego zyskały w ostatnim czasie znaczną popularność w odniesieniu do systemów percepcji. Zwłaszcza w problemach związanych z wykrywaniem obiektów, poprzez wnikliwą analizę danych z czujników, takie rozwiązania dominują obecnie w aplikacjach przemysłowych. Osiągają one doskonałe wyniki i oferują perspektywy dalszych udoskonaleń. Temat badawczy tej pracy skupia się właśnie na systemach percepcji pojazdów autonomicznych opartych o metody uczenia maszynowego, a konkretnie konwolucyjne sieci neuronowe i podejście głębokie do ich projektowania. Te techniki znacznie usprawniły dziedzinę percepcji, umożliwiając pojazdom precyzyjne postrzeganie otoczenia. Dzięki wykorzystaniu nowoczesnych architektury omówionych w tej pracy, takie sieci są w stanie odkrywać skomplikowane wzorce i reprezentacje z danych pochodzących z czujników, co prowadzi do lepszego zrozumienia badanego otoczenia.

Ponadto, metodyka percepcji może znacząco skorzystać z dopełniających się informacji dostarczanych z różnych źródeł. W kontekście takiego rozwiązania, każdy czujnik ma swoje zalety i wady, a połączenie różnych urządzeń w zestawie czujników pojazdu autonomicznego może złagodzić ich niepożądane cechy, co w rezultacie poprawi cały system. Kierując się tą myślą, niniejsza praca badawcza skupia się przede wszystkim na wykorzystaniu fuzji danych z czujników i eksploracji jej korzyści. Koncepcja fuzji danych jest szczegółowo omówiona wraz z różnymi metodami łączenia danych z czujników. Spośród tych metod szczególnie interesującym jest specjalny typ niskopoziomowej fuzji danych, który naturalnie dobrze współgra z podejściem przetwarzania danych z użyciem sieci neuronowych. Głównym celem tej rozprawy dok-

torskiej jest określenie, czy oparte na metodach uczenia maszynowego rozwiązanie fuzji niskopoziomowej może okazać się korzystne dla systemu percepcji pojazdu autonomicznego w zadaniu związanym z wykrywaniem obiektów na drodze. Fuzja ta jest przeprowadzana na obrazach z kamer oraz danych chmur punktów z czujników LiDAR lub radaru.

Kluczową innowacją w dążeniu do tego celu jest nowatorskie podejście do fuzji niskopoziomowej, nazwane metodą Cross-Domain Spatial Matching. Metoda CDSM oferuje alternatywną technikę fuzji, niespotykaną dotąd w tej dziedzinie badań. Składa się ona z dwóch głównych elementów: dopasowania domen danych z czujników i metod fuzji. Pierwszy komponent zajmuje się ujednoliceniem odczytów czujników związanym z różnymi orientacjami próbek w stosunku do wspólnego układu współrzędnych pojazdu. Po takim dopasowaniu możliwe staje się łączenie danych z różnych czujników bez dodatkowych rzutowań i konwersji. Pozwala to także na bezpośrednie zastosowanie zaproponowanych strategii fuzji. W tej pracy badawczej przedstawiono trzy takie unikalne strategie, bazujące na różnych podejściach i niosące różnorodne korzyści, które są poddane dalszej weryfikacji w kolejnych rozdziałach. Oba elementy CDSM są zintegrowane w stworzonej architekturze sieci neuronowej. Ta integracja ułatwia nie tylko uczenie sieci w tak zwanym podejściu *end-to-end*, ale także przyczynia się do efektywnych czasów inferencji modelu podczas jego wdrożenia.

Aby zaimplementować kompletną architekturę sieci neuronowej związanej z percepcją obiektów, zaprojektowane są również modele przetwarzające dane z pojedynczych czujników, oparte na najnowocześniejszych rozwiązaniach z odpowiadających im dziedzin. Modele te spełniają podwójną rolę: po pierwsze, pozwalają na ocenę wydajności systemu percepcji, gdy opiera się on wyłącznie na jednym czujniku, a po drugie, stanowią integralne moduły w architekturze fuzji, odpowiedzialne za wyodrębnianie cech z każdej próbki danych wejściowych. Poprzez dokładne eksperymenty i odpowiednie techniki oceny jakości, przeprowadzane są dalsze badania dotyczące skuteczności fuzji, wykorzystujące dwa publiczne zbiory danych z dziedziny motoryzacji - KITTI i NuScenes. Praca doktorska zawiera szczegółową charakterystykę tych zbiorów wraz z dokładnym opisem przebiegu procesu uczenia poszczególnych modeli. Ponadto, wyniki modeli bazujących na pojedynczych czujnikach oraz modeli fuzji zostają przedstawione w sposób, który umożliwia porównanie ich ze sobą. Przeprowadzona zostaję dogłębna analiza wyników predykcji obiektów zarówno wizualna, jak i pod względem przyjętych wskaźników jakości. Dla rozwiązania opartego o fuzję, analiza zostaje rozszerzona o badania pokazujące zyski nad jedno-czujnikowymi modelami oraz przykłady przypadków brzegowych, oferujące wgląd w scenariusze, w których model fuzji odbiega od odpowiedników bazujących tylko na jednym źródle danych. Zapewnione jest również porównanie do wiodących rozwiązań fuzyjnych, co ułatwia umiejscowienie metody CDSM wśród obecnie wykorzystywanych technik. Analiza tych wyników pozwala na wyciągnięcie wniosków dotyczących zaproponowanej metody fuzji w systemach percepcji pojazdów autonomicznych oraz sformułowania odpowiedzi na pytanie, czy fuzja niskopoziomowa może okazać się korzystna do tego celu.

Contents

List of Abbreviations

RoI Region of Interest . . . . . p. 4

RPN Region Proposal Network . . . . . p. 28

RYYCy Red-Yellow-Yellow-Cyan . . . . . p. 14

SAE Society of Automotive Engineers . . . . . p. 8

SE Squeeze-and-Excitation . . . . . p. 27

SGD Stochastic Gradient Descent . . . . . p. 26

SOTA State-Of-The-Art . . . . . p. 3

SSD Single-Shot Detector . . . . . p. 28

Swin Shifted-window Transformer . . . . . p. 30

ToF Time-of-Flight . . . . . p. 16

TSR Traffic Sign Recognition . . . . . p. 8

VCS Vehicle Coordinate System . . . . . p. 3

VFE Voxel Feature Extractor . . . . . p. 31

ViT Vision Transformer . . . . . p. 30

XAI Explainable AI . . . . . p. 3

Chapter 1

Introduction

Autonomous Driving (AD) research aims to automate the driving process, transferring the driver role from a human to a computer system (Parekh et al. 2022). This system should ensure a consistent performance level and reliable decision-making, minimizing the potential for human errors and subsequently enhancing overall road safety (Morales-Alvarez et al. 2020). Despite considerable progress in hardware, sensor suite specifications, computing capabilities, and broader software improvements, the full AD system potential has not been achieved yet according to the established autonomy classification standards (Galvani 2019).

The core component of Autonomous Vehicle (AV) is the perception system, responsible for generating a digital representation of the environment (Skruch et al. 2022). All other high-level modules depend on it as an interface to the real world. The perception system consists of sensors and algorithms that process their readings. Among different techniques of processing sensor data, the Machine Learning (ML) methods are becoming increasingly popular (Shafiee et al. 2021). Especially Convolutional Neural Network (CNN) and Deep Learning (DL) approaches excel in Computer Vision (CV) and sensors' signal processing tasks, becoming leading industrial solutions in the automotive domain.

Furthermore, the perception algorithms can utilize a sensor data fusion to combine readings from multiple sensors into one coherent representation (Kocić, Jovičić, and Drndarević 2018). This fusion process extracts complementary information from diverse sources, potentially overcoming the limitations inherent in individual sensors and consequently enhancing overall system performance. Particularly, the Low-Level Fusion (LLF) strategy, which involves merging either raw or moderately processed sensor data (Pollach, Schiegg, and Knoll 2020), demonstrates substantial compatibility with processing methodologies based on Neural Network (NN) architectures.

This doctoral thesis is focused precisely on the research regarding ML-based LLF for AV perception system, and more specifically for Object Detection (OD) problem. In the next sections of this chapter, the motivation for exploring the topic of sensor data fusion is presented along with the main goals of the entire dissertation. Furthermore, the most significant additions to this research domain are highlighted in the contributions section. Finally, in the last section, the structure of the entire doctoral thesis is presented, with each chapter summary.

1.1 Motivation

Perception serves as the fundamental core for all AD systems, forming the basis for higher-level functions such as tracking and planning algorithms. An accurate and comprehensive perception of the surrounding environment is vital for AV to make well-informed decisions and navigate safely through traffic. Sensor fusion plays a crucial role in enhancing perception by combining data from various sensors, enabling a more extensive and reliable understanding of the environment (Wei et al. 2022). Furthermore, the utilization of LLF methods carries additional advantages to sensor fusion. These methods possess direct access to unprocessed sensor readings, presenting an array of distinct points of view and diverse data types. This versatile input diversity significantly broadens the scope of fusion possibilities and strengthens the underlying perception system.

Ensuring the safety of AVs and their interaction with the surrounding environment is of paramount importance. Perception systems perform a critical role in detecting nearby objects, such as other cars and pedestrians. Based on the perception outcome, active safety procedures are conducted to ensure secure behaviour and prevent accidents. However, relying on a single sensor for perception may lead to limitations and vulnerabilities. Sensor fusion research addresses this issue by integrating multiple sensors, leveraging their complementary strengths, and mitigating individual sensor limitations (Lindner et al. 2007). By fusing data from cameras, LiDAR (Light Detection And Ranging), and Radar (Radio Detection And Ranging) sensors, the perception system may become more robust, providing redundant information and reducing the chances of false positives or false negatives detections. Furthermore, in scenarios where object-level fusion algorithms struggle with conflicting intermediate detections from different sensors, a LLF-based solution can proficiently exploit raw data context to properly validate respective readings and adjust their impact on the final outcomes.

Sensor fusion extends the capabilities of the perception system beyond what individual sensors can achieve independently (Herpel et al. 2008). Each sensor provides a unique perspective on the environment, and fusing their data allows for a more comprehensive and accurate representation of the surroundings. By combining these sensors, the perception system can overcome individual limitations and weaknesses of each of them. Sensor fusion research aims to optimize the fusion algorithms and techniques to extract the most valuable information from each sensor. Especially, the LLF solutions based on DL approach exhibit the exceptional ability to find cross-sensor dependencies in raw data readings. This ability leads to the creation of new object representations deeply embedded within the NN structure, revealing previously concealed patterns, and enabling enhanced and more complex insights into the overall perception.

In conclusion, sensor fusion research is motivated by the critical role of perception as the core of other AV systems, the emphasis on safety in perception systems, and the potential of fusion to extend perception performance beyond the capabilities of individual sensors. Advancing sensor fusion techniques should drive the progress of AD technology, but it also presents new challenges, including synchronization and perspectives alignment (Lindenmaier et al. 2022), which this research tries to address.

1.2 Goal

The goal of this thesis is to **investigate and verify the effectiveness of a Low-Level Fusion that utilizes an automotive sensor suite to perform Object Detection task in an Autonomous Vehicle perception system**. By integrating multiple sensors and leveraging the strengths of each, this approach aims to enhance the overall performance and robustness of such systems. Through careful analysis and experimentation, the thesis focuses on **establishing the potential advantages of low-level sensor data fusion in improving perception accuracy, reducing uncertainty, and enabling a more comprehensive understanding of the environment** over single-sensor solutions.

Furthermore, by combining DL methodologies with the principles of LLF, the goal is to **investigate an end-to-end fusion of sensor data within one Neural Network architecture**. This approach aims to employ NN to uncover hidden patterns within various sensor readings, particularly in complex, high-dimensional data spaces. This exploration adds a new aspect to fusion techniques, showcasing whether ML can amplify the effectiveness of LLF in improving AV perception systems.

In addition to the aforementioned goal, in the thesis, the growing importance of explainability in AD systems is being recognized. As they make critical decisions, it becomes essential to understand the underlying reasoning behind model predictions. To address this, the thesis also **explores Explainable AI (XAI) techniques to enhance the interpretability of the decision-making processes in perception NN model**.

1.3 Contribution

The following section highlights the main contributions obtained during the research, which are explained in detail in further parts of this thesis. The aim of outlining these contributions is to showcase the most substantial value added, according to the author, within the respective areas of this dissertation:

- A comprehensive literature review was conducted, exploring the State-Of-The-Art (SOTA) techniques and advancements in NN-based AV's perception systems. This review provided a solid foundation for understanding the challenges, limitations, and existing solutions in the field of perception and sensor fusion.

- A novel low-level sensor fusion method called Cross-Domain Spatial Matching (CDSM) was developed and implemented. CDSM leverages extracted 2D camera features information and transforms it into a 3D Vehicle Coordinate System (VCS) to match the pointcloud 3D features. It enables a number of proposed fusion techniques to process sensor data within NN architecture for the final 3D OD task. The fusion results obtained using CDSM model were evaluated extensively, demonstrating improved accuracy, robustness, and reliability compared to single sensor models and other SOTA fusion methods. Those results validate the effectiveness of CDSM in enhancing the perception capabilities of AV perception systems.

- Explainable AI analysis of the perception models' decision-making process was performed using the Gradient-weighted Class Activation Maps (Grad-CAM) technique. This analysis provided visual explanations highlighting the Region of Interest (RoI) and the features contributing to the OD and classification decisions. The new method was proposed to apply the Grad-CAM method to LiDAR and Radar models. By comparing the Grad-CAM results of different single-sensor models, a deeper understanding was gained regarding the added value of each sensor in improving perception accuracy and reliability.

- The research has also resulted in several publications in conference materials and journals. These publications present the findings, methodologies, and novel approaches developed during the research, contributing to both the academic and the industrial communities. Additionally, multiple patent applications have been filled based on the methods and algorithms developed during the course of this research. These patents recognize the novelty of proposed ML techniques and highlight their potential for real-world applications.

1 Daniel Dworak, Filip Ciepiela, et al. "Performance of LiDAR object detection deep learning architectures based on artificially generated point cloud data from CARLA simulator". In: *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*. 2019, pp. 600–605.

2 Daniel Dworak. "BlurNet: Keeping Collected Data Private with a Neural Network Based Pipeline". In: *Advanced, Contemporary Control*. Springer International Publishing, 2020, pp. 1237–1248.

3 Jerzy Baranowski et al. "Analiza danych i optymalizacja w Przemyśle 4.0 — Data analysis and optimization in Industry 4.0". In: *Wydział Elektryczny AGH – Wczoraj, Dziś i Jutro*. 2022, pp. 43–52.

4 Daniel Dworak and Jerzy Baranowski. "Adaptation of Grad-CAM Method to Neural Network Architecture for LiDAR Pointcloud Object Detection". In: *Energies* 15.13 (2022).

5 Filip Ciepiela, Mariusz Karol Nowak, et al. "Automotive Radar Detection Level Modeling with Neural Networks". In: *Advanced, Contemporary Control*. Cham: Springer Nature Switzerland, 2023, pp. 254–265.

6 Mateusz Komorkiewicz et al. "Vehicles, systems, and methods for determining an entry of an occupancy map of a vicinity of a vehicle". EP3832531A1, Patent application. 2021.

7 Filip Ciepiela, Mateusz Komorkiewicz, et al. "Method and system for determining an output of a convolutional block of an artificial neural network". EP3885996A1, Patent application. 2021.

8 Mateusz Wójcik et al. "Method and system for interpolation and method and system for determining a map of a surrounding of a vehicle". EP3975105A1, Patent application. 2022.

9 Ori Maoz et al. "Methods and systems for determining candidate data sets for labelling". EP3985560A1, Patent application. 2022.

1.4 Thesis organization

The subsequent chapters of the thesis are organized as follows:

Chapter 2 explores the concept of AD, emphasizing the automation level classification as well as perception system definition and key components. It further discusses various automotive sensors, including cameras, LiDAR, and Radar, and their roles for AV's perception. Additionally, it introduces the basics of sensor data fusion, discussing fusion levels and stages.

Chapter 3 surveys DL models used in perception systems. Firstly, it forms an introduction to the theory of NNs and modern approaches to their architecture design. Then, a literature review of single-sensor models is presented, providing detailed descriptions of SOTA camera, LiDAR, and Radar solutions. Finally, SOTA fusion models that integrate data from multiple sensors are introduced and explained.

Chapter 4 gives an overview of the evaluation methods employed to assess the perception model's performance. It covers the process of associating labels with predictions. Furthermore, it introduces perception metrics used to evaluate the performance of these models. Additionally, the concept of XAI is discussed as a means of visual analysis and interpretation, with a specific emphasis on Grad-CAM.

Chapter 5 describes the proposed CDSM fusion architecture for AV perception. It addresses the input data preprocessing steps required for the models. Additionally, the process of extracting features from sensor data is explored and single-sensor submodels are defined. It also presents an explanation of the proposed fusion methods that merge extracted features. Finally, the chapter outlines the output format and predictions of the discussed models.

Chapter 6 discusses datasets used for training and evaluation. It examines two open-source datasets in detail, covering hardware setups, data samples, and provided labels. The chapter also outlines the training process methodology, including targets generation from labels, loss function definitions, optimization setup, and hyperparameter tuning.

Chapter 7 reports the outcomes of experiments concerning single-sensor model architectures. The camera, LiDAR, and Radar networks were all trained using the aforementioned datasets. The subsequent results are showcased and subjected to a thorough evaluation, which encompasses visualizing the outcomes and calculating relevant performance metrics.

Chapter 8 centres on the experiments involving fusion models. The ultimate fusion approach is chosen, based on the Key Performance Indicator (KPI)s performance. Both camera-LiDAR and camera-Radar fusion configurations undergo training and evaluation, mirroring the methodology employed for single-sensor models in Chapter 7. Additionally, this chapter discusses the fusion gain accomplished by the CDSM model, a comparison with SOTA solutions, as well as corner cases and encountered challenges.

Chapter 9 revolves around Explainable AI methods. It introduces adaptations to Grad-CAM analysis as a means of interpreting NN's output from different sensor domains. It explains the differences in Grad-CAM application to previously discussed perception architectures, focusing on visualizations and analysis of pointcloud-based LiDAR and Radar models.

Chapter 10 summarizes the main findings and contributions of the research. It discusses the implications and significance of the obtained results. Furthermore, it provides recommendations for future research in the field of fusion AV perception solutions, highlighting potential areas for further exploration and development.

The last pages contain the complete list of references used throughout the thesis.

Chapter 2

Autonomous Driving

Chapter highlights:

- *Levels of AD*
- *AV perception system definition*
- *Automotive sensor suite overview*
- *Sensor data fusion and its classification*

AD refers to the capability of a vehicle to operate and navigate through the street traffic without human intervention (Takács et al. 2018). AD systems aim to replicate or exceed human driving performance by continuously monitoring the environment and responding appropriately to ensure safe and efficient travel. These systems can control acceleration, braking, steering, and other functions required for driving. Furthermore, the development of Advanced Driver Assistance Systems (ADAS) has played a crucial role in shaping the trajectory of AD. ADAS act as stepping stones towards full autonomy by providing partial automation and enhancing driver safety and convenience. These systems serve as a bridge between traditional human-driven vehicles and fully autonomous ones, gradually introducing and familiarizing drivers with automated functionalities. AD has the potential to revolutionize transportation by reducing accidents caused by human error, improving traffic flow and congestion, and providing mobility options for individuals who cannot drive (Morales-Alvarez et al. 2020). It also holds promise for enhancing energy efficiency and reducing environmental impact through optimized driving patterns (Barth, Boriboonsomsin, and Wu 2013).

This chapter covers an examination of the different levels of autonomy, establishing a framework for understanding the progression of automation in vehicles. By reviewing these levels, insights are gained into the capabilities and responsibilities associated with each stage of AD. Next, the vital role of perception systems is described. Perception systems definition and core components are presented with their corresponding functions. Furthermore, various types of sensors used in AD are introduced, such as camera, LiDAR, and Radar. Presented functions and capabilities of each sensor type highlight the integral role which automotive sensors play in enabling AV's perception. Finally, the concept of sensor data fusion is demonstrated. It showcases the process of integrating data from multiple sensors to create a comprehensive model of the vehicle's environment.

2.1 Autonomy levels

Based on their functionalities, AD systems are categorized into different autonomy levels, according to the official Society of Automotive Engineers (SAE) classification (Automotive Engineers 2018). ADAS systems, supporting drivers, are on the lower end of the scale. They enhance safety and convenience through features such as Adaptive Cruise Control (ACC), Lane Keeping Assist (LKA), and Automatic Emergency Braking (AEB). These systems provide partial automation and require driver supervision and engagement. Whereas higher levels of automation aim for full autonomy without the need for driver intervention. These advanced levels of automation enable the vehicle to perform all driving tasks under constrained specific conditions up to any conditions and environment.

The six SAE levels of autonomy are presented in Figure 2.1, starting from Level 0, which is no automation whatsoever, to Level 5, which is full autonomy. At each level, a significant technological gap could be observed with respect to previous ones (Galvani 2019).

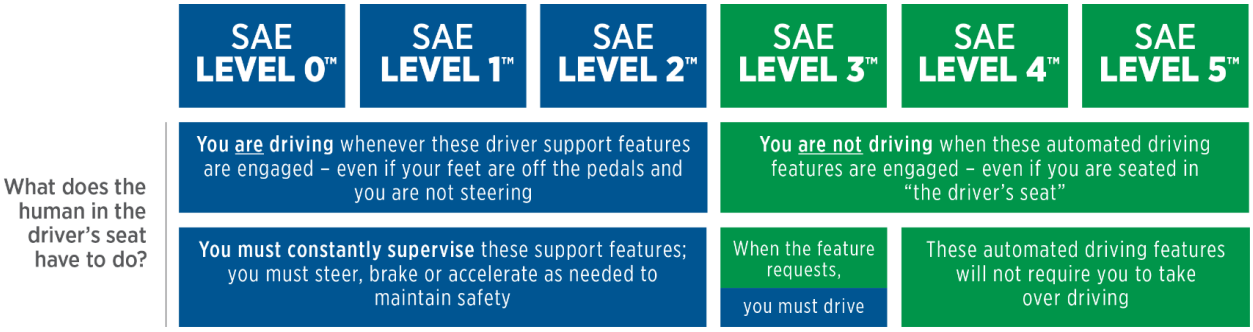

Figure 2.1: Classification of automation levels in AVs according to SAE. There are 6 distinct levels, from no automation to fully autonomous driving level. Source (Automotive Engineers 2018).

With Level 0, there is no automation involved, and the driver has complete control over the vehicle at all times. In this traditional driving mode, the driver is responsible for all aspects of operating the vehicle, including acceleration, braking, steering, and maintaining awareness of the road and traffic conditions. Level 0 vehicles do not incorporate any advanced driver-assist features or automation technologies. There are, however, monitoring systems which help the driver such as parking sensors, surround view cameras, Traffic Sign Recognition (TSR), Lane Detection (LD) and Lane Departure Warning (LDW), Blind Spot Detection (BSD), and Forward Collision Warning (FCW). TSR shows the driver information about current speed limits and other road rules and warnings. LD and LDW warn if the driver is accidentally leaving the current lane. BSD alert the driver if there is an obstacle in the blind spot of the rear-facing mirrors, which he may have overlooked. FCW alarms when there is a chance for an imminent collision with an obstacle ahead. Although those systems are still optional, not forced upon car makers by any laws, they help the driver and already increase safety by a large margin. To that end, they become more popular, even in entry-level vehicles.

At Level 1, there is a limited presence of automation in the vehicle. This level introduces basic driver-assist features that can provide assistance in specific aspects of driving. Examples of Level 1 features include ACC and LKA. With ACC, the vehicle is equipped with sensors that can detect the speed and distance of the vehicle ahead. The system then adjusts the host vehicle's speed accordingly, maintaining a safe distance. This feature can help reduce driver fatigue and enhance comfort during long highway journeys. LKA helps the driver stay within his lane by using sensors to monitor the vehicle's position relative to lane markings. If the vehicle starts to drift out of its lane, the system can provide steering inputs to guide the vehicle back towards the centre. This feature assists drivers in maintaining better lane discipline and can enhance safety, especially on highways or during moments of distraction. Presented systems provide quality-of-life improvements, which are bound to a more premium car class. It is also important to note that at Level 1, the driver still maintains overall control and responsibility for the vehicle. The automated features serve as aids to assist the driver but do not fully take over the driving tasks.

Level 2 introduces a higher degree of automation compared to Level 1. The vehicle is capable of providing combined assistance in steering, acceleration, and braking, which allows for more dynamic control and ease of driving. Level 2 automation includes features such as Highway Assist (HA) or Autonomous Parking Assist (APA). HA, which is a combination of ACC, LKA and BSD, is continuously controlling the vehicle longitudinally and laterally. It is capable of overtake manoeuvres if the driver request so, by turning the indicator on. It accelerates and steers to perform the task, but also avoids collision by monitoring the blind spots during the execution and aborting the procedure if necessary. APA assists the driver both in finding a suitable parking spot and in driving the car into it. Some variants of this system require the driver to use acceleration, and only provide the steering, while others could perform the whole parking task autonomously. Since those systems use a wide variety of sensors and sophisticated software, they are usually included in top-shelf premium vehicle models each car maker has to offer. Even though Level 2 automation provides more advanced assistance features, it still requires the driver to remain engaged. The driver must constantly monitor the road and be prepared to take control if the system requires an intervention.

At Level 3, the vehicle possesses conditional automation capabilities. This means that the vehicle can manage most driving tasks under specific conditions and environments, allowing the driver to disengage from actively controlling the vehicle for certain periods. While the driver does not need to constantly monitor the road, they must be prepared to take control when prompted by the system. The vehicle is capable of managing the driving tasks autonomously, but the driver may need to take control in scenarios such as construction zones, complex intersections, adverse weather conditions, or when the system requests manual intervention. The transition of control between the automated system and the driver is a critical aspect of Level 3 automation. The driver needs to be alert, responsive, and able to regain control of the vehicle quickly when necessary. Therefore, Driver Monitoring Systems (DMS) are typically employed to ensure the driver's readiness and ability to take over control. Level 3 automation represents a significant step towards more advanced AD, offering increased convenience and allowing the driver to engage in non-driving activities during certain periods. Only a handful of current production vehicles are capable of Level 3 automation.

Level 4 signifies a high level of automation where the vehicle can perform all driving tasks under specific conditions and environments without the need for human intervention. At this level, the vehicle operates independently and can navigate through various road scenarios, including complex urban environments and highways, without the constant presence of a human driver. Unlike Level 3, Level 4 automation does not require the driver to be constantly ready to intervene. The system assumes full responsibility for driving and can handle most situations without human input. Despite Level 4 automation showcasing advanced capabilities, it still has limitations. The Operational Design Domain (ODD) defines the specific conditions and environments in which the vehicle can operate autonomously. If the vehicle encounters a scenario outside its ODD, it may require human intervention or be unable to operate autonomously. Therefore, achieving widespread Level 4 autonomy requires extensive mapping, infrastructure support, and addressing complex edge cases to ensure the system's reliability and safety. Level 4 represents a significant milestone in AD, bringing us closer to fully autonomous transportation. However, there are still ongoing efforts and challenges in refining the technology, addressing regulatory frameworks, and establishing public trust and acceptance before Level 4 AVs become customary on the roads. Such high-level autonomy vehicles operate only as proof of concept and demonstration cars, not yet available in mass production.

Finally, Level 5 represents the highest level of automation, where the vehicle is fully capable of performing all driving tasks under any conditions and environments without the need for human intervention. At this level, there is no requirement for a human driver to be present inside the vehicle. It eliminates the need for pedals, a steering wheel, or any controls typically used by human drivers. The vehicle is entirely autonomous and can navigate through complex city streets, highways, rural areas, and even off-road terrains. The system can make complex decisions in real-time, adjusting speed, and direction, and handling any driving scenario that arises. Level 5 represents the pinnacle of AD, where vehicles become self-sufficient, transforming transportation as it is now. That being said, despite much research and engineering efforts, we are still far away from the technology, which could allow for this level of automation. Currently, there is no vehicle which is capable of achieving full AD potential. While the realization of Level 5 autonomy may still require significant advancements and collaboration across various research domains, it holds the potential to revolutionize mobility, making transportation safer, more efficient, and accessible for all.

While there are prototypes of AVs that have achieved higher autonomy levels, it is important to note that the majority of current production cars available in manufacturers' portfolios are categorized as Level 0, Level 1, Level 2 or Level 2+ automation. Level 2+ represents a generation beyond Level 2, which falls short of meeting the requirements for Level 3 autonomy. In terms of technical advancements, some of the challenges associated with Level 3 automation have been addressed. However, it is crucial to consider the regulatory aspects and legal frameworks specific to each region. These regulations need to be carefully examined and amended to authorize the operation of such vehicles on public roads. Resolving these regulatory barriers is an essential step towards the widespread adoption and deployment of higher-level AVs.

2.2 Perception systems

Perception, in the context of ADs, can be defined as a component of an AV which collects and interprets information about its surroundings to gain an understanding of the environment (Skruch et al. 2022). It involves the integration of data from various sensors, such as cameras, LiDAR, Radar, and the creation of a comprehensive environmental model that captures the important aspects of the vehicle's surroundings.

Perception is a fundamental element of AD, forming the foundation for all higher-level features and functionalities based on the environment model (Gruyer et al. 2017). The perception system provides necessary information about the environment, such as OD, LD, or TSR. These components serve as the building blocks for complex functions like path planning, decision-making, and control, which in turn execute driving assist features such as LKA, AEB, or ACC. Therefore, perception is an essential component, which links together the raw sensor data and the intelligent decision-making capabilities of AD systems.

Perception in AVs includes two major elements: sensors and algorithms (Pandharipande et al. 2023). These elements work together to enable the vehicle to sense and interpret its environment. Sensors play a crucial role in perception by capturing data from real-world surroundings. Different types of instruments are utilized to convert different physical readings into digitized information that can be processed by algorithms. These algorithms form the backbone of perception systems. They process the data to extract meaningful information and incorporate it into the model of the surroundings. Through computational techniques, such as signal processing, Computer Vision, and Machine Learning, these algorithms are capable of analysing the collected data and identifying relevant objects, features, and patterns in the environment. Different automotive sensors used in AVs are presented in the next section, whereas data processing algorithms are described in Chapter 3.

2.3 Automotive sensors

Sensing devices, namely camera, LiDAR, and Radar, are essential for the vehicle's perception system setup, as they bridge the gap between the real-world environment and its digital representation. In this section, an introduction to automotive grade sensors' specifications is presented, highlighting their importance in meeting safety requirements and discussing their respective inputs to AV's perception with corresponding strengths and weaknesses.

Sensors convert real-world signals into digital readings (Kocić, Jovičić, and Drndarević 2018). They act as the primary interface between the vehicle and its surrounding environment. They capture various signals and transform them into a format that can be processed and interpreted by perception algorithms. Through this conversion process, they provide broad information about the environment, including objects' presence, distances, velocities, and other relevant characteristics. By digitizing the real-world signals, they enable the perception system to analyze and understand the captured data, ultimately allowing the vehicle to perceive its surroundings and make informed decisions.

Safety requirements have a profound impact on the development and construction of all automotive sensors (Ray 2019). These requirements drive the advancement of devices that are highly accurate, reliable, and resilient to adverse conditions. For example, instruments used in AVs must have sufficient resolution, range, and precision to capture data samples with a high level of certainty. They should also be designed to withstand harsh environmental factors such as extreme temperatures, vibrations, and moisture, ensuring their consistent performance in real-world scenarios. Moreover, safety requirements mandate redundancy and fail-safe mechanisms of such devices (Kohn et al. 2015). Redundancy involves the use of multiple sensors of the same type or ones with different modalities to cross-validate the data and enhance reliability. Fail-safe mechanisms ensure that the failures are detected and appropriately handled, minimizing the risk of falsified perception.

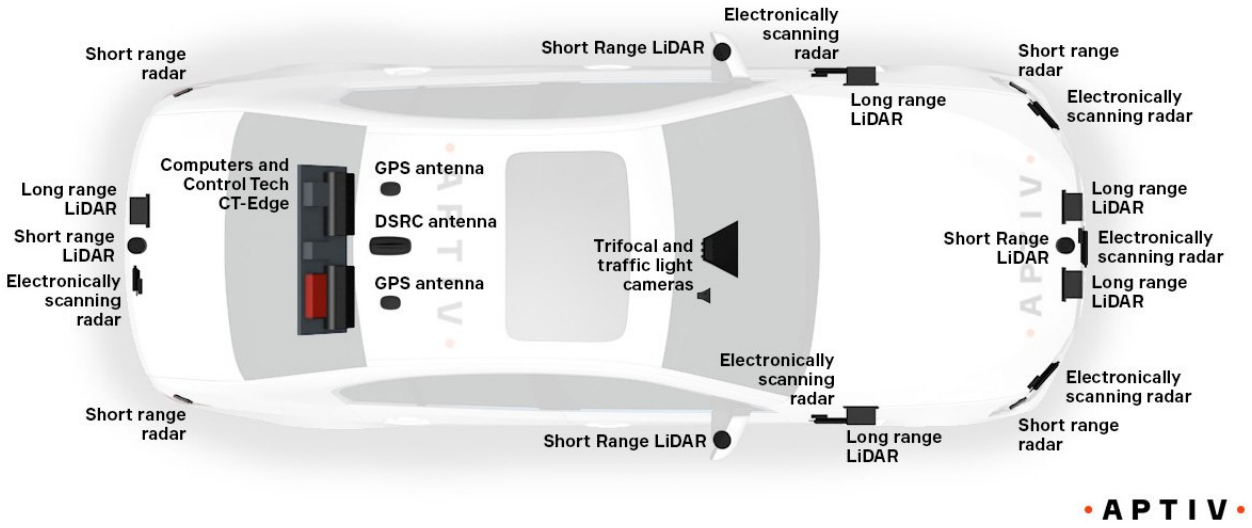

Figure 2.2: Sensor suite of a highly automated AV. It consists of multiple redundant instruments of different types such as cameras, LiDARs, and Radars, located all around the car to provide sufficient coverage. Source (Aptiv 2022).

In Figure 2.2, a modern AD vehicle is presented with its comprehensive sensor suite. Among the vast variety of useful devices, the commonly used automotive sensing equipment comprises cameras, LiDAR, and Radar. Cameras, with their ability to capture visual data, are important for recognizing and classifying object types and shapes. LiDARs offer detailed 3D information about the environment. Providing precise depth information they help in estimating the distance to objects. Radars excel in detecting objects at longer ranges and are useful even in adverse weather conditions. Each sensor type employed in perception systems for AVs exhibits unique advantages and drawbacks, which are described thoroughly in the following sections.

2.3.1 Camera

When it comes to driving a car, cameras can be seen as analogous to human sight in terms of their role in perceiving the environment. Just as the human eye provides vital visual information to the driver, cameras serve as the "eyes" of an AV or ADAS system. Similar to human sight, they help in identifying and interpreting various elements on the road. They capture images of the surroundings, allowing the vehicle's perception system algorithms to analyze and understand the scene.

By using multiple cameras thoughtfully placed around the vehicle, it becomes possible to create a 360-degree view, simulating the panoramic awareness around the host vehicle. In addition, these sensors have the ability to provide real-time feedback. By continuously capturing and processing visual information at high speeds, they enable rapid decision-making by the vehicle's control system. This real-time aspect is crucial for timely responses to dynamic situations, such as detecting sudden obstacles or reacting to rapidly changing traffic conditions. To that end, cameras not only exceed human vision in terms of panoramic awareness and real-time responsiveness, but they also possess the ability to capture visual information at a complexity that surpasses human capabilities. They can quickly adjust to changing lighting conditions, with proper setup exceeding in night-vision scenarios, and track multiple objects simultaneously, making them invaluable tools for enhancing safety and efficiency in AVs.

A digital camera captures and converts light waves into electronic signals, leading to the creation of a digital representation in the form of an image. A general overview of top-level elements of this process is presented in Figure 2.3. It begins with incoming light waves passing through a set of various lenses (Eytan and Belman 2019), that focus it on an imager. The imager is a photosensitive pixel array that detects the incoming light and measures its intensity. The photons that strike the pixels create an electrical charge proportional to the amount of light received. Cameras employ various technologies, such as CCD or CMOS

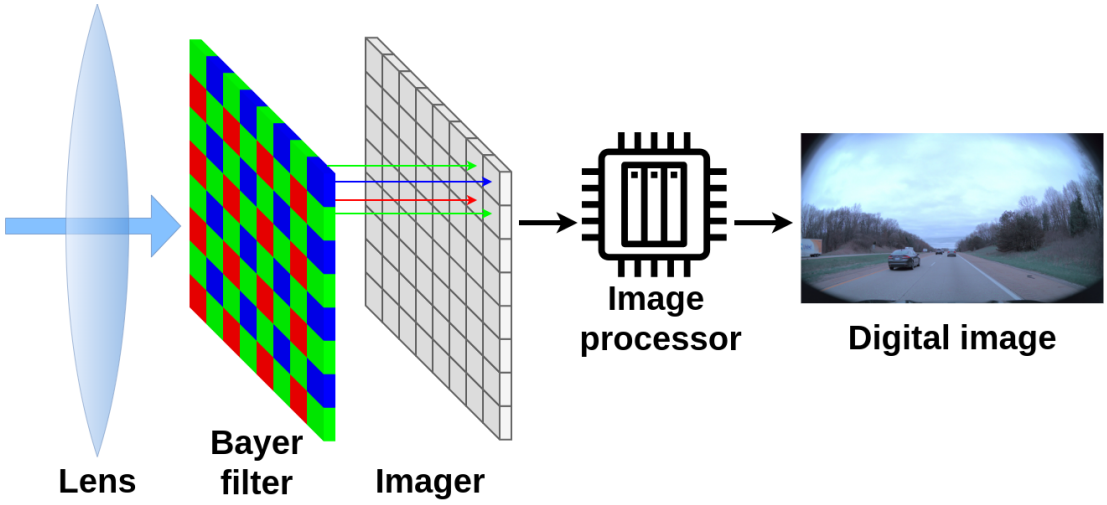

Figure 2.3: Cameras general working principle. The light waves pass through cameras' lenses and Bayer colour filters to reach CCD or CMOS sensors' imager. The physical signal is then converted to a digital value of coloured pixel intensity. The image processor gathers information from all pixels in the sensor array and outputs the final image.

(Luštica 2011), to convert this electrical charge into digital signals. This conversion process assigns numerical values to the electrical charges, creating a digital representation of the captured light intensities. Each pixel's digital value corresponds to its respective location in the image, forming a grid of intensity information.

Furthermore, cameras often employ colour filter arrays on top of the imager (Nakamura 2005), such as the Bayer filter, to capture colour information. The standard array usually consists of RGB (Red-Green-Green-Blue) filters in a mosaic pattern over the pixels, allowing each pixel to record a specific colour component. By capturing colour information for each pixel and interpolating the missing colour values, sensors generate full-colour images with realistic hues and tones. However, in automotive cameras, other colour filters are used. One such alternative is the RCCC (Red-Clear-Clear-Clear) filter (Pawłowski, Piniarski, and Dąbrowski 2021). In this configuration, every four pixels have one red and three clear filters, omitting the green and blue ones. The RCCC alternative can enhance the sensitivity to red light and also helps with low-light intensity scenes such as night-vision use cases (H.-W. Huang, C.-R. Lee, and H.-P. Lin 2017). Another variation is the RYYCy (Red-Yellow-Yellow-Cyan) filter (Lelowicz, Jasiński, and Piłat 2022). This array includes one red, two yellow, and one cyan filter for each set of four pixels. The RYYCy solution aims to precisely distinguish between white and yellow road lines and, at the same time, to maintain good performance in low-light quality scenarios. RYYCy also improves the TSR feature (Weikl, Schroeder, and Stechele 2020).

Camera calibration is another step required for accurate and reliable measurements in imaging systems. It involves two main aspects: extrinsic calibration and intrinsic calibration. Extrinsic calibration is concerned with determining the camera's position and orientation with respect to the host vehicle (Kuruba et al. 2022). This calibration process involves capturing images of known calibration patterns from different angles and positions. By analyzing the detected features on this calibration pattern, such as corners or markers, and comparing them to their known locations, the camera's position and orientation can be accurately estimated. This information is essential for applications that require precise localization or accurate mapping of the captured images in the real world, such as OD. On the other hand, intrinsic calibration focuses on the internal parameters of the camera itself. These parameters include the focal length, principal point, and lens distortion correction (Lelowicz 2019). The focal length determines the camera's Field-of-View (FoV) and affects the scale of the captured images. The principal point defines the optical axis's intersection with the image sensor, providing a reference point for measurements. Lens distortion, caused by imperfections in the camera lens, can introduce unwanted geometric distortions in the captured images. Lens distortion correction (Gonzalez-Aguilera, Gomez-Lahoz, and Rodriguez-Gonzalvez 2011) is particularly important as it ensures that objects in the image are represented in their true form, without any geometric deformations. By applying intrinsic calibration, and distortion correction, measurements and analyses performed on the images can be more accurate, allowing for precise object localization, dimensional measurements, or other quantitative assessments.

Figure 2.4: A sample from the automotive-grade camera. The key aspects of the presented image are wide FoV, which enables the capturing of more information from a single frame, and strong distortion, visible especially in the corners of the image. Sample from the industrial partner company demo car test drive.

Figure 2.4 shows an example of an automotive-grade camera sample. The format of such an image depends on various aspects, including resolution, the FoV and distortion, based on the optical lens used, as well as compression type for a final digital representation of an image. The resolution of image data is determined by the imager size in pixels. Higher-resolution sensors capture more pixels, resulting in finer detail and higher image fidelity. There is however a trade-off between resolution and speed of image processing. The choice of resolution for a particular optical path depends on the desired level of detail and processing capabilities. FoV is another important characteristic influenced by the lens used in the camera system. Different lenses offer varying FoVs, ranging from wide-angle lenses that capture a broader scene to lenses that zoom in on distant subjects. The FoV affects the composition and perspective of the image, enabling the system to capture the desired visual coverage. Different lenses are used in surround 360-degree parking cameras and others in front-facing cameras for highway features. Additionally, the distortion can impact the accuracy and fidelity of the captured image. This could be observed especially in the sample image corners. Finally, the image data can be used in different formats, such as raw or compressed. Raw image formats retain all the original data captured by the sensor without any lossy compression or processing. These formats preserve the maximum amount of information and offer greater flexibility in post-processing, allowing for precise control over factors such as white balance, exposure, and colour grading. The raw data format is often required by safety regulations for automotive-grade sensors. However, it typically results in larger file sizes, requiring more computing resources and longer processing time.

Automotive cameras in perception systems capture high-resolution images with accurate colour representation. This allows for detailed analysis of object features, textures, and shapes, providing valuable information for object recognition, scene understanding, and tracking. Additionally, with proper calibration, they can precisely estimate the lateral offset or position of objects in the scene. This information is valuable

for tasks such as LD, OD, and TSR, enabling spatial awareness during environment modelling. However, these sensors also have limitations for perception tasks. Traditional cameras provide 2D images, lacking explicit depth information. While detected shapes and perspective can provide some depth perception, precise distance measurements require additional depth-sensing technologies, such as RGBD (Red-Green-Blue-Depth) cameras or LiDAR. These depth readings are particularly useful for tasks like distance estimation for 3D models of the environment. Cameras also heavily rely on the availability of visible light to capture images. Adverse weather conditions like heavy rain, fog, or low lighting can severely impact image quality, reducing visibility and deceiving perception algorithms.

2.3.2 LiDAR

LiDAR (Light Detection And Ranging) is another remote sensing technology that can be utilized in the perception systems of AVs and ADAS. It enables these systems to accurately perceive the surrounding environment by measuring distances to objects using laser pulses. LiDARs emit laser beams and measure the time it takes for the light to bounce back after hitting an object. They provide reliable and precise depth information, which is then used to create a detailed 3D map of the environment surrounding the vehicle.

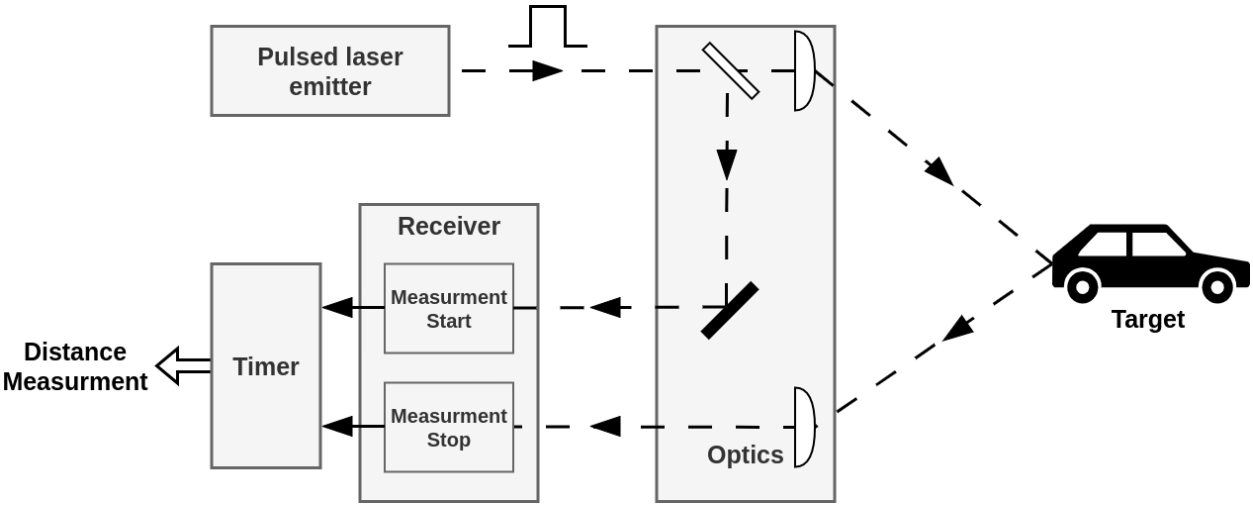

Figure 2.5: LiDAR Time-of-Flight working principle. The sensor emits laser pulses, which bounce back from targets. The backscattered pulses are detected by an optical path within the sensor, and based on measured ToF, the distance to the object is calculated.

LiDARs operate based on the principle of Time-of-Flight (ToF) (Warren 2019), presented in Figure 2.5. They emit laser pulses, typically in the near-infrared range, and measure the time it takes for the reflected light to return to the sensor. By knowing the speed of light and ToF of particular measurements, the distance to the object can be calculated. In order to generate a comprehensive 3D representation of the surroundings, LiDARs utilize scanning mechanisms. These mechanisms allow the laser pulses to be directed and scanned across both horizontal and vertical FoV. The scanning can be achieved through various methods, depending on the type of the sensor.

Rotating LiDARs employ a spinning mechanism to change the direction of emitted and received laser pulses (Roriz, Cabral, and Gomes 2022). These sensors typically consist of a laser emitter and receiver mounted on a rotating platform. As the platform rotates, the laser emits pulses in different directions, covering a full 360-degree horizontal FoV. The receiver captures the reflected light, and by knowing each beam's precise elevation and azimuth, as well as platform rotation angle it can calculate the position of the reflection in a 3D Cartesian coordinate system. By combining the data obtained from each rotation, the sensor creates a complete 3D representation.

In LiDARs based on the Micro-Electro-Mechanical System (MEMS) mirror technology (Thakur 2016), micro-mirrors integrated into silicon chips are utilized to steer the LiDAR electromagnetic beam. These micro-mirrors can change their inclination along two different rotation axes, enabling them to scan the horizontal and vertical FoV. MEMS-based LiDARs offer significant advantages for automotive perception systems. With their compact size and integration capabilities, they enable easier integration into vehicles. Additionally, their cost-effectiveness due to mainstream silicon foundries makes them more accessible. While not achieving a full 360-degree FoV, MEMS LiDARs provide sufficient coverage in specific directions. Moreover, by utilizing the setup of configurable mirrors, MEMS LiDARs can generate various density data in different regions of FoV for precise OD in the most crucial regions. Although there are no moving parts at a macro scale, this technology still uses mechanical mirror positioning at a micro level, which may still lead to sensor physical degradation and eventually malfunction in harsh automotive conditions.

A novel LiDAR scanning technology, called Optical Phased Array (OPA) (Hsu et al. 2021), offers what is often referred to as a completely solid-state device. As a promising alternative to traditional mechanical scanning mechanisms, the OPA employs an array of tiny optical elements, such as waveguides, to electronically steer the laser beams without any moving parts. The OPA method employs light splitters and phase shifters to manipulate the phase delays of signals in the antennas, enabling precise control over the steering of the emitted beams. By adjusting the phase of the signals, the OPA LiDAR system can steer the laser beam in a specific direction, facilitating targeted scanning of the desired FoV.

The output data format of LiDAR, commonly referred to as a pointcloud, provides a detailed representation of the surrounding environment in the form of a list of reflected measurements. An example of such a pointcloud is presented in Figure 2.6. Each measurement point is characterized by its X, Y, and Z coordinates, indicating its spatial position relative to the LiDAR. Additionally, pointclouds may include intensity values, representing the reflected or back-scattered laser energy, providing further information about the properties of the objects or surfaces. The number and the quality of points in a pointcloud are influenced by several LiDAR device characteristics, such as FoV, range, resolution, and accuracy (Bastos et al. 2021). The FoV determines the angular extent of the environment captured by the sensor. It influences the width and height of the pointcloud, defining the coverage range in both horizontal and vertical directions. The range refers to the maximum distance at which it can accurately measure distances to objects. Another aspect, the resolution, describes the level of detail or precision in capturing spatial information. Finally, the accuracy in LiDAR refers to the precision and correctness of the measured distances.

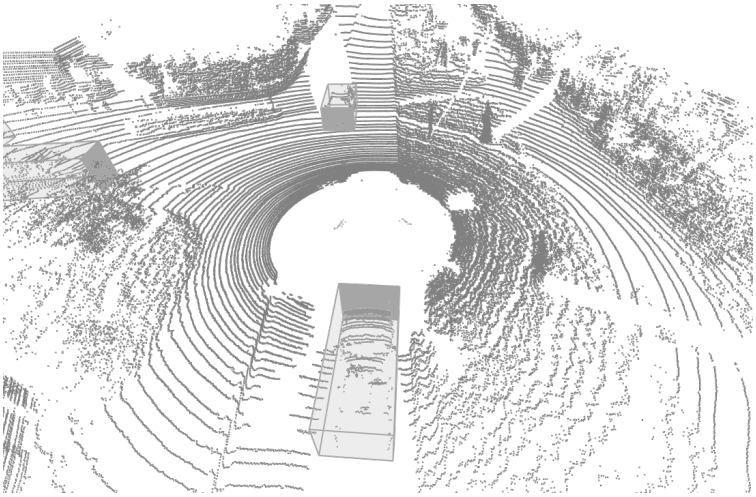

Figure 2.6: Visualization of LiDAR pointcloud data sample. 3D pointcloud showcases the remarkable capabilities of the sensors to very accurately map the surrounding environment. Aside from other vehicles present in the scene, pedestrians on the sidewalk could also be recognized within the pointcloud.

LiDARs offer several distinct advantages in perception systems. One of the key benefits is their ability to explicitly provide highly accurate distance measurements. This accuracy is crucial for tasks such as OD in 3D space. Another significant advantage is their capability to generate comprehensive 3D pointclouds. By scanning the surroundings in a wide FoV, these sensors capture detailed spatial information, facilitating a better understanding of the environment in the perception system. They also offer the advantage of providing groundtruth data for development purposes (Schalling, Ljungberg, and Mohan 2019). The accurate and reliable measurements obtained from LiDARs are often used as reference data for algorithm development, testing, and validation of perception systems.

However, there are certain limitations associated with these sensors that need to be considered. Rotating LiDARs are costly devices, not suitable for mass automotive production from a financial point of view. Additionally, their potential drawback is the risk of mechanical failures in the hardware, particularly in mechanical scanning systems with rotating components. A solid-state LiDAR technology mitigates this, but on the other hand, it is still under development and lacks the required range for automotive use cases (Hsu et al. 2021). LiDARs can also be susceptible to weather conditions such as rain, fog, or excessive sunlight (Wallace, Halimi, and Buller 2020). Adverse weather conditions can interfere with the laser beams and affect the quality and reliability of the data obtained from the sensors. However, advancements in LiDAR technology have led to the development of weather-resistant sensors that mitigate these issues to a certain extent. Lastly, automotive LiDARs are certified with laser class 1 as safe for the human eyes, according to the IEC 60825-1 standard (Dai et al. 2022). However, the interference can occur when multiple LiDARs are used simultaneously in close proximity. The overlapping laser beams can create conflicts and result in interference, potentially impacting the accuracy and performance of the perception system. Proper sensor placement, synchronization, and signal processing techniques (G. Kim, Eom, and Park 2015) are necessary to mitigate interference issues in multi-LiDAR setups.

2.3.3 Radar

Radars have become a common component in ADAS and AV perception systems due to their unique advantages and capabilities. Radio Detection And Ranging (Radar) is an essential technology for perception systems, as it enables them to sense the environment beyond what the human eye can see. While the underlying general principles of measuring the distance to objects are similar, Radar and LiDAR devices diverge significantly in their technological implementations. Radars emit radio waves, instead of infrared light waves, and measure the time it takes for the waves to bounce back after reflecting from an object (Hakobyan and Yang 2019). By analyzing the reflected waves, they can determine not only the distance and angle of detected objects but also their velocity.

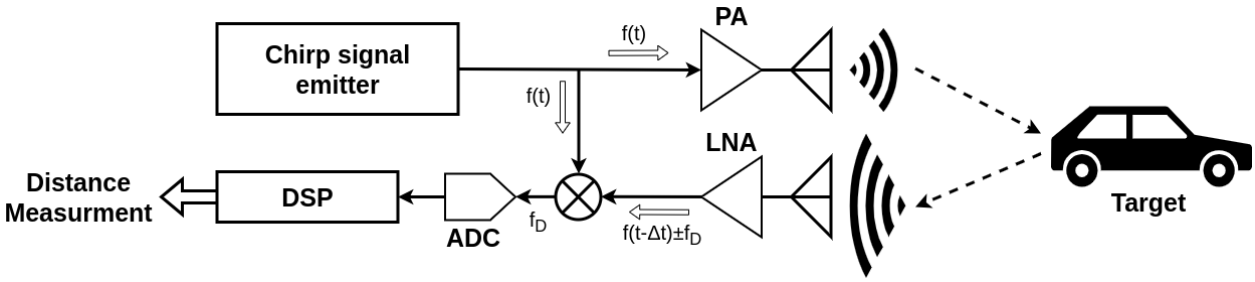

Figure 2.7: Radar working principle. The signal produced by the chirp signal generator goes through a power amplifier and is emitted into the environment. Objects reflect radio waves, which are received back and boosted by a low-noise amplifier. By comparing both signals time difference and Doppler shift f_D are calculated, which are used to determine the position and velocity of the detection.

Radars operate based on the principles of radio wave propagation, echo detection, and signal processing (Patole et al. 2017), which is presented in Figure 2.7. In the automotive domain, a common Radar technique is Frequency Modulated Continuous Wave (FMCW) (Waldschmidt, Hasch, and Menzel 2021), where a transmitter emits continuous waveforms known as chirp signals that vary in frequency over time. These chirp signals are emitted through a specific frequency range, which in automotive standards are typically 24 GHz for short-range Radars and 77-81 GHz for long-range ones (Wenger 2005). When encountering an object, the frequency modulation caused by the Doppler effect (Winkler 2007) occurs, altering the transmitted signal. Specially designed receiver antennas (Menzel and Moebius 2012) capture the back-scattered waves, which contain the frequency-modulated echoes. By comparing the transmitted and received frequencies, the Doppler shift can be determined, providing information about the relative velocity between the sensor and the object (Kok and Fu 2005). Measuring the time delay between the transmitted and received signals allows for calculating the range or distance to the object. The amplitude of the received signal provides information about the object's Radar Cross-Section (RCS), which relates to its size and reflectivity.

Sensor data readings from the Radar devices are similar to previously discussed LiDAR pointcloud, however, there are distinct differences between the two of them. The comparison of both pointclouds is shown in Figure 2.8. Each point within a Radar pointcloud contains several key attributes. The most fundamental ones are the 3D coordinates of the point, indicating the position of the object in the sensor's FoV.

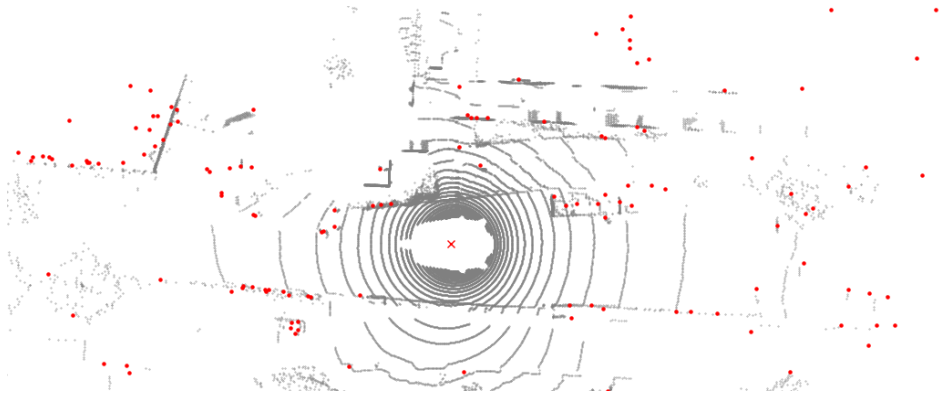

Figure 2.8: Visualization of Radar pointcloud sample data marked in red. In comparison to grey LiDAR pointcloud, the Radar sample is significantly smaller in terms of the number of detections. Detections from Radar are also irregular, due to the reflectivity of radio waves for certain surfaces. However, they can penetrate obstacles and detect occluded objects, which is not possible with LiDAR.

Additionally, Radar pointclouds carry information related to the dynamics of the detected objects, namely the velocity of the detections (W. Yu et al. 2018), which indicates their speed and direction of movement. Furthermore, Radar pointclouds often provide information about the RCS of the detected objects, which describes an object's reflectivity or how strongly it reflects Radar signals. Aside from these primary attributes, Radar pointclouds can also include supplementary information such as uncertainty measures, object identification tags, or confidence scores associated with each detected object.

Radars offer several advantages when used in AV's perception systems. A very significant one is their extended range capabilities, allowing them to detect objects at long distances. This makes Radars well-suited for applications that require a substantial detection range, such as highway driving scenarios. Another benefit is the ability to differentiate between moving and static objects (Hyun, Jin, and J.-H. Lee 2017). By providing accurate velocity measurements, Radars enable the identification of moving objects and their trajectories based solely on a single measurement cycle. This functionality is essential for tasks like object tracking, collision avoidance, and ACC. Radars are also relatively insusceptible to adverse weather conditions, such as rain, fog, or snow (Sheeny et al. 2021). Radio waves have the ability to penetrate these atmospheric conditions, making Radars reliable in various weather scenarios where other sensors like cameras or LiDARs may be impaired.

However, one limitation of these sensors is the sparsity of data they provide (Scheiner, Schumann, et al. 2020). Radar pointclouds typically contain fewer data points compared to LiDAR. This sparsity restricts the ability to capture fine-grained details and complex object structures, which can be important for certain perception tasks that require high-resolution data. Furthermore, Radar signals can face challenges with interference from other Radar sources or environmental factors, leading to inaccuracies in detected object properties like range or velocity measurements (Umehira et al. 2020). Specialized algorithms and techniques are required to mitigate interference and extract reliable information from the Radar's data (Uysal and Sanka 2018).

2.4 Sensor data fusion

Sensor data fusion combines readings from individual sensors, processes them, and assembles a new comprehensive representation to be used in a perception system. The goal of the fusion is to utilize the most valuable information each of the sensors can provide and, at the same time, mitigate corresponding weaknesses. As stated in (Wald 1999), there are many different fusion definitions, such as:
"Data fusion is a set of methods, tools and means using data coming from various sources of different nature, in order to increase the quality (in a broad sense) of the requested information." or
"Data fusion is a multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from multiple sources.".
Based on their research, the authors proposed a broader description of what fusion is, namely:
"Data fusion is a formal framework in which means and tools for the alliance of data originating from different sources are expressed. It aims at obtaining information of greater quality; the exact definition of 'greater quality' will depend upon the application.".

Generally, sensor fusion encompasses the integration of data from various sources to generate cohesive and dependable information (Kocić, Jovičić, and Drndarević 2018). This becomes vital when combining diverse data types into one functional system. Fusion serves to mitigate uncertainty and noise inherent in individual sensor measurements by capitalizing on their complementary strengths and compensating for limitations. Furthermore, it amplifies the redundancy and accessibility of sensor data by offering multiple information sources for the same event. Ultimately, sensor fusion can also empower the extraction of fresh features and insights that individual sensors cannot provide.

Sensor fusion offers several benefits that significantly enhance the capabilities of AVs perception systems (Herpel et al. 2008). One of the key advantages of sensor fusion is enhanced coverage. By combining information from multiple sensors that cover the same FoV, the system can obtain a more comprehensive and detailed understanding of the environment. Each sensor has its strengths and weaknesses, and by fusing their data, those limitations can be mitigated. For example, a combination of cameras and Radars can provide a broader spectrum of information, including visual data, depth measurements, and object velocity (Wei et al. 2022). Cameras can detect and classify objects well, but lack of depth estimation is a problem for a perception system. They are also susceptible to adverse weather conditions such as rain or fog. On the other hand, Radar sensors are not influenced by that and provide accurate depth measurements. At the same time, due to its sparse nature, object classification is rather unobtainable from the Radar pointcloud. By fusing the two sensors' data, a complementary input to the perception system could be obtained. Another benefit of sensor fusion is increased confidence in the detected objects (Lindner et al. 2007). When multiple sensors detect the same object, redundancy is introduced, enabling a safety check mechanism. If the information from one sensor is unreliable or compromised, the system can rely on the data from other sensors to validate and cross-reference the detection. This redundancy improves the system's robustness and reduces

the chances of false positives or false negatives in OD. By considering multiple measurements, the system can make more informed decisions, enhancing safety and reliability in autonomous operations.

However, sensor data fusion also presents its fair share of challenges (Lindenmaier et al. 2022). Synchronization of data from different sensors is crucial to ensure accurate fusion. Various sensors may have varying sampling rates, measurement formats, and coordinate systems, making synchronization complex. Another challenge lies in fusing data from different domains, such as visual, range, and temporal ones. These domains have distinct characteristics, and harmonizing them requires careful calibration and transformation techniques.

Sensor data fusion can occur at different steps of the algorithm (K. Huang et al. 2022), depending on when the information from a single sensor is merged with the other. In the following sections, the detailed classification of fusion methods is presented with respect to these two primary factors, fusion level and fusion stage.

2.4.1 Fusion level

Sensor fusion methods are often categorized into High-Level Fusion (HLF) and Low-Level Fusion (LLF) based on the moment in the processing pipeline when the fusion algorithm is applied. This division helps distinguish between approaches that involve higher-level processing and those that primarily focus on raw sensor data integration.

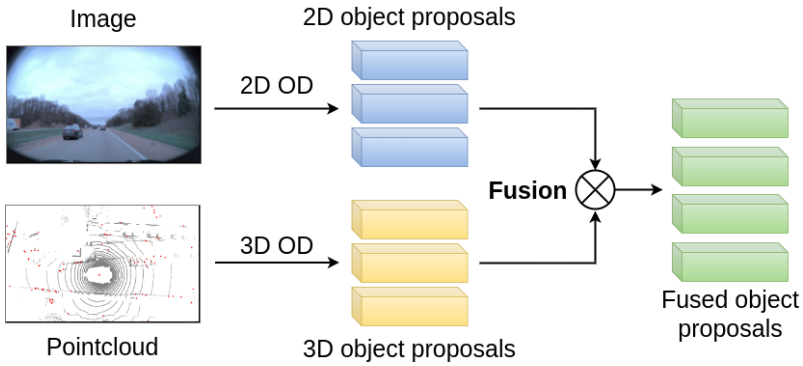

Figure 2.9: High-Level Fusion concept diagram. The main HLF aspect involves separate 2D and 3D OD for each input data. Consequently, the fusion occurs at an object level, where both objects' proposal lists are merged to form a final fused outcome.

HLF, also called object-level fusion, focuses on merging predictions from individual sensor algorithms, as shown in Figure 2.9, to form a consolidated and more accurate understanding of objects in the environment (Duraisamy et al. 2016). In object-level fusion, the algorithms from different sensors, such as cameras, LiDARs, and Radars, generate their own predictions regarding the objects present. These predictions are then combined to create a unified representation of the objects, taking into account the strengths and limitations of each sensor. The main advantage of object-level fusion is the ability to leverage the diverse information provided by different sensors in a high-level object list representation. Such representation simplifies sensor readings to detected object instances, where the fusion process becomes a much more explicit task. How-

ever, object-level fusion may face challenges when the predictions from different sensor algorithms conflict or provide inconsistent information. Resolving such conflicts can be complex and requires sophisticated algorithms and decision-making mechanisms to determine the most reliable and accurate representation of the object.

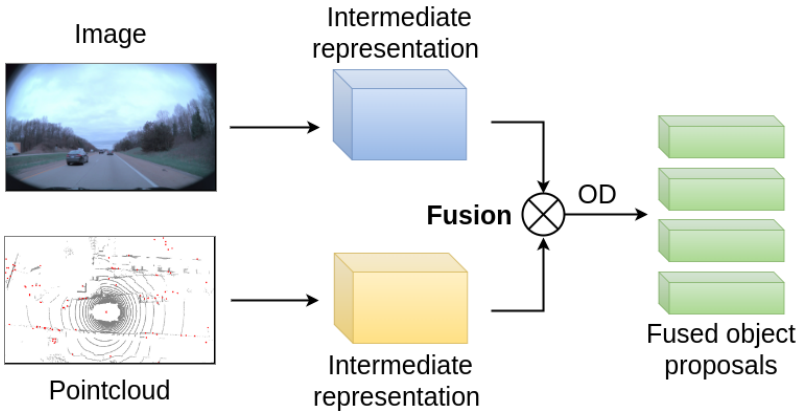

Figure 2.10: Low-Level Fusion concept diagram. The main difference in LLF, compared to HLF, is that fusion occurs at the intermediate data representation level, rather than the object level. Fused representation forms an input to OD algorithm, which produces object proposals.

LLF, presented in Figure 2.10, operates close to the raw sensor data level, where measurements from different sensors are combined to generate a unified representation. This fusion level aims to enhance the accuracy and reliability of individual sensor measurements (Pollach, Schiegg, and Knoll 2020) before applying algorithms such as OD. LLF improves the quality of sensor data by compensating for sensor-specific limitations, reducing noise, and improving calibration. It is particularly useful in tasks that rely on accurate sensor measurements. LLF-based perception is computationally efficient and less sensitive to processing errors. However, it may not capture higher-level contextual information and may struggle with complex scene understanding. Additionally, synchronization and calibration challenges between sensors can still arise at the LLF.

2.4.2 Low-Level Fusion stage

When exploring various LLF methods, it is possible to further categorize them based on a specific division that provides additional characterization. This categorization distinguishes between early fusion and late (or feature) fusion, based on the timing and nature of the data combination (Scheunert et al. 2007).

In the case of early fusion, the LLF operates directly on raw sensor readings, immediately combining them after the sensor output. This approach involves the fusion of the unprocessed sensor data, enabling a direct integration of the raw measurements. Early fusion captures the raw information from multiple sensors and merges it at an early stage, allowing for comprehensive integration of the original sensor readings. An example of such fusion can be the method, where the camera image is extended with projected pointcloud data forming an additional RGBD depth channel in the data, as shown in Figure 2.11. However, different sensors may have variations in their data characteristics, such as resolution, noise levels, or calibration

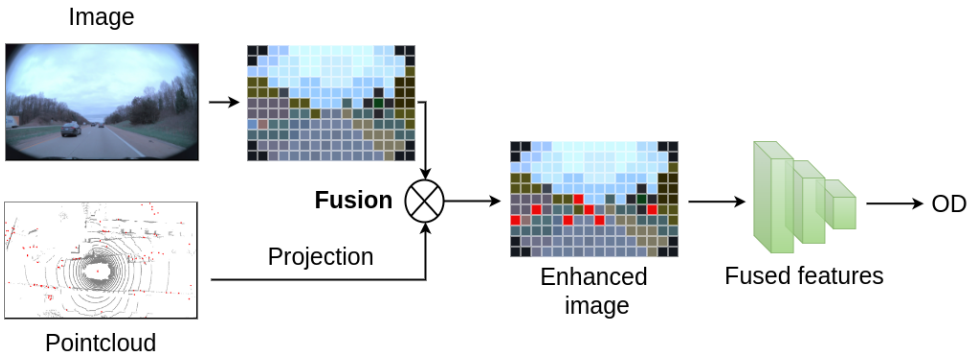

Figure 2.11: Early fusion approaches operate at the data level, where a fused representation is generated directly from raw data with minimal preprocessing. In the illustrated case, the camera image is enriched by incorporating projected pointcloud detections. Subsequent steps involve feature extraction and OD based on enhanced image representation.

differences. Integrating raw data from diverse domains can be challenging and may require techniques to align or adapt the data to ensure meaningful fusion.

In contrast, late (or feature) fusion involves the preprocessing of the sensor data, where an internal representation in the form of features is extracted before the fusion takes place. Such technique is presented in Figure 2.12. In this approach, the LLF occurs after the sensor data has undergone some form of processing, often referred to as feature extraction. Late fusion focuses on integrating the higher-level abstractions from the sensor data, which provides a more refined representation of the underlying information. The difference between late fusion and HLF is that extracted features still encapsulate all or most of the raw sensor data information in opposition to the predefined OD structure, which is a result of each sensor perception algorithm in HLF. When compared to early fusion, by extracting features and fusing them at a higher level, late fusion reduces the impact of individual sensor errors or discrepancies, improving the overall reliability of the fused results. It also allows for more flexibility in incorporating different sensors or replacing specific sensor modalities. As long as the extracted features are compatible, late fusion can easily accommodate changes in the sensor setup or integration of new sensors.

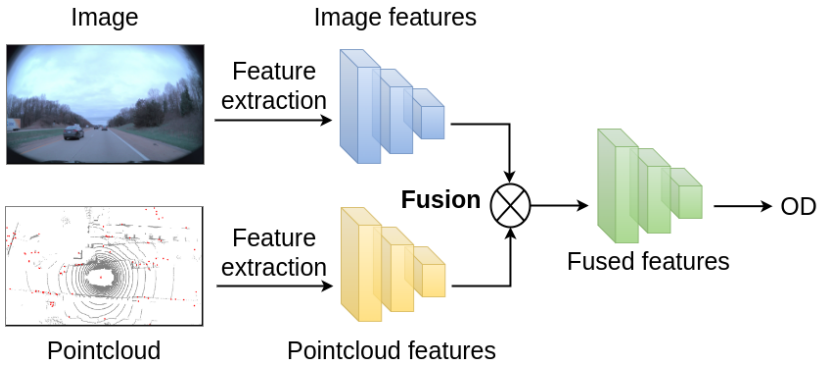

Figure 2.12: In late fusion, data features are integrated at a more abstract level. In the provided illustration, both image and pointcloud data undergo feature extraction, converting raw data into an intermediary representation. Fusion takes place at the feature level, where the two sets of features are combined to form a unified feature representation for subsequent OD task.

Chapter 3

Deep Learning in perception systems

Chapter highlights:

- *General overview of Neural Networks*
- *Literature review of single-sensor and fusion NN solutions for AV perception systems*

In the previous chapter, the various AVs' sensors have been presented, which provide valuable data about the surrounding environment. However, raw sensor data is not sufficient to understand and interpret the surroundings accurately. To extract meaningful information and make informed decisions, data processing algorithms are required. Among a variety of algorithms used in AV perception systems, artificial NNs have emerged as the most effective and widely adopted approach in recent years (Shafiee et al. 2021). This thesis specifically concentrates on perception based on NNs, as they are gaining more and more popularity in both scientific and industrial circles. NNs, a sub-field of ML, are widely used in AVs' perception systems due to their ability to process large amounts of data and learn intricate patterns (J. Ren, Gaber, and Al Jabar 2021). Those features are essential when dealing with a vast amount of training data and a large variety of different road scenarios involved in the creation of an automotive-grade solution.

In this chapter a ML-based approaches are discussed, focusing on the role of NNs in AV perception systems. A general introduction to NNs theory is done, providing supplementary references to the reader, to enable in-depth study of particular topics. Furthermore, a deep CNNs are described, along with the modern approach to designing such networks. The main section of the chapter presents a literature review regarding current State-Of-The-Art Object Detection model architectures. Considering fast-paced development and advances in that domain, it is challenging to provide a comprehensive recap of all related research. Nonetheless, best to the authors' knowledge, corresponding camera, LiDAR, and Radar single-sensor architectures are reviewed, and the motivation behind the utilization of these models is thoroughly explored, particularly within the context of the fusion. Moreover, the literature examination of fusion architectures for various combinations of sensors is presented, providing a summary of their proposed approaches.

3.1 Neural Networks

The theoretical groundwork of NNs encompasses a vast amount of knowledge accumulated over years of research. Incorporating a comprehensive presentation of this theory into the thesis poses a challenge, as many topics demand intricate mathematical explanations and illustrative examples to present the ideas effectively. However, given the increased NNs recognition, this theoretical framework is widely popular and well covered in numerous books (e.g. Bishop 1995; Suzuki 2011; Goodfellow, Bengio, and Courville 2016; Patterson and Gibson 2017; Aggarwal 2018) and articles. To that end, this section will primarily reference existing works rather than going into foundational concepts from the ground up, with a particular focus on their application in AV perception systems.

An artificial NN is a mathematical model that incorporates multiple nonlinear transformations, seamlessly combined to execute more complex functions. The fundamental unit of such a network is a neuron, the detailed structure of which is described for example in (Bishop 1995). Its key components include weights, determining the impact of each input on the neuron's output, and an activation function, such as Sigmoid, Tanh, ReLU, LeakyReLU, or Mish (Kaloev and Krastev 2021), that introduces non-linearity to the computations. Neurons are organized into layers, forming the overall network structure. Examples of complete models' configurations are presented for instance in (Suzuki 2011). The commonly employed Fully-Connected (FC) layers, establish connections between each neuron in the preceding layer and every neuron in the subsequent layer. However, the architecture of FC layers demands a substantial number of inter-neuron connections, which can prove computationally inefficient when handling high-resolution sensor data. In response to this challenge, the Convolutional Neural Network (CNN) was introduced (Cun et al. 1990). A detailed explanation and comparison of both FC and the convolution layer is provided in (e.g. Patterson and Gibson 2017). CNNs exhibit notable advantages in sensors' data processing by capturing spatial relationships, leading to enhanced performance (Krizhevsky, Sutskever, and Hinton 2012). In the final structure of NN model for OD, the information from the sensors passes through successive layers (during the inference process), which transform it into an internal representation known as data features. Based on these features, the final prediction layers produce the network output, containing objects' predictions.

The weights are the parameters learned by the model during the training process. While various ML methods exist for training models, including Supervised, Unsupervised, and Reinforcement Learning discussed in (e.g. Alpaydin 2010), this thesis exclusively focuses on Supervised Learning, given its popularity for OD tasks. With this approach, the model is trained to map input data to a target distribution of labels from the dataset, while maintaining the generalization ability (Vidyasagar 2010). This training method utilizes gradient-based algorithms, such as SGD, RMSProp or ADAM (Zohrevand and Imani 2022), with the aim of enhancing the model by optimizing the loss function value (Mohri, Rostamizadeh, and Talwalkar 2012). A crucial component of this algorithm is the backpropagation method, described for instance in (Haykin 2009), which provides partial derivatives of the loss concerning each parameter of the model. The training process can be tuned by various hyperparameters as discussed in (e.g. Ravichandiran 2019).

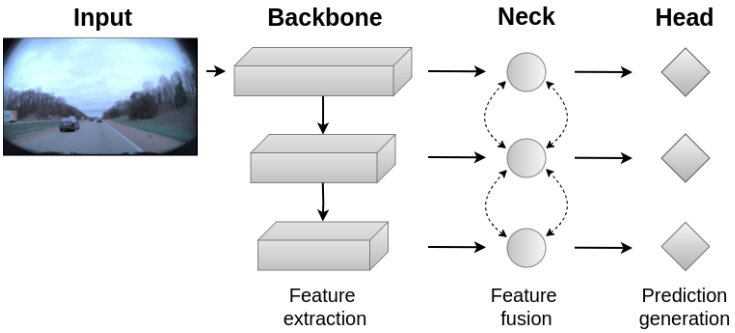

Figure 3.1: The general structure of a deep Neural Network architecture, divided into smaller components: a backbone, neck and prediction head. Each component specializes in a different area required for the model to achieve the overall goal. With such a modular approach, the elements can be exchanged with different solutions realizing the same functionality.

To address complex problems, such as OD, modern Deep NN architectures, described thoroughly for instance in (Goodfellow, Bengio, and Courville 2016), are designed with more layers to extend their capacity in comprehending intricate functions. As networks deepen, challenges like vanishing and exploding gradients emerge (Aggarwal 2018). The introduction of residual connections in (He et al. 2016) resolves these gradient problems by providing alternative "skip connections" during backpropagation. The study in (Szegedy et al. 2015) explores the width of NNs and introduces the inception block, offering multiple pathways for information propagation and aiding gradient flow. These methods allow for extensively deep and wide NN architectures, which increases the models' capabilities.

Considering the extensive range of possibilities available for designing deep networks, a necessity arises for a systematic organization of complex architectural solutions, namely into backbone, neck, and head, as presented in Figure 3.1. The backbone forms the foundational component and is responsible for extracting hierarchical features from input data. It consists of stacked convolution and pooling layers that progressively capture both low-level and high-level features. The neck further refines extracted features, facilitating information compression and fusion. Often designed as bottleneck layers, it can reduce computational demands while maintaining essential information. Finally, the head is the task-specific component responsible for generating the final output. It consists of a relatively small amount of layers, transforming the extracted features into meaningful predictions. Such structured methodology proves to be an efficient approach for crafting large architectures. The division allows for modularity, enabling researchers to experiment with various backbones, necks, and heads independently to improve on a selected component, identify optimal configurations for different tasks and build upon previous work.

Finally, attention mechanisms, introduced in the Transformer model (Vaswani et al. 2017), enhance NNs' ability to model complex relationships in data. Transformer-inspired attention-based architectures like Convolutional Block Attention Module (CBAM) (Woo et al. 2018) and Squeeze-and-Excitation (SE) network (Hu, Shen, and Sun 2018) adaptively emphasize informative features, showcasing the influential impact of attention mechanisms on NNs' architecture design and leading to state-of-the-art results across different domains, including OD in an AV perception systems.

3.2 Single-sensor models

Expanding on the general overview of NN theory, this section provides a literature review examination of popular SOTA model architectures utilized in AV perception systems. Organized by sensor types, the models are categorized into distinct subsections according to their data processing capabilities. In each subsection, the focus is directed towards showcasing the most relevant and innovative methodologies, offering an overall perspective on the current status of research within that specific domain.

Understanding sensor-specific solutions and domain-focused designs for training NN architectures is essential. This process not only reveals how diverse data types are managed but also exposes the advantages and disadvantages of each design, allowing for their direct comparison. Such knowledge is crucial for the review of fusion model architectures. It also forms a basis for the fusion solution proposed in this thesis. The synergistic integration of established single-sensor solutions for feature extraction minimizes reinvention risks and speeds up development by reusing proven architectures and methods.

3.2.1 Camera architectures

An Object Detection task in camera images was the first field, out of discussed AV modalities, to successfully apply Convolutional Neural Network solution (Cun et al. 1990). Ever since then, researchers have been constantly improving the algorithms by applying novel architectures and mechanisms to increase performance. In the context of the AV perception system, camera OD methods can be divided into two major groups: 2D image plane and monocular 3D models.

The 2D OD approaches for camera images can be further split into two-stage and one-stage detectors. Two-stage detectors (Girshick et al. 2014; Girshick 2015; S. Ren et al. 2017) are based on legacy CV image processing methods idea, which uses the sliding window and classification to perform OD. The detection process is divided into two phases. The first step is a region proposal stage, where based on feature maps, promising parts of an image are selected through Region Proposal Network (RPN). In stage two, the propositions are classified, one by one, as specific object types by the detection head of the network. On the other hand one-stage or Single-Shot Detector (SSD) implements the opposite approach. With only a single feed-forward pass all the regions in an image are classified at once, as presented in Figure 3.2. This method tends to be slightly less accurate, but the main advantage is a significant decrease in the inference of a SSD model (Carranza-García et al. 2021). While performance holds significant importance, the crucial aspect for an AV perception model is its inference speed. To that end, in this work, the focus shifts more towards SSDs.

One of the most well-recognized architectures in the realm of 2D SSD is the You Only Look Once (YOLO) network (Redmon, Divvala, et al. 2016). Over time, enhancements have been proposed to augment the initial network performance. YOLOv2 (Redmon and Farhadi 2017) employs an anchor box mechanism, where the prediction of raw bounding box size is carried out relative to predefined anchor sizes that are most suitable for the correlated target. YOLOv3 (Redmon and Farhadi 2018) introduces multi-scale training for small, medium, and large objects at different levels of the NN's Feature Pyramid Network (FPN), which are

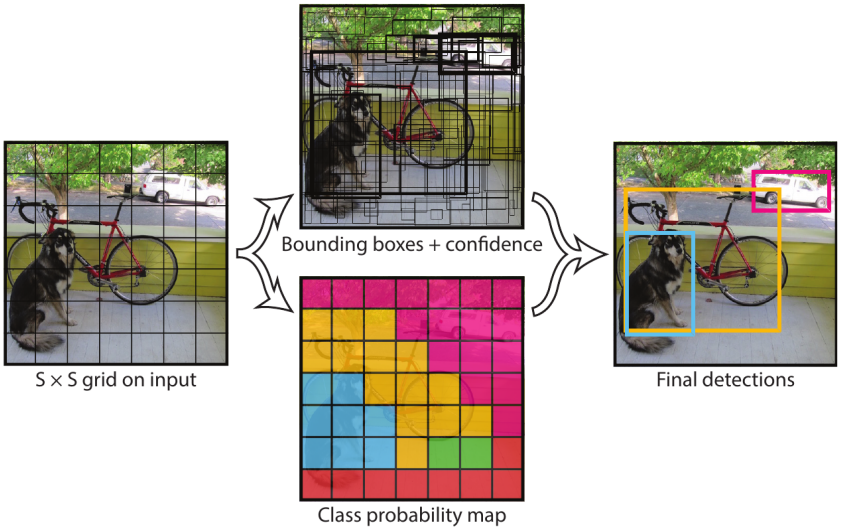

Figure 3.2: The principle of Single-Shot Detector prediction mechanism showcased on YOLO architecture. The input image is divided into grid cells. For each cell classification and regression heads predict the class probability and corresponding bounding box size respectively. A NMS algorithm is used to merge overlapping detections and output the most confident predictions. Source (Redmon, Divvala, et al. 2016).

then combined just prior to the Non-Max Supression (NMS) algorithm. YOLOv4 (Bochkovskiy, C. Wang, and H. M. Liao 2020) refines the network architecture by implementing a Cross-stage Partial Connections (CPS) backbone, Path Aggregation Network (PAN) (S. Liu et al. 2018), attention through CBAM (Woo et al. 2018), a CIoU (Zheng et al. 2019) metric for improved loss calculation, and the Mish activation function throughout network layers. Building on the same SSD concepts, the RetinaNet (T.-Y. Lin et al. 2017) architecture introduces improvements in the form of focal loss. This novel loss function addresses issues related to class imbalance, ultimately enhancing both training speed and stability.

In more recent work regarding the 2D camera OD domain, the CenterNet approach (Duan et al. 2019) has emerged, focusing on estimating key points for the centre of bounding boxes. With this anchor-free technique, detections are represented using triplets of opposite bounding box corners and the central point. According to the authors, this strategy expands upon the SSD model by introducing an additional RoI extraction process, similar to the initial step in two-stage detectors. A comparable solution is presented in the Fully Convolutional One-Stage Object Detection (FCOS) method (Tian et al. 2019). In this approach, the centre of a bounding box is predicted in image pixels as well. However, instead of predicting corners to determine the size, a separate regression head produces width and height values relative to each centre point. It's important to note that these two solutions do not propose new backbones, but rather, they reuse existing ones and even provide results for a diverse array of such.

Given the previously discussed breakdown into distinct components: backbone, neck (FPN), and prediction head, the subsequent research focuses on enhancing a specific element of the overall model. While refining the SSD architecture, researchers examined the concept of model scaling, specifically addressing width, depth, and resolution. In the EfficientNet paper (Tan and Q. V. Le 2019), they introduced a smaller, faster, and yet more effective architecture achieved through a compound coefficient. This coefficient enabled

the design of an optimal architecture from a selection of models with varying width, depth, and resolution parameters. Building upon this notion, they introduced the detection network architecture EfficientDet (Tan, Pang, and Q. V. Le 2020), which employs EfficientNet as its backbone. Moreover, they extended the idea of multi-scale feature fusion by proposing a weighted Bi-directional Feature Pyramid Network (BiFPN) to enhance the propagation of internal network representations across different layers more efficiently. In EfficientNetV2 paper (Tan and Q. V. Le 2021), the optimization of the backbone architecture was taken further to achieve enhanced training speed and improved parameter efficiency in terms of model size. On the other hand, a different backbone architecture is presented in the Deep Layer Aggregation (DLA) approach (F. Yu, D. Wang, and Darrell 2017). The authors suggest a deeper features aggregation to enhance the sharing of fused information across the backbone's layers. This incorporates innovative features aggregation methods, including both Iterative Deep Aggregation (IDA) and Hierarchical Deep Aggregation (HDA). Within a tree-like structure that branches and merges in a skipped connection manner, the DLA facilitates the aggregation of features from distinct levels. Unlike traditional deep architectures, this backbone is characterized by greater width expansion rather than depth.

While transformer-based approaches are currently unsuitable for in-car embedded devices due to their demanding computational needs, this review acknowledges several noteworthy architectures (K. Han et al. 2023). Within this context, the Vision Transformer (ViT) (Dosovitskiy et al. 2021) stands out as a new standard of transformer-based backbone for image processing architectures. ViT processes images in small 16×16 patches, converting them into tokens, similar to text processing models. Another prominent solution is the Shifted-window Transformer (Swin) (Ze Liu et al. 2021), which employs the concept of a small window scanning across the entire image. The self-attention mechanism is then applied to hierarchical features derived from each window. This approach reduces computational complexity and scales linearly with the input image resolution. Both solutions achieve remarkable results in various CV tasks, and with rapid development, they could become feasible for embedded platforms in the near future.

Although accomplishing 3D Object Detection from a single monocular camera image poses a significantly more complicated challenge (Arnold et al. 2019), recent research indicates that specific NN architectures can still yield meaningful results. In the case of CenterNet (X. Zhou, Koltun, and Krähenbühl 2020), an extension of the 2D model (Duan et al. 2019), the proposed approach involves dividing the 3D OD process into two stages. The first step encompasses anchorless prediction of the centre for a given cuboid in the image, while the second step involves regressing all 3D parameters, such as depth, 3D dimensions, and rotation angles. After projecting the predicted centre, the 3D results are derived. Similarly, the FCOS3D approach (T. Wang et al. 2021a) follows a comparable methodology, applying 2D FCOS (Tian et al. 2019) solution to the 3D domain. Authors leverage a predefined set of landmark 3D points within the image to execute 2D centreness prediction. Subsequently, this prediction, in conjunction with 2D position and depth information, is projected into a 3D space. The remaining parameters undergo regression in the 3D space, culminating in the final prediction of the object's characteristics.

An alternate approach involves treating the problem as a depth estimation task based on a single image. PGD (T. Wang et al. 2021b) employs a distribution-based depth estimation for 2D image pixels, incorporating a probabilistic framework to capture uncertainty. Additionally, they leverage a geometric relation graph to enhance the estimation of contextual connections among 3D objects. EPro-PnP (H. Chen et al. 2022) adopts the Perspective-n-Points (PnP) technique, which computes pose solutions from a set of 3D points in object space and their corresponding 2D projections in image space. They propose employing PnP as a differentiable layer, training the network in an end-to-end fashion to learn weighted 2D-3D point correspondences, and interpreting the output as a probabilistic distribution. This distribution can then be utilized for 3D OD. Recent studies demonstrate that transformer-based vision networks can also effectively predict 3D objects from images. One such model is BEVFormer (Z. Li et al. 2022), which learns comprehensive Bird's Eye View (BEV) representations by incorporating both spatial and temporal information via predefined grid-like queries on image features.

3.2.2 LiDAR architectures

The processing of pointcloud data presents challenges when employing NNs. The network must demonstrate invariance to all permutations of input points, altering the order of data in the input list should yield consistent outcomes. Moreover, the length of this list can vary based on sensor readings, yet the fixed input size expectation of NN architectures poses a constraint. Additionally, when dispersed in 3-dimensional space, pointcloud data becomes highly sparse (with up to 95% of the space remaining unoccupied). These challenges have led to two main approaches in pointcloud processing with NNs, namely the pointwise and voxelwise strategies.

The pointwise approach, exemplified by the classification network PointNet (Charles et al. 2017), employs dense transformation layers to extract features for each individual point, with shared weights to ensure order invariance. To achieve global feature extraction invariance, PointNet uses a max pooling layer. Such preprocessed pointcloud is then fed through sets of FC layers, which extract object classification. Another example of a pointwise architecture is PointRCNN (Shi, X. Wang, and H. Li 2019), which employs a two-stage approach for 3D OD. The first stage segments points and generates initial detections. The second stage refines these detections pointwise, using local spatial and global semantic features to assign proper bounding boxes and confidence scores.

When addressing OD problems, voxelwise techniques are frequently preferred. First introduced through VoxelNet (Y. Zhou and Tuzel 2018), this approach involves the scattering of points in 3D space to mitigate computational demands and tackle data sparsity. This is achieved by segmenting the entire 3D space into smaller cuboids known as voxels. Each voxel then becomes the focal point for the calculation of features, executed through the Voxel Feature Extractor (VFE) layers and based on the points encompassed within it. This process can be visualized in Figure 3.3. Subsequent to the feature extraction stage, the resulting fixed-size output tensor is subjected to 3D convolutional layers, ultimately leading to the generation of 3D

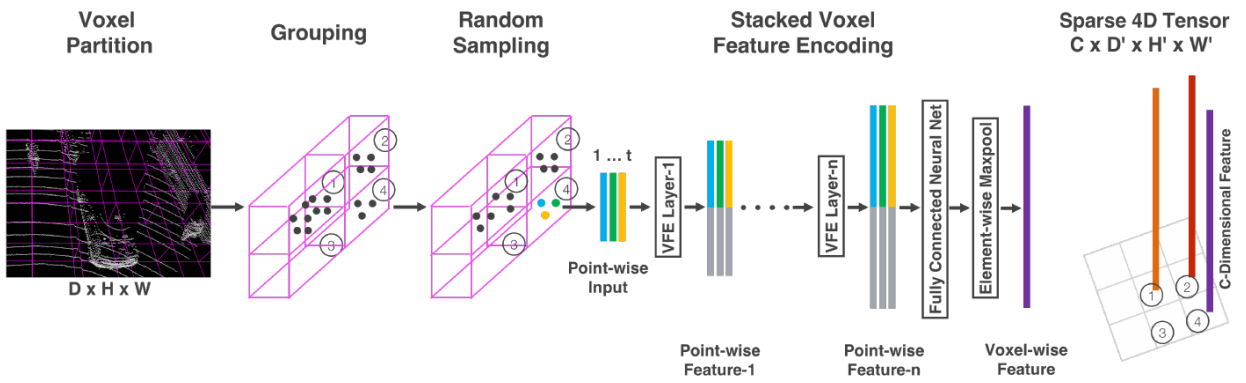

Figure 3.3: The diagram of pointcloud data voxelization and feature extraction. Initial points are assigned to spatially corresponding voxels. Then, all points within the voxel are subjected to VFE processing, which produces general voxel features based on those points. Each voxel is processed separately, and the extracted features are scattered back into a 4D tensor representing the voxel grid. Source (Y. Zhou and Tuzel 2018).

detections. The Pointpillars architecture (Lang et al. 2019) offers a distinct approach to feature extraction within this context. It rearranges voxels in a vertical manner along the z-axis, effectively creating pillars. This modification results in a three-dimensional tensor as the output of the feature extraction, as opposed to the four-dimensional outcome observed in VoxelNet. This alteration paves the way for the utilization of 2D convolutions instead of the more computationally intensive 3D counterparts. Remarkably, this shift significantly accelerates inference times, as demonstrated in the referenced paper, enabling near real-time performance. The Pointpillars innovation showcases the dynamic evolution of voxelwise methodologies in optimizing detection tasks and streamlining computational efficiency.

The work called PV-RCNN (Shi, Guo, et al. 2020) combines both pointwise and voxelwise methods within a single network architecture. In addition to regular voxel feature extraction, PV-RCNN fuses feature maps from voxelwise subnetworks with pointwise features in the Voxel Set Abstraction Module. This fusion generates Keypoint Features, which are subsequently passed to the detection head to amplify specific regions in the output grid, helping with final predictions.

3.2.3 Radar architectures

3D OD ML techniques applied to Radars can be categorized into two groups, characterized by the nature of their input data: raw sensor signal or preprocessed detections pointcloud. The generation of the pointcloud involves antenna signal processing, often incorporating frequency domain operations such as Fast Fourier Transform (FFT) and the application of specific thresholding techniques. As the pointcloud format closely resembles LiDAR one, the same challenges and similar solutions are present when applying NNs to Radars.

Applying CNNs to raw signal data has demonstrated success in object classification tasks (H. Han et al. 2019; L. Wang, Tang, and Q. Liao 2019). With a relatively small number of convolutional layers, features are extracted from the input data, and the final FC layer facilitates the classification of objects. Similar CNN methodologies (Y. Kim et al. 2020; Kwon, S. Lee, and Kwak 2018) are tailored for pedestrian detection

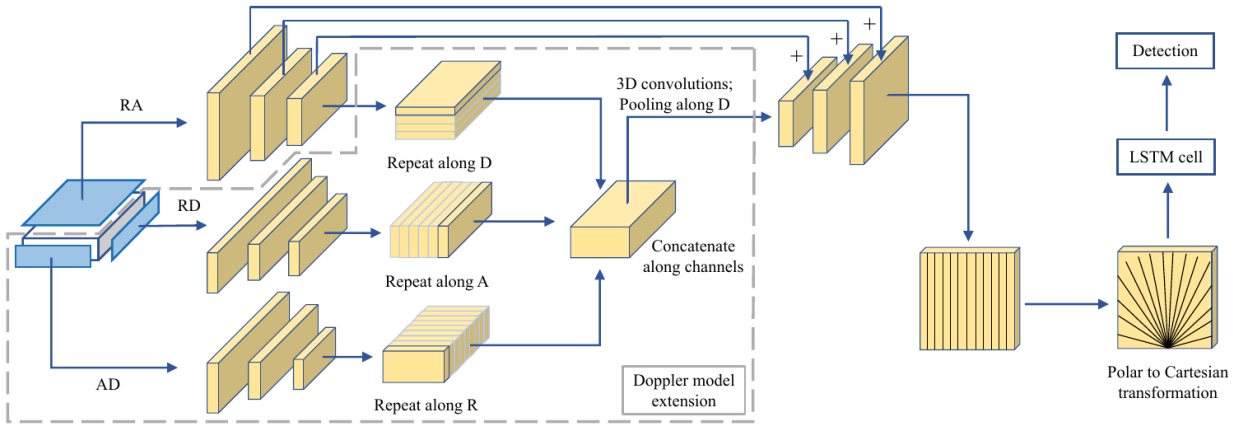

Figure 3.4: Example of raw Radar data processing. The input data forms a 3D "RAD" tensor where respective channels correspond to Radar range, azimuth, and Doppler shift readings. Fully convolutional architecture extracts internal features representation onto the polar coordinate system. Source (Major et al. 2019).

within AV systems. Furthermore, by augmenting the model with an additional recurrent NN module (S. Kim et al. 2018), researchers can track time-series data derived from raw Radar readings. This enhancement empowers the model to learn the dynamics of moving targets, resulting in improved classification performance. While OD presents a more intricate challenge than simple classification, recent advancements (Major et al. 2019) showcase that NNs can also excel in performing OD on raw Radar data signals, organized in a 3D RAD tensor (range, azimuth, Doppler). This approach, outlined in Figure 3.4, involves collapsing one dimension by summing the values along it, thereby generating RA, RD, and AD 2D matrices. These three inputs are separately processed by convolutional layers and ultimately merged through concatenation, resulting in a final representation within a polar coordinate system. Based on this representation, the prediction head generates object proposals.

An alternate approach to OD is demonstrated in solutions such as those presented in (Scheiner, Kraus, et al. 2021; B. Xu et al. 2021; Palffy et al. 2022; Popov et al. 2022), where the network's input consists of a preprocessed Radar pointcloud. Despite employing diverse techniques to generate this pointcloud, a common characteristic is the utilization of a BEV grid as the model's input. Subsequent processing involves method-specific CNN architectures, closely resembling those used for LiDAR data. To address Radar data sparsity and enhance the data sample size, certain methods aggregate detections from multiple timestamps. Although there are limitations and potential information loss in the transformation of raw data into pointcloud representation, this approach offers several advantages. Primarily, processing raw data necessitates specialized hardware setups and Radar devices to be able to obtain such readings. Additionally, the majority of open-source automotive datasets provide processed detections rather than raw data samples. Lastly, the advantage of processing Radar data as a pointcloud has the potential for future fusion solutions that seamlessly integrate images and pointclouds, accommodating both LiDARs and Radars, as they share a similar format.

3.3 Fusion models

ML-based fusion algorithms combine data from distinct sensors to yield enhanced outcomes. Within the domain of AV perception systems, fusion frameworks process image and pointcloud data. Cameras, LiDARs, or Radars perceive the environment in different yet complementary ways, offering advantageous fusion prospects (H.-S. Le, T. D. Le, and Huynh 2022). These fusion architectures often leverage techniques from single-sensor setups to gather information from each sensor's data prior to fusion. This encompasses established backbone architectures, proven FPN structures, or preprocessing methods such as VFE for pointcloud data. While NN-based fusion architectures are capable of both High-Level Fusion and Low-Level Fusion, this literature review concentrates solely on LLF solutions, aligning with the thesis's objectives.

Given the substantial disparities in sensor data domains, ranging from camera image views to BEV grids and pointcloud 3D perspectives, integrating diverse sources of information in fusion becomes a challenge. To address this, researchers have introduced diverse techniques for integrating sensor data within NNs architectures. These methods can be classified based on the processing domain. Depending on whether they focus on a particular view or simultaneously handle them separately until fusion takes place, they fall into the categories of single-view and multi-view methodologies.

3.3.1 Single-view approach

Within the single-view category, methods centre around a specific processing domain and project all other data sources onto that view. Typically, this projection occurs prior to feature extraction through NNs, closer to raw data samples, with minimal preprocessing. To that end, the majority of single-view methods outlined in this section represent an early fusion approach to LLF.

In the fusion approach demonstrated by PointPillars++ (Vora et al. 2020), the authors enhance the front view pointcloud data from LiDAR by integrating corresponding camera pixel information. This combined front view is then projected back to 3D space, subsequently threatening it as an expanded pointcloud, introducing additional features through a NN. This deterministic preprocessing enforces a specific fusion method within the model architecture, thereby enhancing the conditioned utilization of both information sources. The subsequent processing and OD results follow a similar process to the original PointPillars method (Lang et al. 2019) for a LiDAR pointcloud. In the case of Frustum PointNets (Ruizhongtai Qi et al. 2018), the OD process is organised into multiple stages. The image view is supplemented with a sparsely generated depth map, formed by projecting the LiDAR pointcloud onto the image plane. Further modules undertake instance segmentation on this newly fused representation utilizing a frustum mapping mechanism, and the final regression NN predicts estimations for 3D bounding boxes.

Emerging solutions from the drivable area and road detection domain can also be adapted to OD fusion designs. In (Caltagirone et al. 2018), the authors not only project the LiDAR pointcloud onto a camera image but also implement a depth-completion algorithm to generate dense depth maps from sparse front-view points. This augmentation provides additional information to the fusion algorithm. Conversely, an

opposing strategy is presented in (Wulff et al. 2018), where instead of projecting points onto the image, pixel data was projected onto a pointcloud BEV occupancy grid together with LiDAR readings. This unique approach yields results in a distinct fusion domain, yet achieves comparable outcomes.

Beyond camera-LiDAR architectures, the CRF-Net (Nobis et al. 2019) presents promising outcomes in camera-Radar fusion. This approach enriches a camera image by projecting Radar points as extended vertical lines within the image. The OD task occurs within the image space domain, similar to 2D camera architectures, but utilizing additional information from Radars. The authors demonstrate enhancements over a baseline camera-only network. It's important to note that this OD solution operates and predicts objects within a 2D camera image space, rather than a 3D domain.

3.3.2 Multi-view approach

A distinct approach is introduced through multi-view methods. Rather than immediately merging sensor data, dedicated modules extract internal representations of information from data samples collected by each sensor. The fusion process takes place at this higher level of abstraction, categorizing multi-view methods as late LLF solutions. In this approach, the fusion aspect occurs in an end-to-end manner, leaving the method of merging detailed information to the distribution of network weights learned during training. While this approach doesn't guarantee the full utilization of fusion information, it offers the advantage of the NN architecture capturing complex data interconnections. Without strict projections or rigid handcrafted fusion techniques, the architecture's ability to capture intricate relationships between extracted features could potentially lead to improved fusion results, as complex relations are discovered during the optimization process.

In notable precursor multi-view configurations, such as MV3D (Xiaozhi Chen et al. 2017) and AVOD (Ku et al. 2018), each sensor input is individually processed by a dedicated subnetwork to generate view-specific feature maps. These views commonly encompass a BEV, a front pointcloud view (3D points projected onto the camera plane), and a camera view. The process of acquiring sensor-specific features often aligns with the single-sensor architectures discussed earlier. By utilizing concatenated feature maps, the fusion's Region Proposal Network defines Region of Interest for subsequent detection heads, which then predict the final OD proposals.

PointFusion (D. Xu, Anguelov, and Jain 2018) employs the PointNet model (Charles et al. 2017) for LiDAR data and ResNet (He et al. 2016) backbone for image feature extraction, obtaining internal representations of each dataset. The approach introduces dual paths for dense (local) and global feature fusion, merging into a fused representation for subsequent OD prediction. A slightly different method is suggested in Joint 3D Object Detection (G. P. Meyer, Charland, et al. 2019), where custom single-sensor CNNs, also based on the ResNet backbone, extract feature maps for each sensor. Camera features are further enhanced through a novel projection and warp module onto a BEV grid. The enriched view is then processed by LaserNet LiDAR pointcloud architecture (G. P. Meyer, Laddha, et al. 2019), which predicts both 3D OD

and semantic segmentation outcomes. In 3D-CVF (Yoo et al. 2020), VoxelNet (Y. Zhou and Tuzel 2018) and ResNet are employed for pointcloud and image data, respectively. The novel aspect is the introduction of an Adaptive Gated Fusion Network module, merging feature sets from distinct domains. With the help of dedicated fusion RPN, final OD proposals are generated. Multi-Task Multi-Sensor Fusion (Liang et al. 2019) expands the concept of camera-LiDAR fusion to multiple tasks. Separate backbone feature extraction and late feature fusion are applied to predict various outputs, including 2D OD in the image, 3D OD, depth completion for the camera view, and pointcloud semantic segmentation and mapping.

Initial attempts at fusing camera and Radar data are outlined in (M. Meyer and Kuschk 2019), where authors introduce a multi-view network architecture. This architecture involves distinct submodels for feature extraction from each sensor, followed by late feature-level fusion. While their results were less impressive in comparison to camera-LiDAR fusion techniques at that time, they attribute this performance gap to factors like limited training Radar data and sensor-specific differences in pointcloud data between LiDAR and Radar. However, in (Hwang et al. 2022), authors employ more advanced backbone structures to extract features from sensor data. They also propose a 2D to 3D projection technique for both the camera image plane and BEV grid to address missing dimensions needed for 3D fusion. Their approach yields significantly improved results, closing the performance gap to LiDARs. Nonetheless, there remains room for further enhancements.

Among more recent solutions, currently considered SOTA, the fusion predictions in 3D space are obtained in CenterFusion (Nabati and Qi 2021) architecture. The fusion is done on a camera image, processed similarly to the CenterNet (Duan et al. 2019) approach, and Radar detections. First, 2D centre points and object features are predicted in the image, which are then associated with extracted Radar features via the Frustum Association Mechanism. The fusion of two sensor feature maps leads to final 3D predictions. In FUTR3D (Xuanyao Chen et al. 2022), authors proposed a framework to fuse camera images with both LiDAR and Radar pointclouds. They employ a query-based Modality Agnostic Feature Sampler to fuse all sensor features and accommodate a transformer-based decoder to predict 3D objects directly. In BEVFusion (Zhijian Liu et al. 2022), the fusion of image and pointcloud features takes place on a unified BEV grid. To accomplish image-to-BEV conversion of image feature maps, the authors use discrete depth distribution predictions of each pixel. Through the scatter of each feature pixel into discrete points along the camera ray, they create a camera features pseudo pointcloud tensor, which could be directly merged with LiDAR feature maps. Lastly, a transformer-based TransFusion (Bai et al. 2022) is a LiDAR-camera fusion approach that utilizes convolutional backbones and a transformer decoder-based detection head. The decoder includes initial bounding box prediction from LiDAR data followed by adaptive fusion with image features, employing a soft-association mechanism to handle challenging image conditions. The transformer's attention mechanism allows the model to dynamically decide which information from the image should be integrated, resulting in a resilient and efficient fusion strategy. The method also incorporates an image-guided query initialization strategy to address the detection of challenging objects in the pointcloud.

Figure 3.5 comprises all the methods presented in this literature review regarding NN model architectures for OD task in AV perception systems. These single-sensor and fusion methodologies form a basis for the novel approach to the fusion architecture, described in Chapter 5.

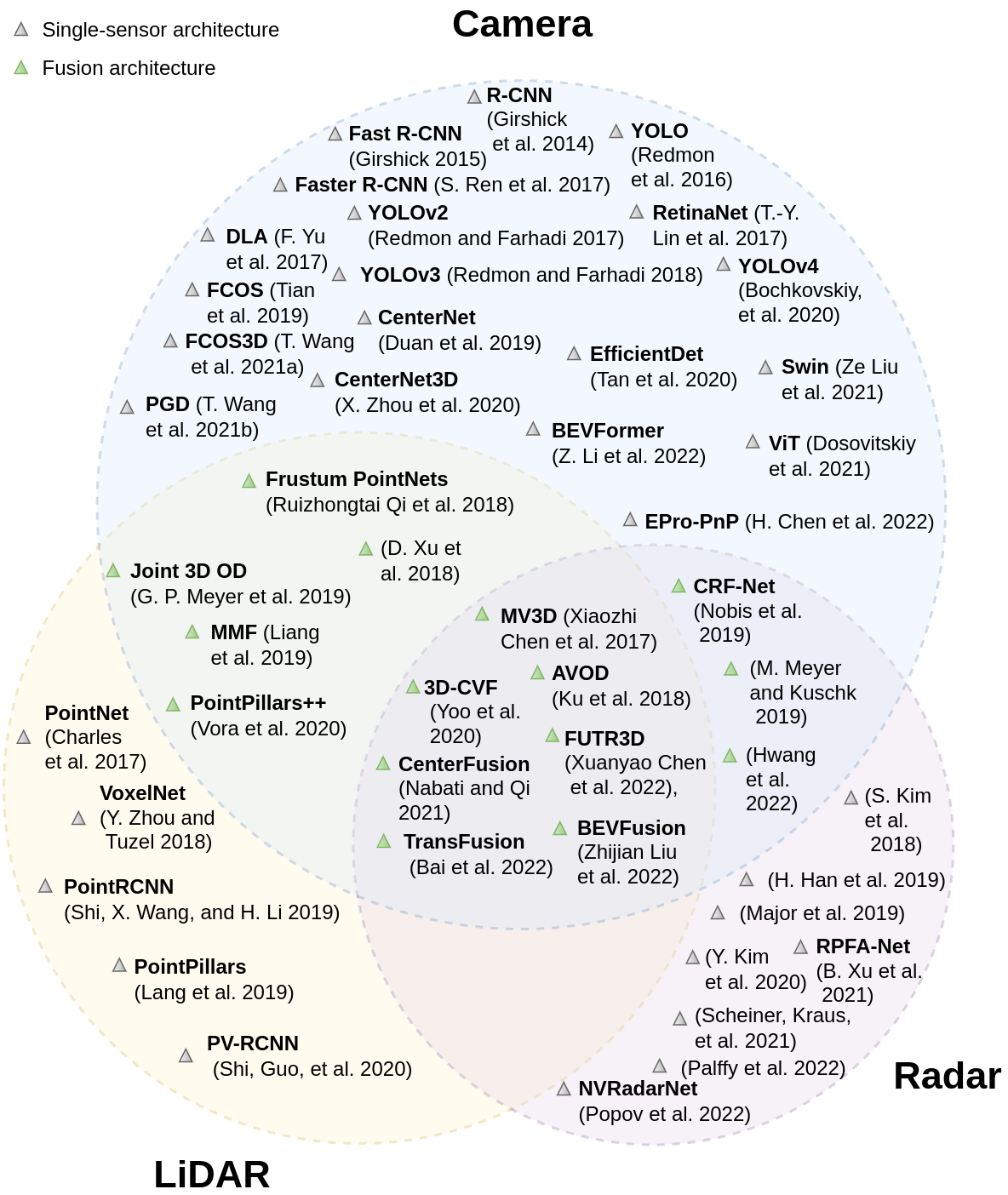

Figure 3.5: The summary of the literature reviewed on single-sensor and fusion NN-based solutions for OD perception in the AV perception systems. Single-sensor solutions are marked in grey, while fusion architectures are marked in green. Each solution is placed in the respective sensor circle. Consequently, fusion methods are positioned on the camera-LiDAR or camera-radar intersections, while general multi-sensor fusion solutions are in the middle, where all three circles overlap.

Chapter 4

Evaluation approaches

Chapter highlights:

- *Labels and predictions association methods*
- *Overview of KPI metrics for 2D and 3D OD*
- *Introduction to XAI, Grad-CAM in particular*

The accuracy and reliability of the perception system components play a critical role in AV application. To ensure their effectiveness, it is essential to employ suitable evaluation methods. Such an evaluation process goes beyond simply confirming that the technique is functioning correctly but it should also provide a quantitative measure of its performance, particularly in adverse automotive scenarios. Nevertheless, the OD field has well-established evaluation metrics that are widely used to assess the capabilities of these systems in the context of a NN model performance. The evaluation methods, also referred to as OD Key Performance Indicator (KPI) metrics, include precision, recall, F1 score, and Mean Average Precision (mAP). By employing these well-established KPIs, the performance of OD models can be quantitatively assessed. Such a quantitative approach not only aids in benchmarking different solutions but also helps in tracking the progress and improvements of models over time. Moreover, these metrics facilitate the identification of specific strengths and weaknesses in the perception system's components, guiding further refinements and enhancements. Within this thesis, the KPI metrics can also be used to assess single-sensor and fusion solutions, highlighting the differences in each approach and leading to a comprehensive comparison between them.

The KPI metrics are extensively described in the following chapter. Firstly, the connection between model predictions and target labels is established. Then the terms of true positive and false positive detections are defined based on selected association methods, corresponding to different OD domains. Subsequently, the core of the evaluation process, which involves OD metrics, is presented. Each KPI method is explained in detail. Additionally, a set of complementary methods specifically focused on evaluating 3D OD performance is introduced. Towards the end of the chapter, the possibility of applying Explainable AI (XAI) methods to OD models is explored, as an alternative to the KPI metrics discussed.

4.1 Association methods

The assessment of OD KPI metrics involves evaluating the differences between the characteristics of labelled objects and the predictions made by the model. However, before conducting this comparison, it is necessary to establish the correspondence between each label and its related prediction by considering the two lists of objects derived from data annotation and model prediction processes. The pairing procedure is determined based on the chosen association criteria, which should be selected according to the specific OD domain. In Figure 4.1, the two most popular association criteria are presented, namely Intersection-over-Union (IoU) and distance-based (DIST) methods.

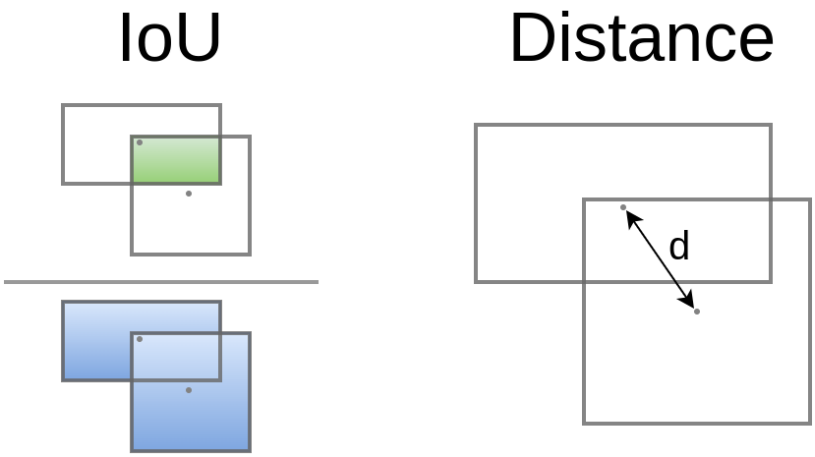

Figure 4.1: IoU and distance-based association methods. The former determines the association by examining the ratio between the intersecting (green) and the union (blue) areas of two bounding boxes. In contrast, the latter method relies on the measurement of distance, represented as d, between the centres of the bounding boxes and is evaluated against a predefined threshold.

The IoU method is a fundamental and widely used technique for establishing the association between two bounding boxes, particularly when dealing with labels and predictions in the 2D image domain. The main principle underlying the IoU method revolves around understanding the relationship between the areas covered by these bounding boxes in relation to each other. The IoU value could be calculated based on the following formula:

$$IoU = \frac{Inter(bb_{label}, bb_{pred})}{Union(bb_{label}, bb_{pred})} \tag{4.1}$$

where the numerator $Inter$ function represents an intersection area or volume of two bounding boxes bb_{label} and bb_{pred}, highlighted in green, while the denominator $Union$ function signifies the total area or volume encompassed by both bounding boxes, marked in blue. The IoU value, which varies from 0.0 (indicating complete separation) to 1.0 (indicating exact overlap), is used to measure the degree of overlap between the two boxes. This association method offers a significant advantage as it not only considers the relative position of labels and predictions but also takes into account their sizes. Commonly used thresholds for IoU include 0.2, 0.5, and 0.8, where positive association occurs when the calculated similarity is above given value. The notation of those IoU-based methods is IoU20, IoU50, and IoU80 respectively.

In contrast, the usage of IoU in 3D OD is less common. This is due to challenges arising when considering 3D volumes rather than 2D areas, leading to significant IoU declines caused by slight misalignments and differences in 3D bounding box positions and sizes. Lowering the IoU threshold to address this issue is a viable solution, but at the same time, the range for positive associations is significantly limited. Instead, a more relaxed distance-based approach is often employed for 3D OD associations, matching label-prediction pairs based solely on the Euclidean distance between their centres. The DIST value is calculated according to the formula:

$$DIST = \|c_{label} - c_{pred}\| \tag{4.2}$$

where $DIST$ metric value represents the Euclidean distance between c_{label} and c_{pred}, the Cartesian coordinates of bounding boxes centres in 2D or 3D space. The L2 norm, by definition, can range from 0, when the centres overlap, to infinity. However, for association purposes, a threshold is utilised to filter out label-prediction pairs that are too far apart. Different threshold values used, such as 0.5m, 1m, 2m, or 4m, are denoted as DIST0.5, DIST1, DIST2, and DIST4, respectively. The DIST association is less demanding, offering more flexible matching conditions than IoU. This flexibility allows for more frequent positive matches, for which the supplementary metrics, such as position, size, and orientation errors could be calculated, as discussed later on. These additional metrics provide deeper insights into the model's performance.

Table 4.1: Different possibilities of the association outcome of prediction and label bounding boxes. "Pos" denotes the existence of the given bounding box and "Neg" means that either it is missing or the association criteria was not met.

Pred \ Label	Pos	Neg
Pos	TP	FP
Neg	FN	TN

Finally, based on the association of label and prediction pairs, each pair can be categorized into one of the four following cases, as shown in Table 4.1, namely true positive (TP), false positive (FP), false negative (FN), or true negative (TN). TP occurs when the model correctly predicts an object that has a matching ground truth label, satisfying the selected association condition. TP indicates a successful object prediction by the model. FP represents cases where the model predicts an object without a corresponding ground truth label. FP constitutes unwanted behaviour, as the model detects something that is not present in the actual data. On the opposite end, FN occurs when the model fails to detect an object that exists in the ground truth annotations. The resulting unpaired target label is also referred to as missed detection. Additionally, when the model detects an object, but the association criteria are not met, it results in both FP detection for the prediction without the matching target and missed detection (FN) for the corresponding label. Lastly, TN denotes instances where the model correctly identifies the absence of an object that is indeed not present in the ground truth annotations. Analyzing these associations helps evaluate the model's performance, by further calculating OD KPI metrics.

4.2 Object Detection metrics

The association and assignment of labels and predictions into the four mentioned categories form the basis for calculating KPI metrics. This thesis employs a range of different metrics to evaluate the performance of both single-sensor and fusion OD models. The fundamental metrics used include precision, recall, and F1 score, which can be applied to any model generating true positive, false positive, and false negative results. For a more comprehensive evaluation of OD performance, a specific metric known as Mean Average Precision (mAP) is employed. Furthermore, considering the additional features in 3D OD compared to 2D, a set of supplementary metrics is introduced, including Mean Average Translation Error (mATE), Mean Average Size Error (mASE), and Mean Average Orientation Error (mAOE), alongside the combination of those metrics in NuScenes Detection Score (NDS).

Precision is a metric that measures the proportion of correctly predicted positive instances relative to all instances predicted as positive. In the context of OD, precision indicates how many of the predicted bounding boxes truly contain objects among all the predicted bounding boxes. The formula for calculating the precision score is as follows:

$$precision = \frac{TP}{TP + FP} \tag{4.3}$$

where TP and FP are the total number of true positive and false positive detections respectively. The precision metric value is within $\langle 0.0, 1.0 \rangle$ range. High precision signifies that the model has a low number of false positives, suggesting that the detections made by the model are mostly accurate.

Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances relative to all actual positive instances present in the ground truth data. For OD, recall shows how well the model can identify and detect all the objects present in the given dataset. It is given by the formula:

$$recall = \frac{TP}{TP + FN} \tag{4.4}$$

where TP and FN are the total number of true positive and false negatives detections respectively. Same as precision, its value fall in $\langle 0.0, 1.0 \rangle$ range. A high recall value indicates that the model has a low number of false negatives, meaning that it can successfully detect most of the objects in the scene.

The F1 score is the harmonic mean of precision p and recall r metrics, which could be also denoted as:

$$F1_{score} = 2\frac{p \cdot r}{p + r} \tag{4.5}$$

The F1 score value is also falling into $\langle 0.0, 1.0 \rangle$ range, where the highest possible score is achieved when both precision and recall are also at their maximum. It provides a balanced measure of the model's performance, taking both false positives and false negatives into account. The F1 score is particularly useful when there is an imbalance between the number of positive and negative instances. It also provides a single-value metric that could be used to compare overall model performance.

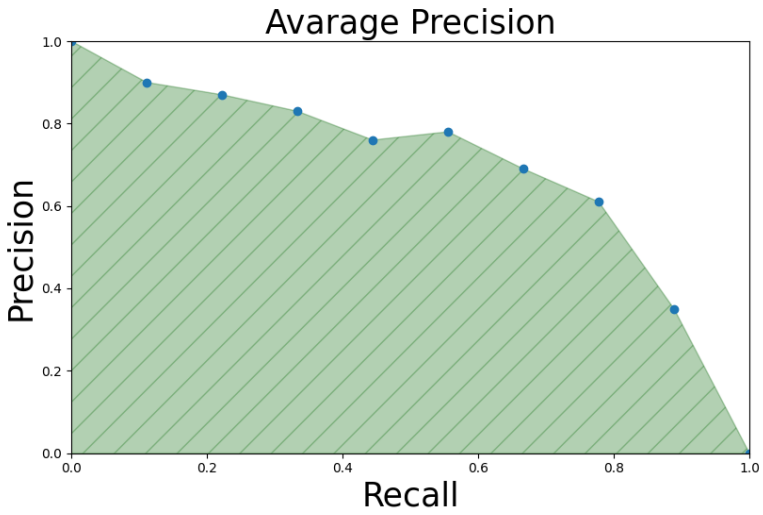

Figure 4.2: Precision vs Recall curve used for model evaluation and calculation of average precision. At different predictions' confidence levels, the value of two metrics is calculated resulting in data points marked in blue. Average precision is the mean of sampled precision values, which with dense sampling equals the area under the precision-recall curve, marked in green.

Mean Average Precision is a popular and comprehensive metric widely used in OD tasks. One of the key aspects of mAP is that it evaluates the precision-recall curve across a range of confidence thresholds, which sets the confidence level required for the model to consider a detection as a positive result. By varying the confidence threshold, a precision-recall curve can be created, as presented in Figure 4.2. Based on this curve, the average precision for each class is calculated, by sampling precision values from the curve at constant intervals and calculating its mean, effectively calculating the area under the curve, which is contained within $\langle 0.0, 1.0 \rangle$ range. Average precision provides a more comprehensive understanding of how the model performs at different confidence levels. The final mAP score is the mean of the individual class average precision values. mAP offers a more comprehensive evaluation of the model's detection performance across different object classes and varying confidence thresholds.

Mean Average Translation Error is the first supplementary 3D KPI metric. It addresses the accuracy evaluation problem related to the domain's more forgiving distance-based association conditions by quantifying the average translation error for all matched label-prediction pairs. The formula for computing the mATE value is as follows:

$$mATE = \frac{1}{n} \sum_{i}^{n} \| c^i_{label} - c^i_{pred} \| \tag{4.6}$$

where the absolute difference in translation, calculated by the Euclidean distance between the predicted and ground truth positions c_{label} and c_{pred} is computed for each true positive pair i. Then, the average of these absolute differences across all n detections is calculated, resulting in the mATE score. The score value falls within the range $\langle 0.0, th \rangle$, where th is the DIST association threshold. The lower the mATE score, the more accurate the model's predictions position is compared to the ground truth.

Mean Average Size Error is another supplementary 3D KPI metric used in evaluating the performance of models in the context of 3D OD tasks. mASE focuses on assessing the accuracy of the model's predictions in terms of the predicted size or dimensions of the detected objects. In opposition to IoU association, such parameters are not taken into account in the DIST method. The following formula is used to calculate mASE value:

$$mASE = \frac{1}{n}\sum_{i}^{n}(1 - IoU(bb^{i}_{label}, bb^{i}_{pred})) \tag{4.7}$$

where for each matched TP detection i, size error is computed by measuring the difference in dimensions between the predicted bounding box bb_{pred} and the corresponding ground truth bounding box bb_{label}, using the IoU function. This results in an error value within the $\langle 0.0, 1.0\rangle$ range. To ensure a lower error score reflects a better fit, the value is reversed by subtracting it from 1. The final mASE score is then obtained by calculating the average of these errors for all n detections. It is important to note, that for size error calculation, the respective bounding boxes are translated to the same centre position and their orientation is ignored. This ensures the metric value reflects size difference only, without any additional variables during IoU calculation. A lower mASE score indicates that the sizes predicted by the model are more accurate compared to the ground truth dimensions.

Mean Average Orientation Error is a final supplementary KPI metric among the evaluation methods for 3D OD. mAOE assesses the accuracy of the model's predictions in terms of the orientation or rotation of the detected objects. Similarly to mASE, the orientation is not validated during DIST association. To provide a more precise evaluation of this aspect of the predicted objects, the mASE is calculated based on the formula:

$$mAOE = \frac{1}{n}\sum_{i}^{n}|y^{i}_{label} - y^{i}_{pred}| \tag{4.8}$$

where the absolute difference in yaw angle orientation between the predicted y_{pred} and ground truth y_{label} rotations for each detected TP pair i. By averaging these differences, the resulting mAOE score value is acquired. The yaw angle value of both bounding boxes is in radians, thus the normalized error must fall within the range of $\langle 0.0, \pi\rangle$. By incorporating mAOE as a supplementary metric, insights are gained into how well the model can accurately estimate the orientations of the objects in 3D space.

Lastly, the NuScenes Detection Score, which was introduced in the NuScenes dataset paper (Caesar et al. 2020), represents the most comprehensive metric for assessing the overall performance of the model. The modified formula used to compute NDS in this thesis is as follows:

$$NDS = \frac{1}{6}(3 \cdot mAP + \sum_{err \in \mathbb{E}} max(1 - err, 0)) \tag{4.9}$$

where the final NDS value is a weighted mean of mAP and previously described supplementary errors metrics $\mathbb{E} = \{mATE, mASE, mAOE\}$. The errors' values are subtracted from 1 in order to match mAP value scaling, namely the higher the better. In addition, as those errors can be greater than 1, the max of the

subtraction and 0 is calculated to cap the negative impact on NDS. The NDS considers multiple aspects of OD task, offering a balanced assessment of model capabilities in a practical, single-value manner.

4.3 Explainable AI

Presented State-Of-The-Art Neural Network architectures in the camera image processing domain are widely used but are often considered as "blackbox" methods (Guidotti et al. 2018), meaning their internal data processing is not fully understood. This approach raises concerns, particularly in the context of Autonomous Driving, where a wide variety of scenes must be accurately classified, and untested situations during development could lead to potential safety issues and regulatory implications with human lives at stake. To address this issue, a new field of research known as Explainable AI (XAI) is being developed. XAI aims to provide insights into how NNs process data, enabling the decision-making methodology understandable to humans (Adadi and Berrada 2018; Samek, Wiegand, and Müller 2017). By using XAI methods, NN solutions can be more reliably evaluated in the context of safety regulations, enhancing transparency and accountability. Even for applications where evaluation is not mandatory, XAI can assist researchers in gaining a better understanding of ML models, leading to potential improvements in overall performance.

Explainable AI is gaining attention in the development of Autonomous Vehicle systems as well (Omeiza et al. 2021). In AVs, the perception system plays a crucial role in decision-making, serving as the foundation for all other components. Neural Networks are used to process sensor data from various devices, such as cameras, radars, and LiDARs, to create a model of the environment. The XAI methods related to camera image processing and OD networks have been well-developed. Especially gradient-based methods (B. Zhou et al. 2016; Selvaraju et al. 2017; Chattopadhay et al. 2018) are particularly valuable in explaining vision tasks, as they generate heatmaps to show the importance of each input pixel in the final prediction output. However, the application of XAI methods for sensor data from LiDAR and Radars, which are represented as pointclouds, poses a challenge. Extending these methods to pointcloud data is further discussed in this thesis in Chapter 9. For now, the gradient-based methods are explained in more detail for camera-only models.

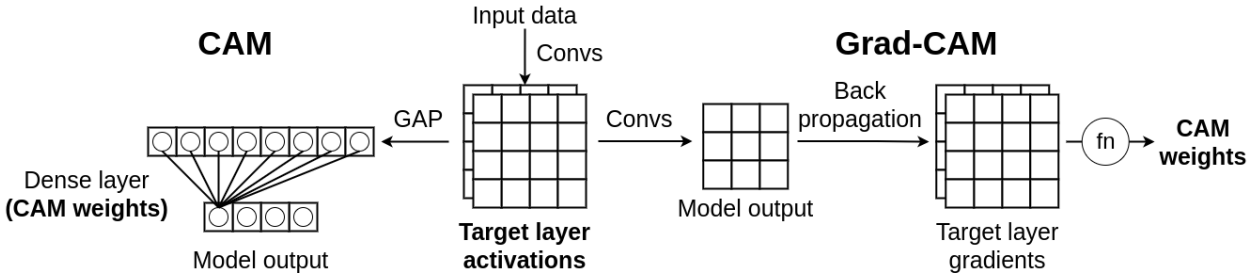

Figure 4.3: Comparison between CAM and Grad-CAM methods. Both techniques utilize the set of target layer activations. The difference is in obtaining the weights for each activation. While CAM requires a specific output structure, Grad-CAM employs a backpropagation algorithm and uses processed gradients as weights.

XAI methods have found widespread application in Computer Vision tasks, particularly due to the ease of visualizing input data processing. Visualization in the same image domain as the NN's internal representation allows for graphical representation and analysis. In (B. Zhou et al. 2016), the authors introduced the concept of Class Activation Map (CAM) to generate a visual heatmap indicating what part of the input image influences the model predictions. This process requires two components: activations from the last convolutional layer and weights for each activation. The activations are calculated during the forward pass of the NN. The CAMs are created by calculating the weighted sum of these activations. The weights are determined using a combination of Global Average Pooling (GAP) and a FC output layer. The GAP creates a vector of importance values per activation and the FC layer yields classification for each class based on this vector. Then, extracting FC layer weights provide the importance of each activation in a final CAM, as presented in Figure 4.3.

An important limitation of the original CAM method is its requirement for a final FC layer, which is not suitable for convolutional models. However, Gradient-weighted Class Activation Maps (Grad-CAM) (Selvaraju et al. 2017) addresses this drawback by being compatible with both FC and convolutional output layers. Moreover, Grad-CAM offers the flexibility to calculate CAMs for any intermediate convolutional layer within the network architecture. The fundamental principle of generating CAMs remains the same, namely to multiply activations by certain weights, corresponding to their importance. However, Grad-CAM employs a backpropagation method to compute gradients of a classification score with respect to the chosen convolutional layer's activations. The weight of each activation is then determined based on these gradient values. In the original paper, the weights are obtained by taking the mean of gradient values along the tensor dimensions corresponding to the input image width and height. A more intricate weight calculation is introduced in Grad-CAM++ (Chattopadhay et al. 2018), where the authors incorporate the pixel-wise significance of the gradient values based on spatial position in the feature map during the mean calculation.

Figure 4.4: Results of Grad-CAM application for a classification network. The original image as well as CAMs for the "dog" and "cat" classes are presented from left to right. The results show that a dog is detected based on mouth features, whereas a cat depends more on fur patterns. Source (Selvaraju et al. 2017).

Gradient-based methods for generating Class Activation Maps exhibit remarkable flexibility, making them applicable to diverse network architectures, layers, and methods of calculating activation weights based on gradient values. These adaptive characteristics hold great potential for future adaptation to an entirely new pointcloud domain and fusion solutions.

Chapter 5

Cross-Domain Spatial Matching method

Chapter highlights:

- *CDSM domain alignment principle*
- *Features extraction submodels for each sensor*
- *Three different approaches to CDSM fusion*

The fusion of 2D images and 3D pointcloud data poses significant challenges in the context of AD perception systems. On the other hand, such a fusion of the two modalities has the potential to unlock a more comprehensive understanding of the scene, overcoming the limitations of each individual sensor. In this chapter, a novel NN architecture is introduced, specifically designed for camera images and Radar or LiDAR pointcloud data to achieve a synergistic fusion of information based on a DL approach. It is important to highlight that the fusion architecture presented in this chapter forms a crucial and central part of the overall thesis. Moreover, this chapter constitutes an extension to previous research, which resulted in the provisional patent application. Through this chapter, the proposed fusion perception NN architecture is thoroughly explored, with an additional, compared to the previous work, detailed description of the mechanisms and methods behind it, as well as supplementary experiments that led to final results.

The chapter is structured as follows: Firstly an overview of the fusion method is presented with fusion level and stage classification, as well as a general explanation of the proposed solution. Then, a detailed description of the Cross-Domain Spatial Matching (CDSM) custom layer is provided, highlighting the idea behind the domain alignment mechanism. Furthermore, the single-sensor architectures are described, focusing on the specific features extraction methodologies employed for camera image and pointcloud processing. Additionally, the monocular 3D OD model with the CDSM alignment layer is presented. Finally, different fusion methods are analysed with a division into three distinct approaches. By examining each method, valuable insights are gained into their motivation, working principle, and fusion idea justification.

5.1 Architecture overview

The proposed fusion approach follows a multi-view setup concept, leveraging separate network architectures to process camera images and the 3D pointcloud data (Figure 5.1). The camera input is processed within the 2D image domain, while the pointcloud data is handled in an enhanced Bird's Eye View (BEV). Through these distinct processing paths, both NNs generate predictions in their respective domains. Additionally, to accommodate for the fusion, the feature maps from both models are passed to a common block, where the final 3D predictions are made. Such a solution, according to previously discussed level and stage classification in Chapter 2, falls under Low-Level Fusion (LLF) category because the different sensor information is fused together before predicting objects, during the sensor data processing phase. Moreover, the sensor data is already preprocessed by single modality architectures when the fusion occurs, resulting in corresponding feature maps entering the fusion block, rather than raw sensor readings. Thus, by definition, such an approach belongs to the late (feature) fusion stage class.

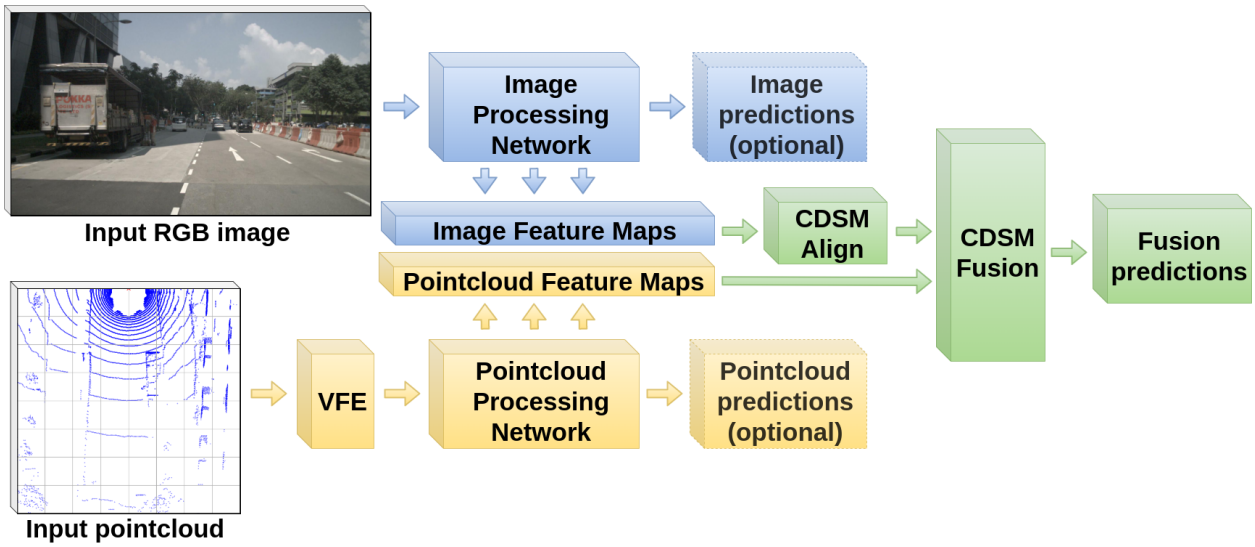

Figure 5.1: CDSM solution architecture overview for image and pointcloud fusion. The image processing and feature extraction elements are marked in blue, whereas pointcloud ones are coloured in yellow. Fusion components are highlighted in green. This colour scheme continues throughout all diagrams in this chapter.

To enable LLF, a novel component called Cross-Domain Spatial Matching fusion block is introduced. The goal of this block is to fuse feature maps from intermediate layers of the camera and radar networks, creating a unified internal representation based on both sensors' readings, that ultimately leads to object predictions in a 3D space. A critical challenge arises from the fact that these feature maps originate from completely different domains, namely the 2D camera images and the 3D BEV. Therefore, in order to fully benefit from the information provided by both sources, it becomes essential to spatially align these feature maps before the fusion process. This alignment is precisely achieved through the utilization of the CDSM block.

5.2 CDSM domain alignment

The fundamental idea behind the CDSM fusion block revolves around the spatial alignment of information extracted from 2D camera images and 3D pointcloud data. In the initial stages, the feature maps obtained from the intermediate layers of each network are inherently misaligned. The CDSM method incorporates two key elements: Domain Alignment and Fusion, with the former particularly addressing the misalignment issue.

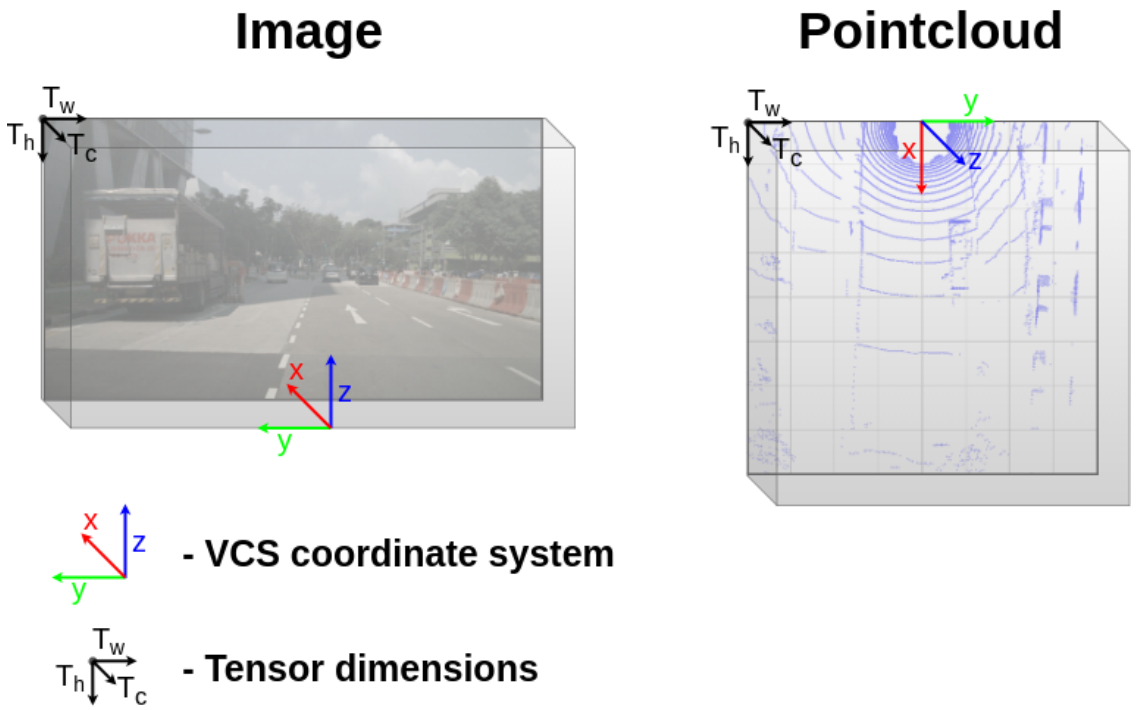

Figure 5.2: Comparison of input data tensors with respect to the VCS. Tensors dimensions, namely width, height, and channels, are denoted by the black coordinate system indicator in the top left corner. The VCS coloured indicator is arranged according to the host position in both views.

Firstly, a unified space known as the Vehicle Coordinate System (VCS) is introduced to help with the explanation of this concept. The VCS provides a standardized framework that allows for consistent representation and alignment of data from various sensors with respect to the host vehicle. It is a Cartesian coordinate system centred on the car's front axle, with the X-axis pointing forward, the Y-axis pointing to the left of the car, and the Z-axis pointing up. By considering the VCS, the camera image and the pointcloud voxel grid can be aligned in a consistent manner. Figure 5.2 illustrates the related 3D tensors for the camera image and the pointcloud voxel grid, highlighting their different orientations within the VCS. For the camera image, the first two dimensions correspond to the VCS ZY-plane, while the learned features (initially RGB values) span throughout the X-axis. On the other hand, for the pointcloud voxel grid, the first two dimensions correspond to the VCS XY-plane, and the features (initially stacked VFE outputs) span along the Z-axis.

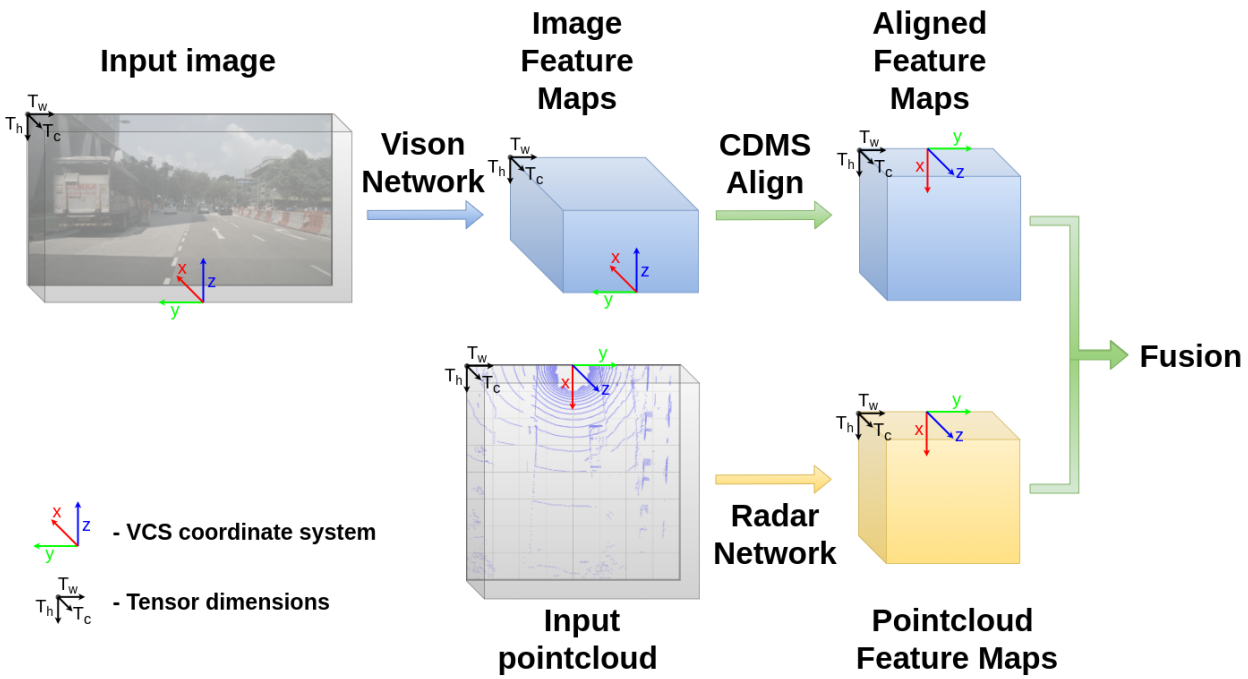

Figure 5.3: CDSM domain alignment method visualization. Both data sources are processed by the corresponding sensor-specific networks to provide extracted feature maps. Additionally, the misalignment of those features with respect to the VCS is addressed by applying rotation operation to image features. Aligned maps are passed to the fusion module.

The representation of the pointcloud voxel grid is consistent with the expected output of a fusion perception network, namely a BEV grid in the XY-plane with detected objects and the values describing their features in the tensors' channels. On the other hand, fusing information from the camera poses a challenge due to the differing perspectives represented in the tensors. To overcome this, the CDSM block incorporates a spatial alignment process based on a novel rotation layer implementation embedded into the network architecture, as shown in Figure 5.3. The alignment of the extracted features within a VCS not only enhances the integration of information across different modalities but also opens up new possibilities for fusion methods.

The alignment of tensors to match their spatial orientation in the VCS is accomplished through a custom CDSM rotation layer. Its implementation principle is illustrated in Figure 5.4. Firstly, the rotation layer extracts the indexes from the original tensor. These indexes represent the positions of the tensor values in their current camera sensor orientation. In parallel, a rotation matrix is calculated using quaternion rotations. By utilizing a chain of such quaternion rotations and the multiplication operation defined for this data structure, the final matrix can be obtained, combining all needed transitions in just one array, that captures the transformation needed to align the tensor with the VCS. The rotation matrix is then applied to the extracted tensor indexes through a simple matrix multiplication operation. This process generates new rotated indexes that correspond to the desired spatial alignment. In some cases, the rotation may result in negative indexes. To handle this, an offset is calculated to ensure the indexes retain their proper position in the tensor coordinates. The new indexes are shifted by the calculated offset, effectively adjusting for any negative values and ensuring correct spatial alignment. Then, a new empty tensor is created with dimensions that conform to the

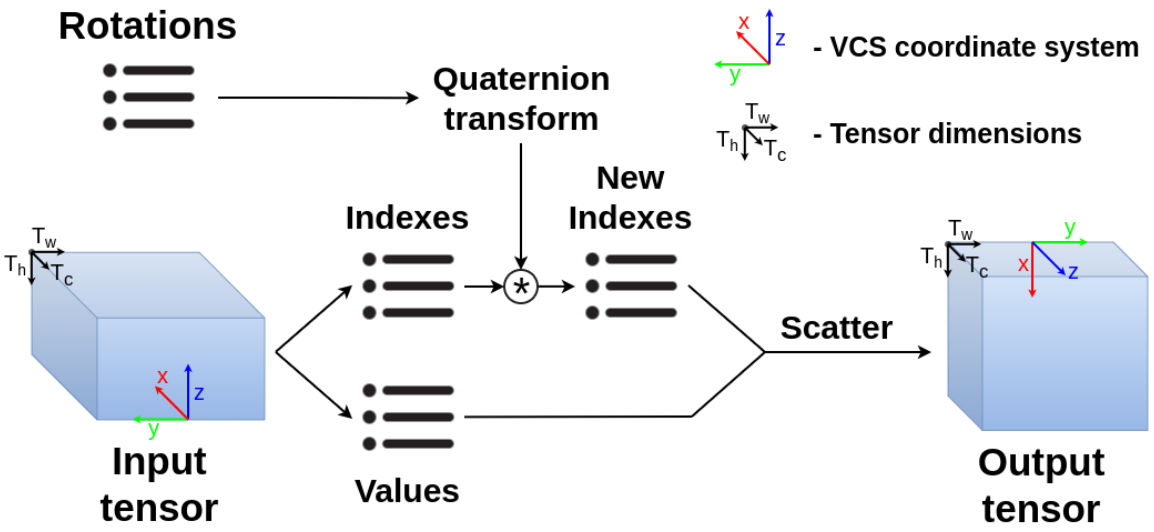

Figure 5.4: CDSM rotation layer working principle scheme. Inputs to the layer are the target tensor and a list of rotation transformations consisting of the angles and the axes around which the rotation should be applied. Each rotation is converted to a quaternion and by multiplying them together the combined quaternion transform is obtained. Meanwhile, the input tensor is converted into a list of values and their corresponding indexes. By multiplying the quaternion transform rotation matrix and indexes array, new indexes are created. Finally, with scatter operation, tensor values are placed in the new tensor under the fetched rotated indexes, which results in the output tensor that reassembles the input one after specified rotations.

rotated orientation. Finally, the values from the original tensor are scattered to their respective positions in the newly rotated tensor based on the revised indexes. The resulting output tensor of the CDSM alignment layer represents the original tensor after the rotation operation.

Specifically, for the camera feature maps tensor, a carefully designed set of rotation parameters is employed to ensure that the tensor undergoes the necessary transformations to align with the VCS. The process involves two key operations, applied in a specific order. Firstly, a rotation of 180 degrees around the tensors' first dimension (VCS Z-axis) is performed. This effectively flips the camera feature maps tensor, aligning left-to-right features' orientation with the VCS. Following it, a second rotation of 90 degrees around the tensors' second dimension (VCS Y-axis) is applied. This second step adjusts the camera feature maps tensor by changing values that span throughout channel dimension from VCS X-axis to Z-axis and aligning the width and height dimensions with VCS XY-plain, similarly to BEV representation. A successful alignment is achieved by employing the CDSM rotation layer with these specific rotation parameters. It ensures that the camera features are positioned and oriented correctly, enabling seamless fusion with the BEV pointcloud feature maps tensor. It is important to note that the specific combination and order of rotation operations selected for aligning the tensors not only ensures their orientation consistency within the VCS but also aligns the centres of the VCS to the same position. This alignment is significant as it establishes a unified reference point for both tensors. It is also worth emphasizing that achieving this level of alignment is not possible with arbitrary combinations of permutations or transpositions of the input tensor dimensions. The chosen rotation operations address the unique spatial requirements of the fusion process, ensuring that both tensors are not only properly oriented but also positioned in a coherent manner within the VCS.

5.3 Features extraction

Proposed fusion NN architecture, as stated in architecture overview sections, constitutes LLF at the feature level. Application of CDSM alignment and later on CDSM fusion methodology, requires preprocessed sensor data in the form of extracted feature maps. To that end, the integration of SOTA methods used in single-sensor perception algorithms into fusion algorithms can accommodate the solutions that have been developed in previous research. When designing the CDSM fusion NN for an AV perception system, inspiration was drawn from the 2D image and 3D pointcloud OD domains. The incorporation of these single-sensor feature extraction networks allows for the utilization of optimized network structures that have been extensively validated in their respective use cases. Moreover, such an approach enables obtaining intermediate results from single-sensor networks, which in turn can be used to compare them with each other and with the fusion outcome. Such comparison helps with establishing each sensor's contribution and the overall fusion gain. Consequently, in the following section, a single camera image and LiDAR and Radar pointcloud OD NN architectures are described, starting from the base concept of the solutions to more detailed implementation aspects, hyperparameters selection and modifications done to account for fusion model integration.

5.3.1 2D camera image network

For the purpose of camera image processing, a single-stage detector is developed, utilizing the popular EfficientDet network structure. The model consists of three key components, as illustrated in Figure 5.5: an EfficientNetV2 backbone responsible for extracting initial features from the input images, a Bi-directional Feature Pyramid Network (BiFPN) that effectively combines and integrates features at various levels of abstraction, and classification and regression heads to generate predictions for object detection. However, there are specific modifications to the network structure to better suit the intended fusion purpose. To begin with, the input image resolution is adjusted to more accurately match the target dataset's image data aspect ratio. The pixel values are also normalized to (0,1) value ranges, separately for each color channel.

The extraction of features from the backbone network is performed at three levels P3-P5, illustrated in Figure 5.5. These levels represented feature maps at different resolutions, capturing information at varying scales and levels of abstraction. However, to ensure compatibility with the subsequent BiFPN block, additional levels of P6 and P7 are artificially incorporated into the backbone feature extraction process. Such inclusion of P6 and P7 levels aims to provide even higher-level features and capture more context-rich information. As those levels constitute a higher abstraction layer, the model can encompass a broader range of spatial and semantic representations, which ultimately enhances its ability to detect and classify objects accurately. Standard P6 and P7 levels of an EfficientNet backbone are not utilised, due to their deep layers structure and the vast amount of trainable parameters. Instead, to generate them, simple downsampling convolutional layers are used, allowing the network to propagate information based on single trainable layer parameters.

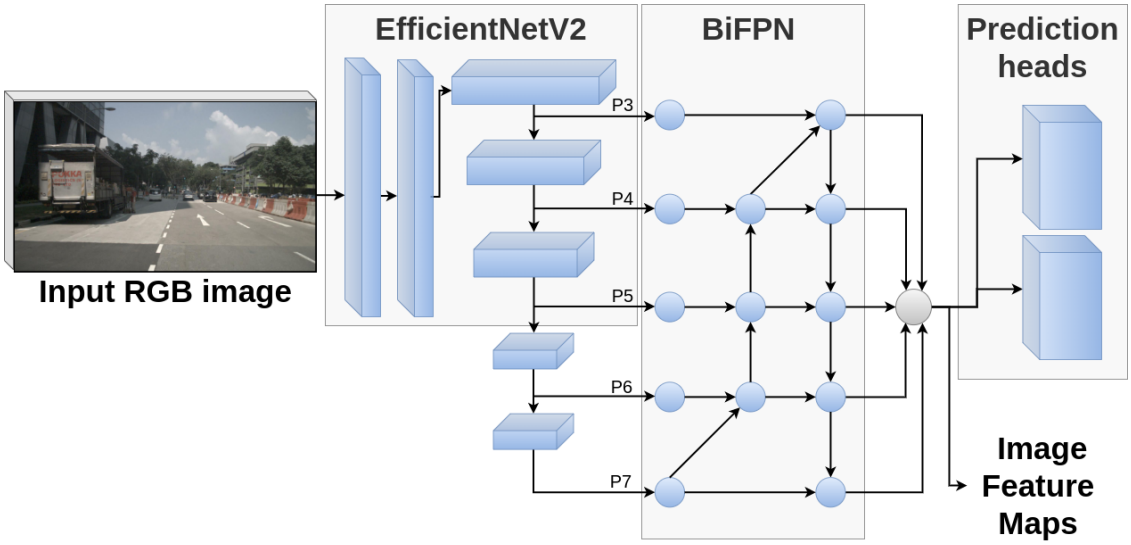

Figure 5.5: Camera image Neural Network architecture diagram. All major components, namely the EfficientNet backbone, BiFPN, and prediction heads are separated by grey background blocks. Additionally, the intermediate image feature maps input to the fusion is highlighted, yielding raw features extracted after BiFPN block.

In order to refine backbone outputs, the BiFPN block integrates multi-level features and enhances their representations. The proposed architecture incorporates BiFPN with 4 iterations, meaning that an upscale-downscale cycle (Figure 5.5 - BiFPN block) is being repeated four times. During each iteration, the refined feature tensors with 256 channels are passed through a series of operations, including horizontal and vertical feature maps connections in both directions. By leveraging the BiFPN interconnections, the refined features become more enriched with contextual information and achieve a more balanced representation across different scales. This improved feature representation enabled the subsequent classification head and regression head to make more accurate predictions. Furthermore, raw feature maps obtained during BiFPN execution are yielded as an additional model output, which could be used in CDSM fusion component.

The final OD results are obtained through the utilization of prediction heads, illustrated in Figure 5.6. There are two types of heads in the proposed architecture: the classification head and the regression head. The classification head is responsible for predicting object class labels and generating corresponding confidence scores that indicate the likelihood of the detections. The regression head focuses on estimating precise bounding box coordinates and sizes for the detected objects. By utilizing a predefined set of anchors with varying object sizes, the regression head accurately predicted the object's location and size within each grid cell. Outputs of both heads are merged together to create a set of BEV grids. Object predictions are made on five different scales corresponding to YOLO-like output grids with sizes of $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{64}$, and $\frac{1}{128}$ relative to the input size. Anchors are automatically generated based on each given grid size and combinations of three scale factors (1, 1.33, 1.66) with respect to grid size and three ratio factors (0.5, 1, 2) of anchor width to height proportion, resulting in nine anchors per grid cell. To generate the final predictions list, the NMS algorithm is applied to detections from all five scales simultaneously.

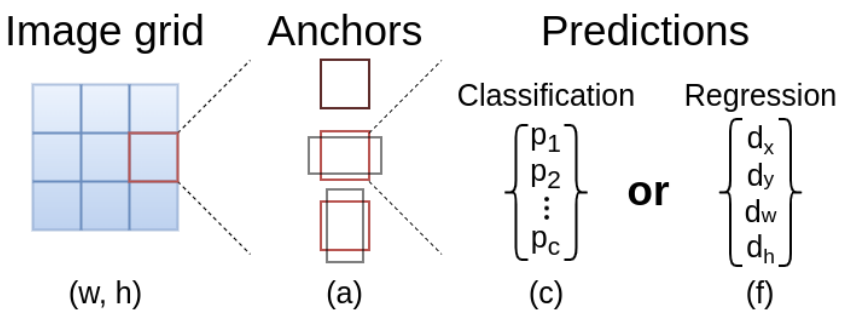

Figure 5.6: 2D prediction heads format description. Classification and regression heads are 4-dimensional tensors of shapes (w, h, a, c) and (w, h, a, f) respectively. The first two dimensions (w, h) denote image output grids of different cell sizes for each scale level. Within that grid, each cell encapsulates a defined number of a anchors. Finally, for each anchor, the network yields a prediction vector. For the classification head, this vector consists of class probabilities p_c, and its length is dependent on a number of classes c in the dataset. Regression head yields a vector of objects' features, for 2D image OD those are (d_x, d_y, d_w, d_h).

To accelerate the training process and leverage existing knowledge of image features, ImageNet pretrained weights are utilised for the EfficientNetV2 backbone. These weights have been obtained through extensive training on the ImageNet dataset, allowing the backbone to initialize with learned representations of various visual features. By utilizing pretrained weights in such a transfer learning approach, the camera processing model is able to benefit from the wealth of knowledge acquired during the lengthy training process on a large-scale dataset. It enhances the ability to extract meaningful features from input images and facilitates faster convergence during training on the different automotive datasets.

The network is further adjusted to incorporate custom normalization layers and activation functions. In order to improve the stability and generalization of the feature aggregation process, various normalization layers, such as BatchNormalization, InstanceNormalization, LayerNormalization, and GroupNormalization, could be changed in the model architecture via training configuration parameters. The utilization of different activation functions, including LeakyReLU and Mish activation functions, is also enabled. This exploration aims to evaluate diverse combinations of normalization layers and activation functions to identify the most effective configuration for the specific OD task on camera image modality.

5.3.2 3D LiDAR and Radar pointcloud network

The designed fusion model architecture incorporates 3D pointcloud data secondary input, which can originate from either LiDAR or Radar sensors. While there are notable differences in the functional characteristics of these two types of pointcloud data, such as the presence of intensity information in LiDAR data or velocity information in Radar data, the fundamental format remains largely consistent. This shared format is defined by the coordinates of the points within the pointcloud together with their additional features. Considering this commonality, a unified architecture can be employed to extract feature maps from either LiDAR or Radar pointcloud data with only a few configurable adjustments. Although the primary focus is centred upon the fusion of camera and Radar sensor data, comparison to the fusion of the camera and LiDAR sensor data might lead to getting valuable insights as well.

The pointcloud processing model takes inspiration from a voxel-wise approach, used in LiDAR State-Of-The-Art NN architectures, to extract valuable information from 3D pointcloud data and predict objects based on its internal representation. Similar to the vision feature extraction model, it is structured in a modular manner, as presented in Figure 5.7. It has a VFE input pointcloud parser, DLA-based backbone, BiFPN feature refinement block, and both classification and regression prediction heads.

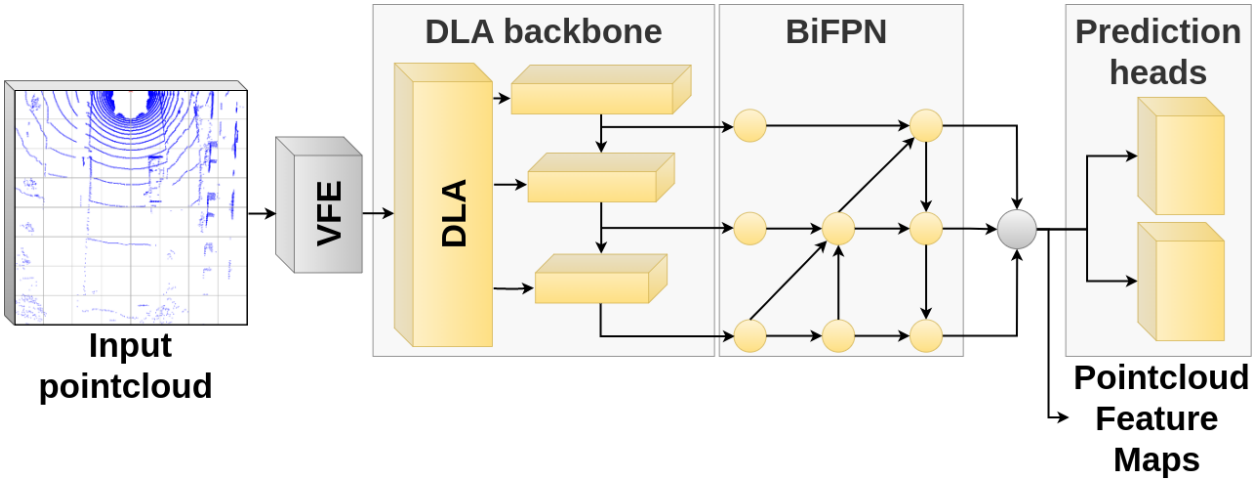

Figure 5.7: Pointcloud data processing Neural Network architecture diagram. In addition to the DLA backbone, BiFPN, and prediction heads, this model features VFE module, which preprocesses input pointcloud list into sparse voxel representation, suitable for further processing by 2D convolutions. Furthermore, similar to the image network, raw feature maps are extracted for fusion purposes.

Upon entering the network structure, the initial step in the data flow pipeline involves voxel feature extraction and processing. The 3D space is divided into a voxel grid, with each voxel representing a spatial cell that accommodates the incoming points. The grid size varies depending on the sensor type. For dense LiDAR point clouds, a grid size of 12.5×12.5cm is used, while for sparser Radar data, a grid size of 1×1m is employed. These sizes are carefully selected to ensure adequate coverage of object readings within each cell based on the distribution of sensor detections. The Voxel Feature Extractor (VFE) module, inspired by VoxelNet, calculates the features for each voxel based on the points contained within it. This calculation takes into account both the points' coordinates and sensor-specific features. To maintain a consistent number of points for VFE, a maximum threshold of 16 valid points per voxel is considered for LiDAR data, while for Radar data, it is limited to 5 valid points. In cases where a cell does not contain enough sensor data samples, it is filled with zeros. This approach ensures that the output grid always maintains a fixed tensor size, which is expected by the subsequent convolutional layers of the backbone architecture. To optimize inference time and streamline the data representation, the voxel feature tensors are stacked along the Z-axis, following the approach utilized in the Pointpillars architecture. This stacking operation transforms the original 4D representation into a 3D representation, facilitating more efficient processing of the voxel features by convolution layers.

The applicability of ImageNet pretrained weights is limited to image processing tasks since they were obtained through training solely on visual data. Consequently, when pretrained weights are unavailable, the

suitability of EfficientNet as a backbone architecture diminishes. However, this presents an opportunity to explore alternative adaptations for the model's backbone, with a focus on developing more lightweight solutions that suit specifically the desired objectives. By deviating from the reliance on ImageNet pretrained weights, novel possibilities emerge for selecting other SOTA backbone architecture. The DLA34 backbone, known for its ability to capture rich contextual information and complex spatial patterns, was chosen as the replacement. However, the backbone is adapted specifically for pointcloud processing with modifications, which ensure that it is optimized for the unique characteristics of sparse pointcloud data embedded into a relatively large voxel grid. Initial convolutional blocks are replaced with strided convolutions to downsample large tensor sizes and deep DLA34 tree-like structure is reduced significantly from 7 to 4 levels. The motivation behind this approach is to optimize hardware resource utilization and improve inference time, as pointcloud data is inherently sparser and does not require as many processing layers as image pixels. The DLA-like backbone still effectively integrates features from different scales and levels of abstraction within the voxelized pointcloud data. Despite being much smaller in size compared to EfficientNet, it maintains its capability to perform sufficient feature capturing, which is a crucial task of a backbone module.

The BiFPN module, follows the same principles as its counterpart in the vision model but is adapted for the pointcloud processing model. Given the sparse nature of the data, the number of BiFPN repeats is reduced to 3, while maintaining the convolutional filters at 256 channels. The output of the backbone architecture includes three grid scales, which subsequently pass through the BiFPN module, resulting in 3 refined output feature maps. Both the backbone and BiFPN utilize configurable normalization layers and activation functions described in 2D image architecture, enabling them to be subjected to experimental testing during the training. This approach ensures consistency and facilitates comparative analysis of different normalization techniques and activations in the context of the pointcloud processing model as well.

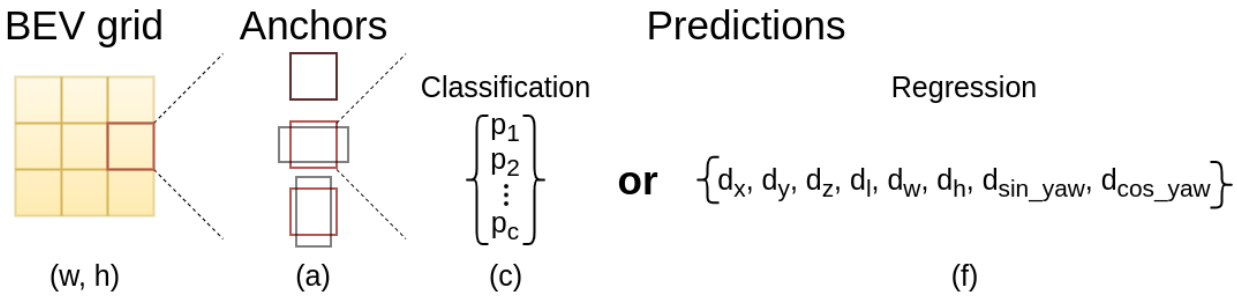

Figure 5.8: 3D prediction heads output format is similar to one presented in Figure 5.6. Both heads predict 4-dimensional tensors of shapes (w, h, a, c) and (w, h, a, f). The first two dimensions (w, h) are also grid cell coordinates, but in this case, they correspond to 3D locations in BEV grid, rather than image pixels. The third dimension denotes a anchors per each cell. Lastly, the classification output is identical to the 2D solution, and the regression output is extended with additional features needed for the 3D OD task.

The pointcloud processing network produces its output through regression and classification heads as well, but they differ from 2D vision ones in terms of the prediction domain (Figure 5.8). Rather than using a 2D image plain grids, the network generates a set of three BEV grids with sizes 80×80m, 40×40m, and 20×20m. These grids cover an 80×80m RoI in VCS, with cell sizes of 1m, 2m, and 4m respectively. The

decision to have only three heads is due to the nature of 3D BEV objects, where their sizes remain constant regardless of their position or the sensor's perspective. Consequently, the three grids, covering the range of three distinct grid sizes, effectively capture the necessary information for object size estimation based on the available anchors. Furthermore, the encoding of additional dimensions is employed to represent the centre and height of the bounding box, as well as the yaw rotation angle in 3D for each prediction. By combining these predictions using a NMS algorithm, the network produces the final OD results.

5.3.3 3D monocular camera network with CDSM

Although the presented solutions accommodate feature extraction from both sensors for the fusion, comparing the results of 2D and 3D networks is challenging due to differences in their domains and the complexity of the task. OD is inherently easier on 2D camera images as it does not involve depth estimation or predicted object rotations in a 3D coordinate system. To assess the impact of the camera image on 3D detection and determine the potential fusion gain, an additional network structure can be employed. In this regard, the previously described CDSM alignment block can be utilized alongside the presented 2D camera model. This alignment block serves to bridge the gap between the 2D and 3D domains, enabling the acquisition of results in 3D from image input only, which can be easily compared to the 3D pointcloud data results and evaluated for fusion performance.

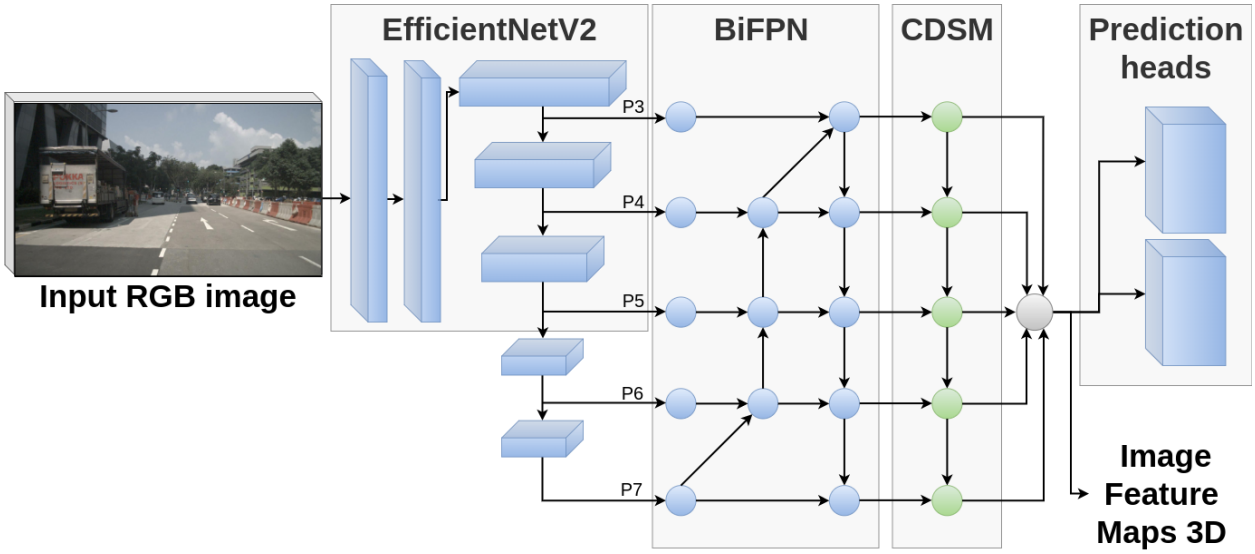

Figure 5.9: 3D monocular camera Neural Network architecture, which is an extension of previously described 2D camera model. All main components remain exactly the same, however, in order to perform OD predictions in a 3D domain, a dedicated CDSM domain alignment block is present after the BiFPN. Consequently, the prediction heads utilize a 3D prediction format rather than one for 2D bounding boxes.

As presented in Figure 5.9, the structure of the 3D monocular OD model for the camera image closely resembles that of the 2D image architecture. The backbone and BiFPN components of the network remain unchanged, with no modifications applied to them. However, to facilitate the alignment between the 2D camera domain and the 3D VCS domain, a CDSM alignment block is introduced. This block operates on

the 2D image feature maps at each level individually, applying discussed rotations, and subsequently transforming them into the 3D VCS domain. The objective of this alignment is to ensure the spatial compatibility needed for the 3D OD task. The prediction heads in the network are designed to generate outputs in the form of objects in BEV grids, leveraging the same approach that is used in LiDAR and Radar networks.

In this model, there are no explicit projections or depth estimation mappings. Instead, the CDSM layer allows the network to learn the spatial mapping of 2D features onto a BEV grid. This mapping process is solely dependent on trainable convolutional weights, which are adjusted through backpropagation to align with the provided 3D targets. During the training, the network implicitly learns the optics of the camera, which enables it to perform required mappings. Within the CDSM block diagram, additional vertical connections are established between different rotated 3D feature maps. These connections represent internal aggregations of features across different scales. Such aggregation occurs prior to the prediction processing in the heads, enriching the grid data with information from other rotated BiFPN levels. The goal of such an operation is to refine features in the new domain, providing an additional layer of trainable convolutions along different data dimensions after the alignment process. The specific methods employed to perform these aggregation operations are further described in the subsequent fusion method section.

It is important to highlight that the 3D monocular camera network is used solely to obtain camera-only 3D detections for comparison purposes and KPI evaluation. Although it shares a similar concept of a 2D to a 3D CDSM features alignment, it is not directly a part of the fusion architecture. Regardless, this solution can be effectively applied to an AV system that solely relies on camera sensors. In such a context, the 3D predictions obtained through this approach hold significant value for perception tasks over simple 2D image plain detections.

5.4 CDSM fusion

Upon reviewing an architecture overview diagram, it becomes apparent that all the fundamental components necessary for sensor data fusion are now properly established. The described single-sensor submodels process both inputs, resulting in extracted feature maps that internally represent the data from each sensor. Additionally, a novel method called CDSM domain alignment is utilized on the 2D image features to transfer the learned representation into a unified VCS in BEV. This method opens up new possibilities for sensor fusion between camera images and LiDAR or Radar pointcloud data.

Among many ideas, a few most promising approaches have been selected, implemented and tested, to determine the best technique in terms of KPIs performance. Those methods represent different strategies regarding how to merge information from both sensors within a NN architecture structure. In the upcoming section, three distinct fusion solutions are introduced, starting with a straightforward approach of concatenating feature maps from the single-sensor submodels on a one-to-one basis. Subsequently, more sophisticated techniques are discussed involving the feature-wise aggregation of image features and range-based positioning in a BEV grid.

5.4.1 One-to-one fusion

Due to the CDSM alignment method, the fusion of camera and pointcloud features has become a relatively simple task, as they are now converted into a shared coordinate system. In the BEV representation, the camera feature maps spatially correspond to those obtained during pointcloud data processing, and the origins of the VCS overlap in both representations. The concept of one-to-one fusion (Figure 5.10) assumes an equal number of feature maps from both modalities, allowing them to be merged at each level. To that end, only three top-level 2D image feature maps are used to match the number of pointcloud ones. This decision is based on the fact that they encapsulate the most intricate and precise information derived from the image. The actual fusion process involves concatenating pairs of feature map tensors along the channel dimension. During the architecture design, compatible grid sizes were ensured so that the tensors from both sources can be concatenated without any modifications to their width or height. Both feature map tensors from their respective subnetwork BiFPN blocks consist of 256 channels. By concatenating them, a new feature map with 512 channels is obtained, incorporating information from both sensors for each grid cell.

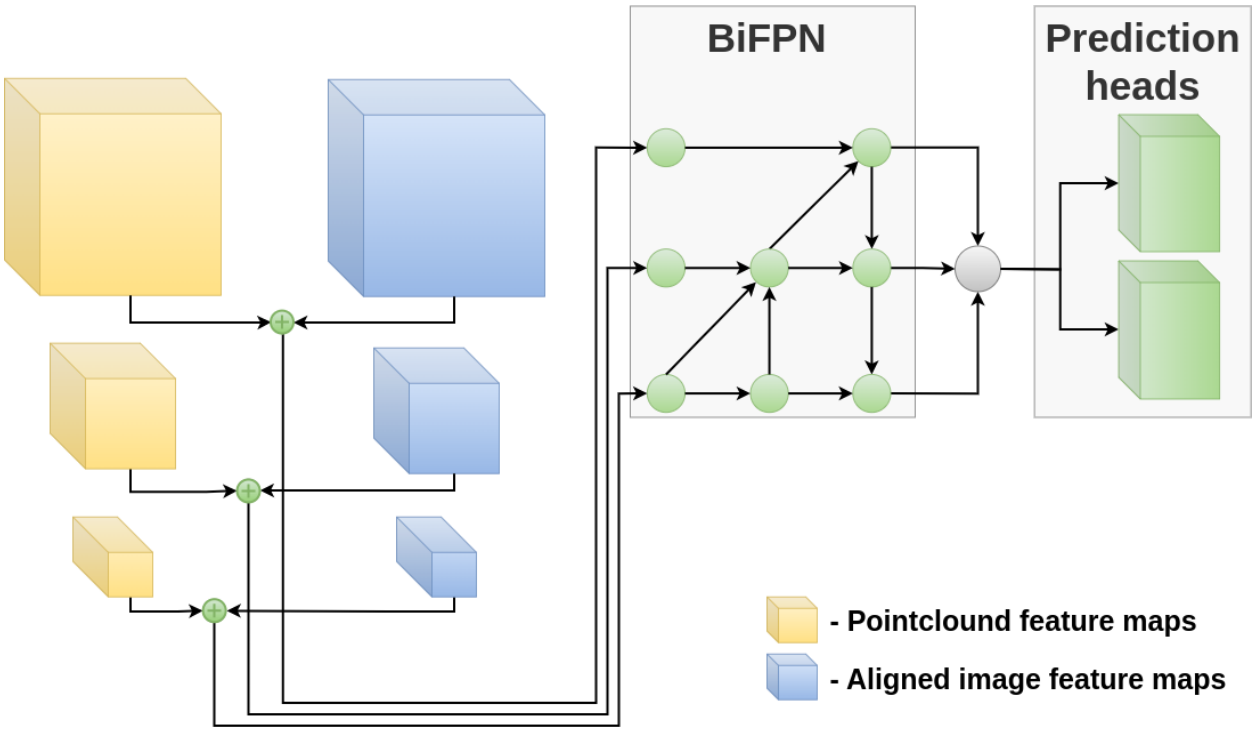

Figure 5.10: One-to-one CDSM fusion approach structure. In this baseline solution, feature maps from the pointcloud network are fused, via concatenation, with three top-level image feature maps, after domain alignment application to the latter ones. Further BiFPN refinement enhances fused features and 3D prediction heads yield OD results.

Afterwards, another BiFPN block is applied to the concatenated feature maps at different levels to combine the information from both sensors into a unified 3D internal representation. This step also utilizes cross-level connections to enrich the new representation, making use of both sensors' features at the same time. However, this BiFPN block consists of only two repeats to limit computational overhead in the overall

architecture. Additionally, the feature map channels are being reduced from 512 to 256 during the BiFPN feed-forward inference. Each sensor features should already be well-defined in single-sensor modules, thus the main task of this BiFPN is to employ cross-modality features fusion. The resulting representation, in the form of fusion feature maps, is then utilized in prediction heads similar to those found in previous 3D networks to produce the final object predictions for the entire system.

Such a one-to-one concatenation approach represents the most simple solution that could be conducted to achieve both sensors' fusion on given aligned feature maps. The purpose of this implementation is to serve as a baseline method, enabling the evaluation of more complex approaches, whether they introduce any enhancements compared to it.

5.4.2 Feature-wise aggregation fusion

The subsequent two fusion methods are designed on top of the straightforward one-to-one architecture approach. They involve the process of combining vision and pointcloud features using the same concatenation layers, followed by the application of BiFPN and 3D prediction heads of the exact structure. However, these methods introduce an additional step of merging aligned image feature maps before concatenating them with the pointcloud features, as illustrated in Figure 5.11. The entire process is divided into two distinct actions: features aggregation and features refinement.

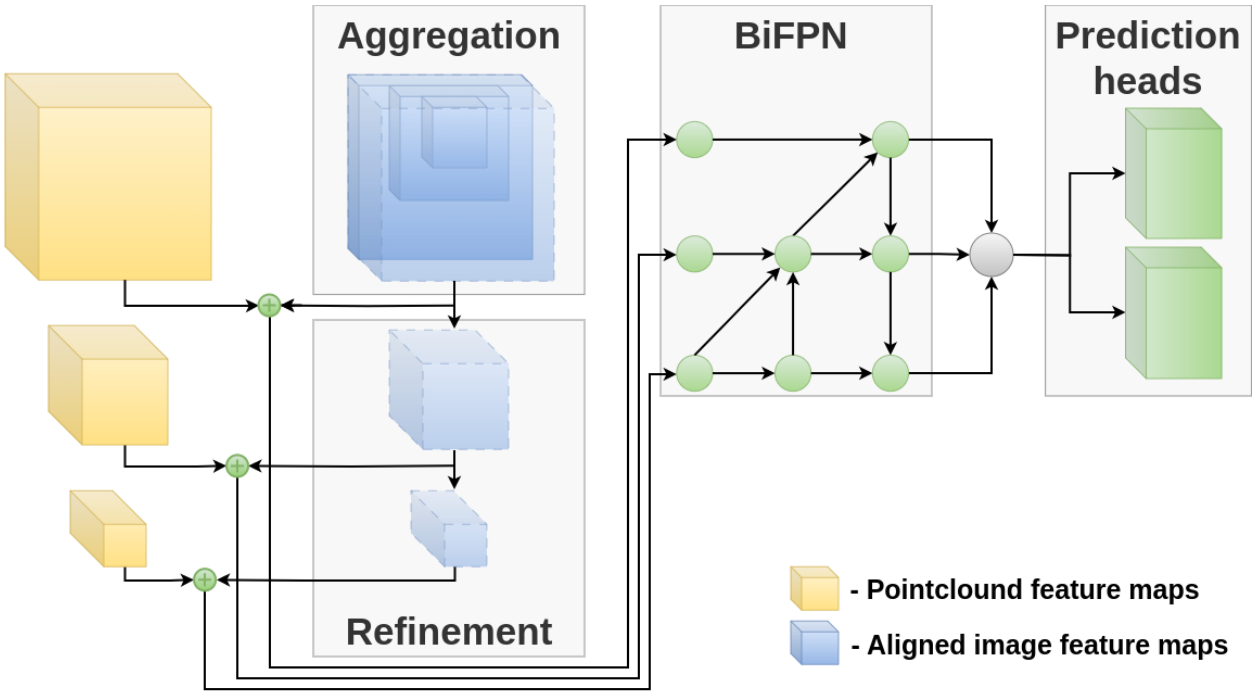

Figure 5.11: Feature-wise CDSM fusion approach structure shows another, more sophisticated take on merging feature maps obtained from both sensors. Camera features from all levels are concatenated in a single BEV grid, combining the information from each scale. Further down-scaling with the use of strided convolutional layers generates refined grids, which are fused with corresponding pointcloud feature maps.

The enhanced fusion concept is motivated by the fact that, prior to the CDSM alignment block, the image feature maps are processed in a different coordinate system. The incorporation of contextual information naturally occurs during the convolution operation across tensor width and height dimensions. In the 2D image feature extraction model, those dimensions correspond to the YZ-plain of the VCS. A rotation layer aligns the 2D features with the VCS and positions them correctly in a corresponding cell of the BEV grid. However, in this new perspective, the feature values within each cell do not contain information from neighbouring cells, due to the dimensions shifts. Specifically for this system it relates to the change of tensor height and channel dimensions that relates to the VCS Z-axis and X-axis.

To address that and build more spatially aware vision feature maps, feature aggregation and refinement stages are introduced. In the aggregation component, all camera feature maps from different scale levels are concatenated on a unified BEV grid. This concatenation occurs along the Z-axis of the VCS, combining all levels of features in each respective cell. To account for different map sizes, zero padding is applied to extend smaller maps to the highest resolution. The outcome of the aggregation operation is a consolidated BEV feature map that contains enriched features information obtained from all camera image BiFPN levels.

Following the aggregation step, a feature refinement is conducted for two primary reasons. First and foremost, the refinement comprises a set of convolutional layers with a 3×3 filter size to establish spatial correlations among the features. This step enables the model to learn relations between nearby cells in rotated feature maps, as the convolutions are now applied to tensor width and height in the BEV, which corresponds to the VCS XY-plain. Additionally, it generates higher-level features that encompass larger areas within the BEV perspective. Similarly to the concept of the backbone, the features are processed from detailed to more general ones, creating smaller grid representations of the same BEV area, based on the initial aggregated map. That also leads to the second reason for feature refinement, which is that it produces three distinct BEV grid feature maps in a 3D domain. These refined feature maps from the camera sensor can be directly fused with the three pointcloud feature maps, facilitating the integration of information from both sensors in the previously discussed later stages of the fusion architecture.

5.4.3 Range-based aggregation fusion

The range-based concept of CDSM fusion also incorporates aggregation and refinement steps, driven by similar objectives as described in the previous section. The aggregation and refinement steps are employed to enhance the initial 2D image feature maps that play a significant role in the fusion process. Moreover, through a detailed analysis of each individual camera feature map, an observation was made, leading to the development of the range-based aggregation approach.

Range-based aggregation idea originates from the explored Explainable AI multi-scale Grad-CAM method, described in detail in Chapter 9. Generated Grad-CAM maps, presented in Figure 9.2 on page 116, show the range-specific nature of features extracted by the image network. After further investigation, the features-to-range connection could be attributed to the nature of 2D image labels used during the

single-sensor model training, in particular the anchors they are based on. As previously described, each level prediction head uses a set of auto-generated anchors, based on the grid size as well as scale and ratio parameters. When a grid size is small and fine-grained, the anchors corresponding to it are also little, which in the camera perspective view, match objects that are far away from the sensor. On the contrary, when the objects are close, the bounding boxes match the large anchors best, which in turn are placed on a low-resolution grid with a larger cell size. This observation underlines the significance of considering the range or distance information within each vision network's BiFPN level, as they may contain valuable information that varies based on the object's proximity to the camera sensor. To that end, the new aggregation stage takes a completely different approach, as illustrated in Figure 5.12. The range-based strategy is proposed, allowing for the integration of distance-specific information during the fusion process.

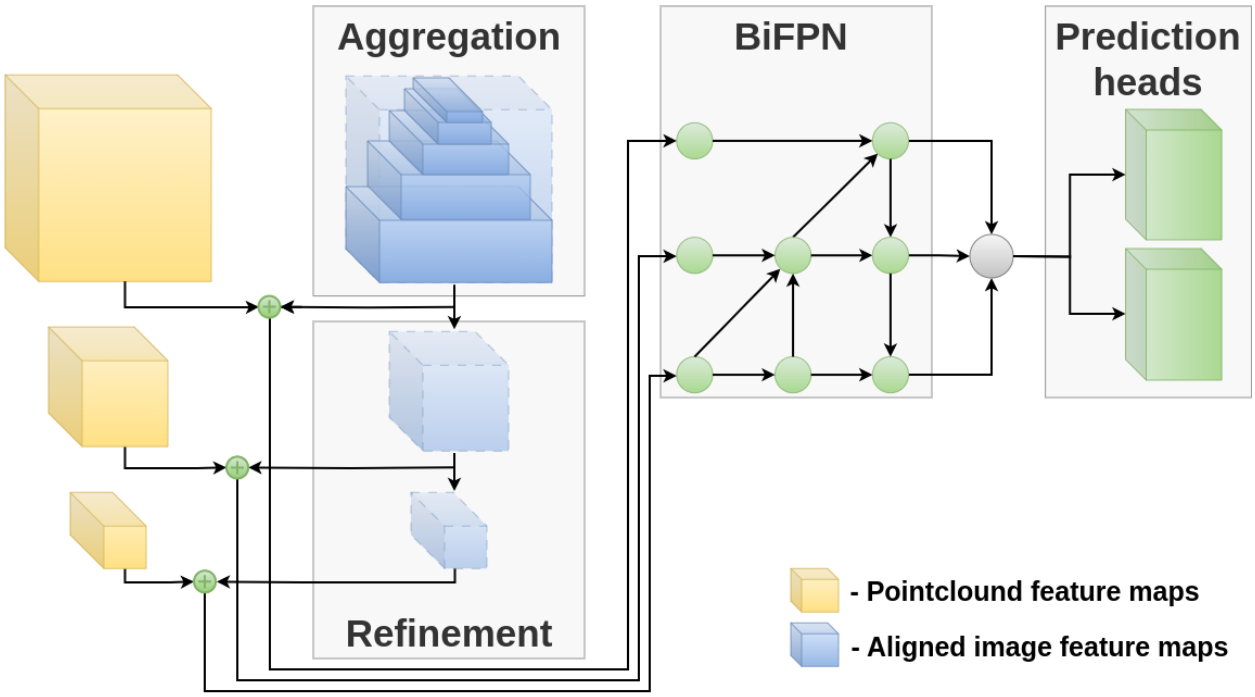

Figure 5.12: Architecture of the final approach to CDSM fusion utilizing range-based aggregation method. Similarly to feature-wise aggregation, all intermediate image feature maps are merged on a single BEV grid. However, instead of simple concatenation, a range-based technique is applied. According to selected ranges, each feature map spans a certain distance in front of the host. Additionally, the width of the maps resembles the camera sensor FoV.

This knowledge is utilized by incorporating 2D image feature maps into the aggregation process through the concatenation of tensors along the X-axis of the VCS. This enables the fusion model to account for different distance ranges on the BEV grid, thus accommodating the range-specific characteristics of the image data features. To ensure compatibility and consistency, a common BEV grid size of 80×80m is established at the top level, matching the largest pointcloud feature extraction map size. The range-based aggregation method considers five camera sensor feature maps that are placed on the common grid according to the insights from Grad-CAM. These feature maps are adjusted to match specific distance ranges, namely 2m, 4m, 10m, 24m, and 40m. Each range corresponds to a different segment in front of the host vehicle within

the BEV representation. The ranges covered are 0-2m, 2-6m, 6-16m, 16-40m, and 40-80m, respectively. By distributing the camera feature maps based on these distance ranges, the fusion method ensures that the information captured by the image sensors aligns with the corresponding distances in the BEV space, facilitating a comprehensive fusion process.

Furthermore, to maintain consistency with the camera's Field-of-View in the fusion process, additional measures are taken to situate the feature maps with the corresponding regions in the BEV grid. It includes specific features placement along the VCS Y-axis, to account for camera blind spots that occur in close proximity to the sensor's placement within the BEV grid. Such limitations in visibility arise from the camera sensor's perspective during image data acquisition. The aggregation process reassembles the same coverage by adjusting the feature maps sizes in the Y-axis dimension. This is achieved by padding the feature maps with zeros outside the FoV, ensuring that only the areas visible to the camera are considered during fusion. By zero-filling the feature maps beyond the FoV, the fusion model restricts the information to the regions that can be captured by the camera sensor. This alignment enables the fusion model to effectively incorporate the relevant information from the camera's perspective while excluding data from areas outside the FoV that may introduce noise or irrelevant details. This approach ensures that the fusion model remains consistent with the camera's perception and avoids incorporating misleading or erroneous information from regions outside the FoV onto the fused grids.

After the introduction of the innovative aggregation layer, the common feature map, which consists of the concatenated vision features, undergoes further processing through convolutional refinement layers, similar to the prior method. This refinement process generates smaller grids that contain more refined and detailed information. Once the refinement stage is complete, the refined feature grids are ready to be fused with the pointcloud feature maps in a channel-wise manner. The fused feature maps then undergo the same processing steps as in the feature-wise fusion method. They are passed through the BiFPN and the prediction heads, which utilize these refined features to generate 3D prediction results.

Chapter summary:

- Novel Cross-Domain Spatial Matching sensor data fusion architecture was introduced and classified as late Low-Level Fusion. The clear separation between single-sensor feature extraction and fusion modules was presented.

- CDSM domain alignment method was discussed in terms of the idea behind sensors' coordinate systems unification as well as the implementation details.

- Both the camera and pointcloud feature extraction submodels were described in detail. Aside from backbone and BiFPN, an in-depth explanation of prediction heads in 2D and 3D domains was conducted.

- The use of CDSM alignment technique was proposed for vision-only monocular model architecture for OD in a 3D domain, based solely on the camera sensor. This novel approach not only provides a lightweight and simplistic alternative to current SOTA solutions in that field but also enables camera sensor impact evaluation in further fusion experiments.
- Lastly, three distinct approaches to CDSM fusion were proposed. Each solution was motivated and the implementation details of methods were provided.

Chapter 6

Data and training

Chapter highlights:

- *KITTI and NuScenes datasets description*
- *Common training process details*

In the domain of ML, the model architectures discussed in the previous chapter represent only one aspect of the whole solution. The outcome results are also heavily influenced by two other crucial components: the data used and the configuration together with hyperparameters of the training process. Before discussing experiments and KPIs results in the upcoming chapters, it is essential to closely examine these two elements. The model is trained in a supervised manner, aiming to closely mimic the prediction distribution of the provided dataset labels. Therefore, properly preprocessed data plays a vital role in the entire system. A detailed description of this preprocessing step is needed for any NN-based solution. In addition to the data, the configuration of the model training is another critical factor. An important element is properly crafted loss function. Additionally, the chosen values of training hyperparameters influence how effectively the model learns the target labels' distribution and generalizes on the overall OD task. Taking into account a wide variety of proposed single-sensor and fusion models, there are subtle changes, when it comes to optimization of each and every architecture. However, there are some high-level general settings that apply to all training processes. This chapter primarily focuses on these common aspects, describing the mechanisms and solutions used across all experiments. Specific differences and exceptions for individual models are discussed while explaining each particular example.

This chapter begins with a detailed description of the datasets used for training and validation. Two open-source datasets are utilized, and an overview of their characteristics is provided. Subsequently, the training process is presented, with a comprehensive explanation of its various components. This includes the target generation methodology based on provided labels, the definition of loss functions for the 2D and 3D domains for both classification and regression heads, the settings of the optimizer used during parameters optimization, scheduling of the Learning Rate (LR) as the training process progresses, and thorough hyperparameters tuning done to maximize the performance metrics of trained models.

6.1 Datasets

Datasets play a crucial role in training models in a supervised manner, as they provide both input data and target data. The input data comes from the sensors used in the test vehicle during the data acquisition process. Obtaining target data involves a manual human annotation process. Several open-source datasets have been developed to facilitate research and development in the field of CV and sensor data fusion for AV perception purposes. There are key beneficial aspects of using such datasets.

When it comes to data synchronization, those datasets ensure that the data from different sensors, such as cameras, LiDARs, and Radars, are carefully timestamped and aligned. This temporal synchronization is crucial for valid sensor fusion and enables researchers to fuse corresponding sensor data samples together and explore multi-modal approaches effectively. Furthermore, open-source datasets also pay attention to calibration. Both extrinsic and intrinsic calibration is provided for the whole dataset sensor suite. This information is essential for accurate fusion of sensor data. By providing well-calibrated sensor parameters, fusion methods could use domain mappings and precise data transformations.

Moreover, the manual human annotation process involved in obtaining target data is a time-consuming and labour-intensive task. It requires annotators to carefully examine each frame or data point, outline object boundaries, assign class labels, and provide additional annotations. The process also involves multiple rounds of review and quality control to ensure accuracy and consistency, adhering to specific guidelines and standards. This attention to detail reflects in reliable and high-quality labelled data. With the use of open-source datasets, the models can be trained on such quality data, without the burden of manually annotating large amounts of data.

Among the various open-source datasets available, KITTI (Geiger, Lenz, and Urtasun 2012) and NuScenes (Caesar et al. 2020) datasets were specifically chosen for this study due to their availability and unique features that facilitate comprehensive evaluations and the exploration of sensor fusion techniques involving various sensors. Moreover, due to high popularity in automotive research community, these datasets provide a unified platform to compare the results of different solutions.

6.1.1 KITTI dataset

The KITTI data collection holds a significant position as one of the pioneering datasets made accessible to the public as an open-source automotive dataset. It has been widely used for evaluating and benchmarking various AV perception methods across tasks such as OD, tracking, semantic segmentation and depth completion. Being one of the first comprehensive sensor suites collection, KITTI established its position as a common framework for SOTA methods comparison.

The KITTI dataset utilizes a Volkswagen Passat B6 as the recording platform vehicle, which has been modified to perform accurate data acquisition. For data recording, a system consisting of an eight-core i7 computer with a RAID system is employed. This computer runs on Ubuntu Linux and is provided with a real-time database. Such a setup provides the necessary computational power and storage capabilities to

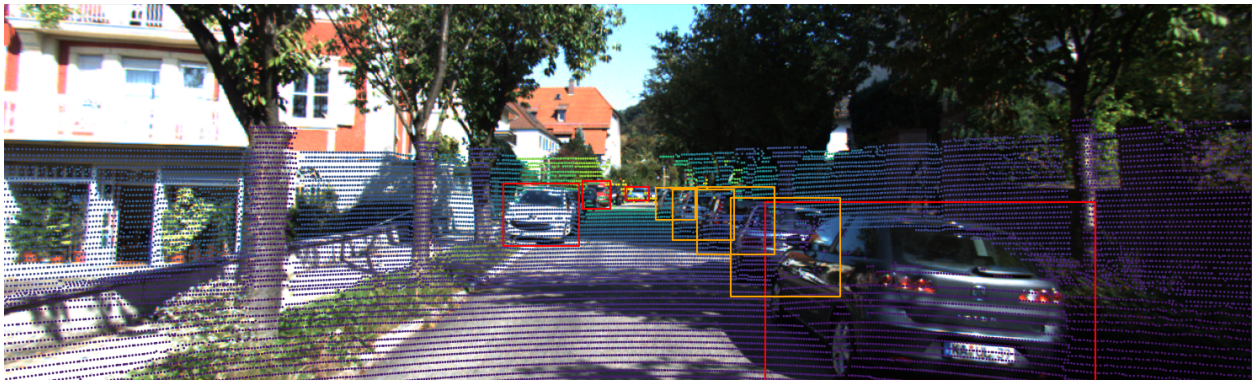

Figure 6.1: An example of a KITTI camera view rectified image sample with projected LiDAR points (colour coded according to intensity readings) and labels. Additional overlays are shown to present data, proposed neural network models are fed with raw RGB images. Labels with visibility over 50% are marked in red, otherwise in orange.

efficiently capture and process the data from the recording process. The sensors suite includes an Inertial Measurement Unit (IMU) - OXTS RT 3003, a Velodyne HDL-64E LiDAR, two grayscale cameras and two colour Point Grey Flea v2 cameras. The RGB cameras are positioned approximately parallel to the ground plane and are triggered at a rate of 10 frames per second by the complete scan of a LiDAR sensor. The shutter time is dynamically adjusted, with a maximum shutter time of 2 ms. The resulting images are cropped to a size of 1382×512 pixels. Then, the images undergo a rectification process, which eliminates the distortion from camera lenses and yields a slightly smaller output resolution size of roughly 1242×375 pixels. The rectified camera image from the KITTI dataset is presented in Figure 6.1. In order to reduce the computing resources needed in the proposed NN architecture, a letterbox resize with a constant aspect ratio kept from the original image is applied to further decrease the resolution to 640×256 pixels. The same aspect ratio helps with recognizing and learning various objects' features in the image and provides a robust possibility of reusing the trained model with input images of different resolutions prior to it. Additional preprocessing step normalizes RGB pixels values from 0 to 255 integer range to 0.0 to 1.0 float one. This operation aims to improve information propagation in initial convolutional layers.

KITTI sensor suite utilizes a LiDAR sensor, with no additional Radar devices as a pointcloud data source. The mechanical rotating Velodyne HDL-64E operates by spinning a laser scanner at a rate of 10 frames per second across 360-degrees FoV. The vertical resolution of the LiDAR is 64 beams that capture 64 distinct measurements of distance and reflectivity at different vertical positions within its scanning range of up to 120m. During each rotation cycle, the scanner captures approximately 100,000 points, which combined generate a sensor pointcloud data sample. The example of such a pointcloud sample is illustrated in Figure 6.2. Velodyne LiDAR sensor used for the data collection process was among high-end devices at the time of KITTI data collection. As a result, the pointcloud sample is very dense and rich in detections, especially for close-range objects, near the host vehicle. Such data characteristic does impact perception algorithms significantly. A dense representation in the LiDAR sensor data improves learned features quality and yields better OD task predictions due to the detailed information it contains.

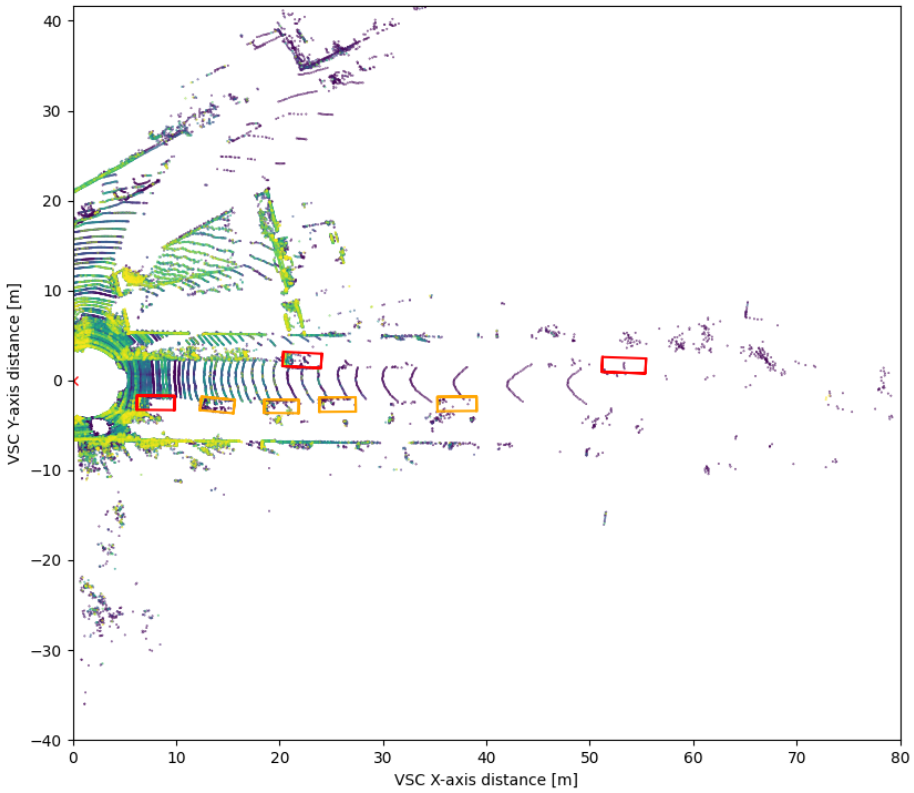

Figure 6.2: An example of a KITTI Bird's Eye View with projected LiDAR pointcloud (colour coded according to intensity readings) and labels. The same scene as in Figure 6.1 is presented from a different perspective. Label visibility over 50% is colour-coded in red, otherwise in orange.

The KITTI dataset provides labelled objects in four main classes: car, truck, bicycle, and pedestrian. Each label contains both 2D bounding box coordinates in image space (pixels) and 3D bounding box coordinates in the VCS. Additionally, the dataset includes information about label occlusion in the camera view, providing insights into the visibility of objects. Furthermore, the distribution parentage of heavily occluded objects in the KITTI dataset is rather low, compared to fully visible and only partially occluded ones. To that end, the application of filtering objects with camera visibility over 10% does impact the overall dataset label number only slightly but provides a better target for the training process. It's also important to note that the labelling process in KITTI considers only objects that are present in both the camera and LiDAR sensor data. For the proposed models in this study, a FoV covering 80×80m grid in front of the host vehicle is selected. However, as discussed in the CDSM aggregation technique, camera FoV in a BEV grid is limited by angular sensor resolution. To address this, for vision models the labels are filtered by their position, keeping only those visible in the image. This translates roughly to ± 40 degrees horizontal FoV in front of the host vehicle. Additionally, the LiDAR sensor covers a much broader 360-degree FoV. To ensure compatibility with the provided labels, front-only OD, and subsequent fusion, a detection filtration process is employed. This process discards detections outside of the selected FoV, reducing the size of the pointcloud and making it more suitable for planned experiments.

6.1.2 NuScenes dataset

The NuScenes, another widely used automotive dataset, was introduced and released in 2019 by Motional (formerly known as nuTonomy). The dataset features recorded scenes obtained from actual test drives conducted in various environments and cities. Specifically, it contains 1000 driving scenes captured in the cities of Boston and Singapore, both renowned for their congested traffic and demanding driving conditions. Each scene, carefully handpicked, spans a duration of 20 seconds and offers a diverse and captivating collection of driving manoeuvres, traffic scenarios, and unpredictable behaviours.

The NuScenes dataset, version 1.0, was utilized in this study as well. Inspired by the influential KITTI dataset, NuScenes stands out as the first extensive dataset to encompass data from a complete sensor suite found in AVs. This comprehensive dataset consists of recordings from six RGB cameras covering the 360-degree area around the host vehicle, one rotating LiDAR sensor, five Radar sensors, as well as Global Positioning System (GPS) and IMU sensors. All sensor data samples are synchronized and calibrated. Access to the data is provided via the NuScenes development kit library supported by Motional. Notably, NuScenes provides seven times more object annotations compared to KITTI. While many previous datasets primarily focus on camera-based object detection, NuScenes aims to encompass the full range of sensors available, which aligns perfectly with the fusion approach presented in this work.

The front CMOS RGB camera sensor utilized in the NuScenes dataset captures data with a frequency of 12Hz. The resulting rectified images have a resolution of 1600×900 pixels and are subject to auto exposure and JPEG compression. Within the scope of this research, emphasis is placed on the front-view

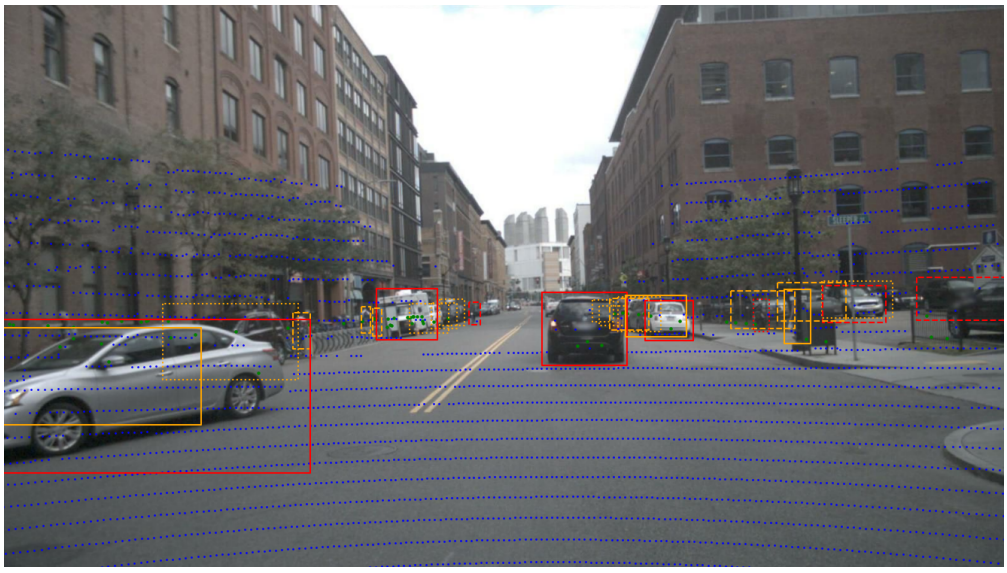

Figure 6.3: An example of a camera image view in the NuScenes dataset, showcasing projected LiDAR points represented in blue, Radar points in green, and accompanying labels. Labels that have a visibility exceeding 50% are denoted by red colour, while those with a visibility below 50% are indicated in orange. Additionally, the style of the lines provides further context: a solid line signifies the presence of both LiDAR and Radar points within the labelled object, a dashed line represents only LiDAR points, and a dotted line signifies the absence of both LiDAR and Radar points within the labelled object.

RGB camera, specifically within a selected FoV, as shown in Figure 6.3. In order to accommodate the computational requirements of the proposed models, the NuScenes front RGB camera images are resized from their original resolution to a reduced size of 512×384 pixels. A letterbox resizing mechanism, inspired by the methodology employed in KITTI preprocessing step, is employed to preserve the aspect ratio of the image and ensure the normalization of pixel values from 0.0 to 1.0 across all RGB channels.

The NuScenes dataset incorporates pointcloud data collected from both LiDAR and Radar sensors. The LiDAR sensor employed in the dataset is a spinning mechanical device featuring 32 vertical beams, which is half the number used in the KITTI dataset, consequently providing sparser pointcloud samples. With a capture frequency of 20Hz, the LiDAR sensor provides a horizontal FoV spanning 360 degrees and a vertical FoV ranging from -30 degrees to 10 degrees at a maximum range of 70 meters. However, for fusion preprocessing efficiency, only a specific front-view grid FoV is selected, resulting in the subset of the complete LiDAR scan. The Radar sensors utilized in NuScenes operate at a frequency of 77GHz using a Frequency-Modulated Continuous Wave method. These sensors have a reported range capability of up to 250 meters. Sampling from the Radar sensors is performed at a capture frequency of 13Hz. For conducted experiments, only the front Radar sensor data is utilized as it covers the designated fusion FoV. In terms

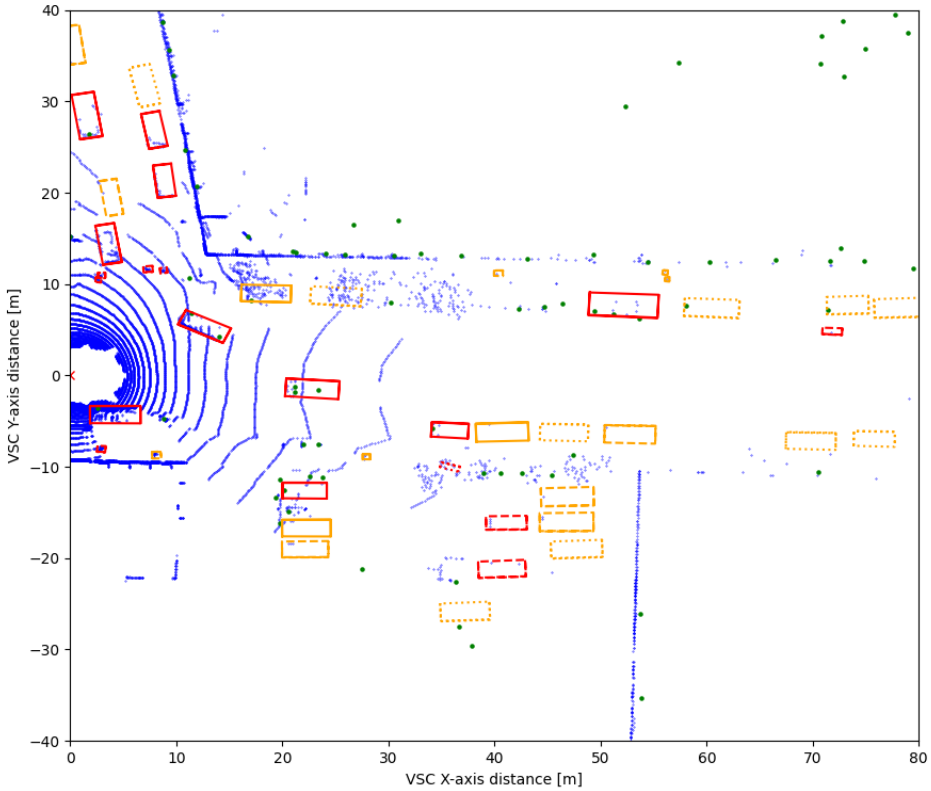

Figure 6.4: The image illustrates a Bird's Eye View in the NuScenes dataset, displaying projected LiDAR points shown in blue, Radar points in green, and labels. This view corresponds to the same scene as in Figure 6.3. The colour-coded and line-style labels are also utilized here, indicating visibility and the presence of LiDAR and Radar points within the bounding box of each label.

of the data statistics, on average, each sample in the pointcloud data contains approximately 14,000 points from the LiDAR sensor, while the Radar data typically includes around 45 points as illustrated in Figure 6.4.

The NuScenes dataset underwent manual 3D annotation by humans, with annotations based on LiDAR pointcloud and camera images. The labels are divided into classes, such as cars, or pedestrians, with sub-classes for specific variations of each class. However, for the purpose of OD perception research, the focus is on the top-level classes, without distinction between e.g. sitting and walking pedestrians, as this level of detail is not required for a general perception system. The utilized classes are cars, trucks, heavy vehicles (like construction machines), bicycles, motorcycles and pedestrians. The labelled data is annotated only in the 3D domain. Thus in camera-based models, the corners of 3D labels are projected onto the camera image plane based on calibration matrices, and the smallest rectangular bounding box containing all the projected points is drawn. The bounding boxes are scaled according to the input image resize factor. For pointcloud and fusion-based detection, the labels are directly obtained from the NuScenes database, as they are already positioned in the VCS. However, post-processing is performed to filter the labels.

Table 6.1: Comprehensive comparison of label's visibility information in the camera, LiDAR, and Radar sensors' data. The visibility statistics were obtained through human annotation in the NuScenes dataset, based on the detailed object information for all label instances. The presented results focus on the overall visibility of all label classes, with a specific emphasis on two crucial ones: cars and pedestrians. Percentage values in the brackets correspond to the total label count of each considered class type.

	All classes	Cars	Pedestrians
Total labels count	549289	219328	116952
Camera visibility over 40%	346475 (63%)	126355 (58%)	72459 (62%)
Labels with LiDAR points	451621 (82%)	170519 (78%)	102295 (87%)
Labels with radar points	173836 (32%)	101049 (46%)	12839 (11%)
Mean LiDAR points per label	97	127	14
Mean radar points per label	2.26	1.96	1.14

For this research, only a front-view RGB camera is used, along with LiDAR and Radar readings within the chosen FoV, which closely match KITTI data. The FoV is determined by the overlap between the point-cloud data and the camera view. In the VCS, the BEV FoV spans from 0m to 80m on the X-axis and from -40m to 40m on the Y-axis. Furthermore, camera-specific labels are filtered to ± 35 degrees horizontal FoV in front of the host vehicle due to the sensor angular resolution limitations. Additional challenges arise regarding the visibility of objects in the selected sensor's view. To address this, the visibility parameters provided in the dataset for each label are utilized. The NuScenes labels deliver information about the visibility of objects in the camera image, as well as the number of LiDAR and Radar points associated with each labelled object. This information is used to filter out labels that are not detectable in the specific sensor setup. Based on the characteristics outlined in Table 6.1, labels with a visibility threshold of over 40% are chosen as the ground truth for camera-based OD. For 3D enhanced BEV, labels with at least one LiDAR or Radar detection are selected. In the fusion process, where the goal is to demonstrate its robustness, the ground truth should either be visible in the camera, have related pointcloud detection, or both.

6.2 Training process

The NN architectures proposed in this study, including single-sensor models and various fusion concepts, were trained separately on the KITTI and NuScenes datasets. However, it should be noted that the pointcloud models utilizing Radar sensor data were trained exclusively on the NuScenes, as this data source is only available in that dataset. To conduct the trainings, both datasets were divided into three parts: training, validation, and testing. For the testing dataset, particular care was taken to ensure that no frame from any test scene was included in either the training or validation sets. This strict separation guarantees proper testing conditions, as the networks were never optimized on those specific data samples. The datasets were divided into approximately 5,000 training, 1,500 validation, and 500 testing samples for the KITTI dataset, and 22,000 training, 5,500 validation, and 4,000 testing samples for the NuScenes dataset.

The training process was conducted using a custom high-level Python ML framework built on top of the open-source PyTorch library. This framework not only facilitated the training of models but also handled functionalities related to the whole data pipeline workflow, hardware utilization, performance metrics calculation, and documentation of conducted experiments via the MLFlow plugin. All trainings were done on a remote computing cluster equipped with several Nvidia 2080 Ti, 3090 Ti and Titan RTX GPUs to accelerate parallel calculations and increase the overall training speed. The entire framework configuration varied for each experiment, with several key aspects being adjusted. These aspects included defining different model architectures, preprocessing input data specific to each sensor and adapting it to different domains, as well as generating target data and calculating loss functions tailored to match proper 2D or 3D predictions of each model. However, there were also common steps shared across the experiments, such as general dataset handling scripts, standard weights optimization process using the gradient-based method and the KPI tests in both the 2D and 3D domains with the same implementation of the metrics across all trials.

6.2.1 Target generation

The generation of targets plays a crucial role in the training process as it sets the goal for the network predictions output, which in turn influences the loss function, gradients calculation, and parameters optimization. In YOLO approaches, targets are created by scattering labels to the best matching anchors based on the IoU metric, resulting in a sparse grid with one object per matching anchor in the specific cell. However, in this thesis, a different approach was employed, based on EfficientDet one. Instead of assigning labels to the best matching anchor, each label was assigned to all possible anchors above a 0.3 IoU threshold, resulting in multiple assignments for each label instance, as shown in Figure 6.5. This approach generates denser target grids with duplicated assignments between different grid scales and neighbouring cells. Additionally, an extra flag was set for each anchor-label pair with an IoU between 0.3 and 0.5, which is used during the loss calculation to ignore the contribution of these assigned pair to the loss value. This approach prevents the network from being penalized for predicting objects in nearby cells. Anchors with an IoU above 0.5 are encouraged to yield positive predictions, as the non-max suppression algorithm handles the combination of

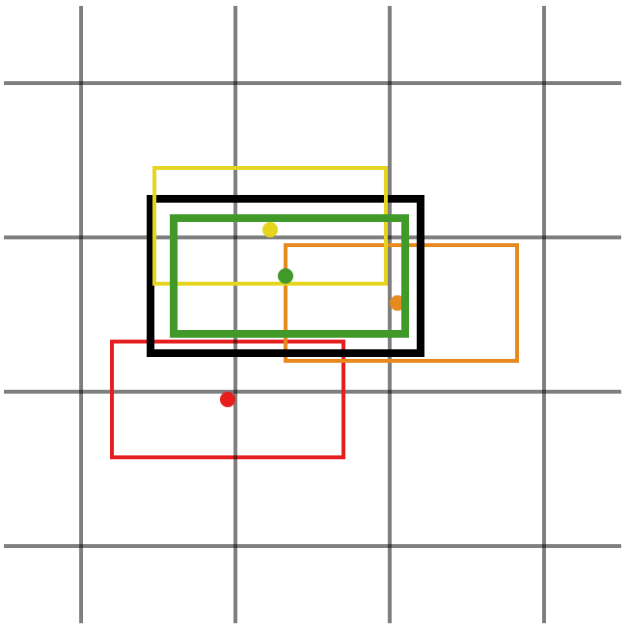

Figure 6.5: An example of anchors association during the target grid generation for a single label instance. The black bounding box represents the label. A green rectangle is the best matching anchor from a middle grid cell. Yellow, orange and red rectangles correspond to additional anchors from neighbouring cells, where IoU is higher than 0.5, between 0.3 and 0.5 and below 0.3 respectively. In the YOLO approach, only a green anchor would be the target, whereas any other predictions would increase the loss function value. With the alternative approach, both green and yellow anchors are considered positive targets, the orange is ignored and only the red one is negative, increasing the loss.

similar predictions during the post-processing step. However, if the best anchors' object should be missed, there are a lot of supplementary ones to still yield proper object prediction. In cases where the IoU metric is between 0.3 and 0.5, indicating a partial match between the anchor and the label, the loss value is ignored for that cell, preventing unwanted parameter updates based on gradient values, which might have a negative impact on the model performance. Finally, cells with an IoU below 0.3 are considered hard negatives, and the model is penalized with a high loss value for any positive predictions made in those cells.

The target generation approach was used for both the 2D and 3D domains, with a few adaptations made for the latter one. The primary difference lies in the domain for which the target grids are generated. In the 2D domain, each grid cell corresponds to a portion of the image in a 2D plane, while in the 3D domain, each cell corresponds to a section of the BEV grid in the VCS. However, despite this distinction, the target tensors' formats remain consistent. For the classification head, the format is $h \times w \times a \times c$, where h and w represent the grid dimensions, a represents the anchor's dimension, and c represents the class vector. For the regression head, the format is $h \times w \times a \times f$, where f represents the objects' features vector. The class vectors are constructed to indicate the presence or absence of a label associated with an anchor at a specific grid position. A value of 0 is assigned when there is no label of a particular class, and a value of 1 is assigned when a label of that class is present. Additionally, for anchors falling between the 0.3 and 0.5 IoU threshold, a value of -1 is assigned across all classes, to ignore the cell values as mentioned previously. On the other

hand, the feature vector encodes the position and size of the bounding box relative to a given anchor. The encoding of the 2D bounding box can be represented as:

$$\begin{cases} x &= x_a + d_x + c_x \\ y &= y_a + d_y + c_y \\ w &= w_a \cdot e^{d_w} \\ h &= h_a \cdot e^{d_h} \end{cases} \tag{6.1}$$

where the left-hand side of the equation represents a tuple that describes the label parameters (x, y, w, h) of the bounding box, including its centre position and dimensions (width and height). On the right-hand side, the label is encoded relative to the selected anchor, denoted by (x_a, y_a, w_a, h_a). The variables c_x and c_y indicate the position of the grid cell in the global domain coordinate system. The above equations can be solved for (d_x, d_y, d_w, d_h) to obtain a target delta features vector, which corresponds to the proper cell and anchor position in the target regression head tensor.

In a similar manner, for the 3D domain, both the class vector and the feature vector can be generated for each cell in the BEV grid. The class vector follows the exact same generation process as in the 2D case. However, to account for additional features, the 3D feature vector can be denoted as follows:

$$\begin{cases} x &= x_a + d_x + c_x \\ y &= y_a + d_y + c_y \\ z &= z_a + d_z \\ l &= l_a \cdot e^{d_l} \\ w &= w_a \cdot e^{d_w} \\ h &= h_a \cdot e^{d_h} \\ yaw &= atan2(d_{sin\_yaw}, d_{cos\_yaw}) \end{cases} \tag{6.2}$$

where the left-hand side encompasses the label parameters $(x, y, z, l, w, h, \text{yaw})$ of the bounding box in the 3D domain. These parameters describe the position of the bounding box centre, its dimensions (length, width, and height), and the yaw angle of rotation in the BEV relative to the VCS Z-axis. On the right-hand side, the label is encoded with respect to the selected $(x_a, y_a, z_a, l_a, w_a, h_a)$ anchor. The variables c_x and c_y indicate the position of the grid cell in the global BEV coordinate system. Unlike c_x and c_y, there is no c_z coefficient since the target grid tensor does not include the height dimension. Additionally, the yaw angle is encoded using two target values, $sin(\text{yaw})$ and $cos(\text{yaw})$, and can be obtained by applying the $atan2$ function to them. By solving the above equations, the target delta features vector $(d_x, d_y, d_z, d_l, d_w, d_h, d_{sin\_yaw}, d_{cos\_yaw})$ can be calculated, representing the appropriate target tensor values for the regression head.

6.2.2 Loss function

The final loss value is determined by combining the results of the loss functions for both the classification and regression heads, which are based on EfficientDet paper. The classification head utilizes the Focal Loss function, which enhances the standard cross-entropy loss by incorporating modulating terms. By dynamically adjusting the scaling factors of the cross-entropy loss, the Focal Loss function reduces its impact as the model's confidence in predicting the correct class increases. This adaptive scaling factor effectively down-weights the influence of easy examples during training, enabling the model to focus on challenging instances. Through this approach, the Focal Loss function guides the learning process by emphasizing difficult examples and mitigating the challenges posed by class imbalance. The formula for the Focal Loss function used for the classification head is as follows:

$$Loss_{cls}(y_{pred}, y_{target}) = \sum_{w_i=1}^{w} \sum_{h_i=1}^{h} \sum_{a_i=1}^{a} loss_c(p_c, t_c)$$

$$\text{where} \quad \begin{matrix} p_c = y_{pred}(w_i, h_i, a_i) \\ t_c = y_{target}(w_i, h_i, a_i) \end{matrix} \quad \text{and} \quad loss_c = \begin{cases} -\alpha(1-p_c)^{\gamma} log(p_c) & t_c = 1 \\ -\alpha(p_c)^{\gamma} log(1-p_c) & t_c = 0 \\ 0 & t_c = -1 \end{cases} \tag{6.3}$$

The classification loss $Loss_{cls}$ is determined by two tensors: the model predictions y_{pred} and the provided label targets y_{target}. The final loss value is obtained by summing each class probability loss $loss_c$ for all anchors a across the grid's dimensions w and h. The specific formula for $loss_c$ varies depending on the target class value t_c and the predicted probability p_c for that class by the model. When an object of a particular class is present in a cell, t_c is equal to 1, and the loss value increases when the predicted probability p_c is lower than 1.0. Conversely, when there is no label data for a given class in the cell, t_c is equal to 0, and the model is penalized with a larger loss if it predicts a non-zero probability for that class. Additionally, a special mechanism is employed for anchor-label pairs with an IoU between 0.3 and 0.5. In this case, the loss value is ignored when the targets t_c for a given anchor are set to -1. This mechanism ensures that the loss calculation does not negatively impact predictions for anchors in this IoU range. The Focal loss formula incorporates two parameters, α and γ, which are essential for adapting the loss function to tackle class imbalance and emphasize challenging examples. The values suggested by the authors of the Focal loss method for these parameters are 0.25 and 1.5, respectively. Such values shape the loss function curve in a way that provides a balance between specific different class examples in dataset samples.

The regression head employed a combination of weighted square error and absolute error losses. The choice between the two was determined by the difference value between the targets and predictions, with a threshold set at $\frac{1}{9}$, chosen through the trained model's performance experiments. The coefficients for the loss functions were carefully selected to ensure a smooth transition at the threshold, effectively balancing the contributions of the two methods. The inclusion of the absolute error loss serves the purpose of obtaining

a consistent loss value that is proportional to the difference between the target and predicted feature values, particularly when they differ significantly. Contrarily, when the difference is smaller than the threshold and approaches zero, the loss decreases exponentially towards zero as well. The formula for this loss is expressed as:

$$Loss_{reg}(y_{pred}, y_{target}) = \sum_{w_i=1}^{w} \sum_{h_i=1}^{h} \sum_{a_i=1}^{a} \sum_{f_i=1}^{f} loss_f(p_f, t_f)$$

$$\text{where} \quad \begin{matrix} p_f = y_{pred}(w_i, h_i, a_i, f_i) \\ t_f = y_{target}(w_i, h_i, a_i, f_i) \end{matrix} \quad \text{and} \quad loss_f = \begin{cases} 0.5 \cdot 9 \cdot (t_f - p_f)^2 & |t_f - p_f| < \frac{1}{9} \\ |t_f - p_f| - \frac{0.5}{9} & |t_f - p_f| \geq \frac{1}{9} \end{cases} \tag{6.4}$$

The regression loss, denoted as $Loss_{reg}$ in the formula, is calculated using the model predictions y_{pred} and the provided label targets y_{target}. The final loss value is computed by summing the feature losses $loss_f$ for each feature of all objects across w, h, a and f tensors dimensions. The value of $loss_f$ is determined based on the feature target value t_f and the predicted feature value p_f. Despite the distinct meanings of the features, the objective remains the same - to minimize the difference between the target and predicted values as much as possible. Therefore, the same loss function can be applied to all features.

The overall loss value for a given input sample is obtained by combining the classification and regression losses across all scales of the model prediction heads. This combined loss serves as a starting point for the backpropagation optimization algorithm during the training process. Through gradient-based optimization, the model parameters are iteratively adjusted to minimize the calculated loss value, thereby improving the performance of the model.

6.2.3 Optimization tools

The gradient-based optimization algorithm is a crucial component of the training process, alongside the loss function. Among the various optimizer variants available, the ADAM (Adaptive Moment Estimation) optimizer was utilized in all conducted trainings. ADAM is a widely adopted algorithm in DL that adjusts the learning rates of model parameters based on historical gradients and momentum. This dynamic adjustment facilitates more efficient learning and faster convergence of the NN towards the optimal parameter values that minimize the loss function. Considering the size of the automotive datasets used and the numerous architectures and experiments conducted, the benefits of ADAM are particularly significant. ADAM is also well-regarded for its effectiveness on noisy and sparse datasets, which is highly advantageous for the NuScenes dataset. As demonstrated earlier, occlusions and challenges in sensor data labelling can potentially impact performance. However, the inclusion of momentum in ADAM helps address such issues and mitigate their impact during optimization. The optimizer was used with default values of betas coefficients used for computing running averages of gradients $\beta_1 = 0.9, \beta_2 = 0.999$ and epsilon $\epsilon = 1e^{-8}$ term added to the denominator to improve numerical stability.

One of the most critical hyperparameters in the training process is the Learning Rate (LR). The LR, represented by the symbol α, is crucial in determining the speed at which an algorithm updates or learns the parameter values. Specifically, the learning rate controls the magnitude of weight adjustments in the NN based on the loss gradients. The initial value of LR parameter is of the utmost importance for the whole training process. To address it, the value of α was the subject of hyperparameters tuning, which sets the final value of $\alpha = 3e^{-5}$ across all the experiments. However, the need for different weight updates magnitude changes over the course of the training. For that, various LR schedulers are used, that dynamically adjust the α value based on the given formula. Among those, the cosine annealed warm restart learning scheduling method (Loshchilov and Hutter 2016) has been utilized, given by:

$$\alpha_t = \alpha_{min} + \frac{1}{2}(\alpha_{max} - \alpha_{min})(1 + cos(\frac{T_{cur}}{T_i}\pi)) \tag{6.5}$$

where current α_t depends on the combination of initial LR value α_{max} and minimal one, set as a threshold parameter to $\alpha_{min} = 1e^{-7}$. The proportion between the two is controlled by a cosine function of a ratio between the current training epoch since the last reset T_{cur} and the reset period parameter of $T_i = 8$ epochs.

The scheduler utilized in this training process combines two components: cosine annealing and warm restarts. Cosine annealing refers to the use of a cosine function as the LR annealing function. Empirical evidence has shown that the cosine function performs better compared to simple linear annealing methods. It allows for a smoother transition of the LR during training. Warm restarts, on the other hand, involve periodically resetting the α_t parameter to its initial value after a specified number of optimization iterations. This mechanism enables the optimization process to break away from local minima and explore the search space for potentially better global minima. It provides a way to introduce exploration and prevent the algorithm from getting stuck in suboptimal solutions.

6.2.4 Hyperparameters tuning

In order to ensure optimal model training, numerous hyperparameters need to be fine-tuned. Hyperparameters refer to the higher-level parameters, such as the LR, that configure the model and training process. Tuning these hyperparameters involves conducting multiple training iterations with different settings to identify the values that yield the best results in terms of the KPI metrics. With many different trainings performed while tuning each model hyperparameters, it is hard to present every single development done during this thesis research. To that end, this section provides an overview of hyperparameter tuning across various aspects, including normalization layers, activation functions, trainable or frozen backbones, and training duration. The best-performing solutions for each aspect have been selected, and the overall performance of the models is described in the following sections based on these settings.

Normalization layers rescale tensors' values during neuron activation to ensure their sufficient range, which helps with gradient propagation and generally improves the training. The experiments were conducted with BatchNorm, InstanceNorm, LayerNorm, and GroupNorm. Ultimately, LayerNorm was chosen

as it consistently delivered the best overall results when combined with the CDSM rotation operation. This is attributed to the specific behaviour of LayerNorm, which applies normalization along the spatial VCS dimensions of the intermediate tensors.

Different activation functions were tested across all models. Although the impact on results was minimal, a slight difference was observed when using a combination of LeakyReLU and Mish functions compared to plain ReLU or Sigmoid ones. This is because the gradients propagate more effectively for negative activations with LeakyReLU and Mish, whereas ReLU function returns zero value for all such cases.

Another area of hyperparameter tuning involves the use of freezing the backbone mechanism. For pretrained backbones and fusion architectures that utilize single-sensor models, a choice can be made between optimizing the entire network structure or freezing the pretrained parts. Freezing the selected parts implies that gradients are not applied to them, and only the subsequent elements of the architecture are trained, effectively treating the frozen parts as the fixed input. However, it was found that training the entire structure in an end-to-end manner yielded better results.

Finally, regarding training duration, the networks were trained until the early stopping mechanism did not interrupt the process. Early stopping monitors the loss function value on the validation set, which is separate from the training set used for parameter optimization. When the model starts overfitting to the training data, resulting in a significant increase in validation loss, training is halted to prevent further overfitting. The threshold limit for consecutive epochs with no improvement in validation loss, referred to as the patience parameter, has been set to 10 epochs.

Chapter summary:

- An overview of the open-source datasets used throughout the thesis was conducted. KITTI and NuScenes sensor suites were described, presenting data samples from each dataset. Additionally, the datasets' labels were discussed, especially considering the NuScenes, where the annotation process in 3D poses a challenge in some single-sensor and fusion setups.

- The common training process was introduced, highlighting the standard elements of every experiment conducted. Those shared elements are the target generation methods for the image and BEV grid domains, classification and regression loss functions for both prediction heads, optimization tools such as ADAM optimizer and LR scheduler, and default hyperparameters set for training pipeline. Those parameters are the results of dozens of training configurations tried for each of the models.

Chapter 7

Single-sensor results

Chapter highlights:

- *Separate single-sensor models experiments*
- *Visualizations and KPI metrics evaluation*

The initial experiments revolve around training single-sensor models for several important reasons. Firstly, these models are crucial components of the fusion architecture. Each sensor's data extraction is performed using a dedicated model, and without them, the fusion concepts cannot be effectively implemented. The weights acquired during these trainings serve as initial parameters for the fusion models in the corresponding sub-networks of the architecture. Moreover, the outcomes of the camera-only, LiDAR-only, and Radar-only NNs are examined to evaluate the capabilities of each sensor. This analysis not only demonstrates the effectiveness of the proposed architectures but also facilitates a comprehensive comparison of the advantages and disadvantages for OD tasks. The evaluation process is based on a set of KPIs described in Chapter 4. These KPIs serves as essential metrics to assess the performance and effectiveness of the trained models. By analyzing and comparing these indicators, valuable insight into the capabilities and limitations of each approach is gained.

The experiments described in this chapter are organized into sections according to the sensors used for the networks' input data and the specific domain in which the final predictions are made. Based on the theoretical foundations discussed in Chapter 5 and the training process explained in Chapter 6, the implementation and training of all proposed models are thoroughly executed. The findings of these experiments are presented through a combination of visual predictions and KPIs metrics. The chosen metrics are specifically selected for the domain of predictions being made. By utilizing both visual and numerical results, a comprehensive evaluation of the models' performance is achieved. Subsequently, each model undergoes a detailed analysis. The results obtained from individual sensors are examined, shedding light on their independent contributions. Additionally, the practicality for the fusion approach is carefully considered and discussed. This analysis includes a critical assessment of the fusion-related benefits and its overall impact on the predictions.

7.1 2D camera model

The very first model trained in this study focuses on 2D camera image OD. The model's output consists of 2D bounding boxes that are located on the image plane. To build this model, the proposed architecture utilized the EfficientNet backbone. The advantage of using this backbone is that it allows for the reuse of pretrained weights, which were originally obtained from training on the ImageNet computer vision general dataset. This approach leverages transfer learning, enabling the network to benefit from the pre-existing knowledge stored in the backbone weights from the beginning of the training process. Although the backbone was initialized with ImageNet weights, it was not kept frozen. On the other hand, the remaining parts of the model, including the BiFPN and prediction heads, were initialized with random initial parameters using the Xavier initialization method. These components were optimized together with the backbone during the training process. The training procedure followed the conditions discussed earlier, ensuring consistency across the experiments. The model required image data and corresponding labels, which were sourced from either KITTI or NuScenes dataset. Therefore, the training results and evaluation are presented for both of these data collections, providing a comprehensive assessment of the model's performance across different datasets.

Figure 7.1: The results of camera-only 2D OD model on KITTI dataset test image samples. Blue bounding boxes represent dataset labels and yellow ones are model predictions. Both bounding box types have class name indicators above, along with confidence scores for the model predictions. "TP" denotes True Positive relationship between prediction and label according to the association method employed.

The trained model predictions are presented in Figure 7.1. The performance of the model can be characterized as satisfying overall. The results demonstrate the model's ability to detect all labelled instances accurately, with minimal false prediction or missed target cases. The accurate predictions and close alignment with the ground truth prove the effectiveness of the model in capturing and understanding the visual features based on an initial single input image. This indicates that the model has successfully learned to identify and localize objects within the 2D camera images. Furthermore, the model's successful performance on the KITTI dataset indicates its generalization capability. The dataset comprises diverse real-world driving scenarios, encompassing various environmental conditions, object types, and occlusion levels.

In evaluating the performance of the 2D OD model, several KPIs are employed to assess its accuracy and effectiveness in that particular domain. First of all, the association at IoU20 indicates true positive matches between predictions and labels at the thresholded similarity score of 0.2. On top of that, the Mean Average Precision (mAP) is the most essential metric that captures the combined performance of precision and recall at various thresholds. This metric provides a comprehensive evaluation of the model's overall effectiveness. It serves as a key comparison metric throughout this thesis, playing a significant role in assessing and comparing different models. In addition, supplementary KPIs such as precision, recall, and F1 score were utilized to provide further insights into the model's performance.

Table 7.1: 2D OD KPI metrics evaluated on KITTI test dataset. Labels and predictions matching is done with the use of IoU20 association method for 2D image bounding boxes. The results present general KPI values for all objects, as well as each of the 4 individual classes separately. ↑ indicates the higher metrics' values are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑
All	0.635	0.861	0.856	0.858
Car	0.891	0.875	0.922	0.898
Pedestrian	0.620	0.673	0.708	0.690
Truck	0.633	0.915	0.684	0.783
Bicycle	0.398	0.592	0.442	0.506

The model KPIs results, presented in Table 7.1, show a high mAP scores across all classes, reflecting its strong performance in an OD task. Overall, the model demonstrates solid performance, with high metrics values across the board. The reliable mAP scores across all classes suggest that the model is consistently accurate in detecting objects across various categories. Such a camera-only model's performance is particularly suitable for integration into a fusion architecture. The model's ability to provide firm image feature maps makes it a desired component for image data preprocessing in preparation for a CDSM fusion.

When evaluating the same NN architecture trained on the NuScenes dataset, the model achieves similar but slightly worse performance compared to the KITTI dataset experiment. This outcome can be attributed to the NuScenes being larger and containing more challenging cases. General model performance is still sufficient for overall satisfactory results, as shown in Figure 7.2. Although there are instances of false positive predictions, most of them are related to the lack of labels, as in the middle of the top left and top right im-

Figure 7.2: The results of camera-only 2D OD model on NuScenes dataset test image samples. Labels and predictions overlay presentation are consistent with the one used in Figure 7.1. Additionally, orange "FP" bounding boxes represent False Positive detections, not associated with any labelled objects.

ages. These false positives can be attributed to the dataset's limitations in providing 2D labels rather than the model's shortcomings. On the other hand, the model still demonstrates good accuracy in terms of bounding box estimation for objects' width and height. This underscores the advantage of using camera sensor-based architecture. The model effectively captured and estimated the dimensions of objects, contributing to accurate and reliable bounding box predictions based extracted on 2D image features.

The performance evaluation of the KPIs, presented in Table 7.2, utilizes the identical 2D metrics as previously. First and foremost, they reveal significant differences in the model's performance, that vary depending on the specific object class being detected. For the car class, the observed accuracy is quite good, which can be attributed to its plurality as the most common class in the dataset. The model demonstrates accurate and reliable detection results for cars, leading to high precision and recall values, resulting in a favourable mAP and F1 score overall performance for this class. On the other hand, the performance for the trucks and pedestrians classes shows signs of degradation over KITTI. This can be attributed to the fact that labels for these classes are less frequent in the dataset. Moreover, classes such as bicycles, motorcycles, and heavy vehicles occur rarely in the dataset. As an outcome, the model encountered difficulties in accurately detecting and localizing instances of these classes, leading to even lower performance. The rarity of these classes limits the model's exposure to such instances during training, affecting its ability to generalize and achieve high accuracy. Additionally, the mapping of original 3D labels into 2D space introduces lots of occluded targets in the camera view. Such targets, even after discussed filtration, pose a challenge for the model during the optimization process as well, as visible features do not match internally learned representation.

Table 7.2: 2D OD KPI metrics evaluated on NuScenes test dataset. Labels and predictions matching is done with the use of IoU20 association method for 2D image bounding boxes. The presented KPI metrics values relate to all objects, as well as each of the 6 individual classes separately. ↑ indicates the higher metrics' values are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑
All	0.323	0.713	0.514	0.597
Car	0.741	0.731	0.699	0.715
Pedestrian	0.446	0.574	0.398	0.470
Truck	0.414	0.691	0.359	0.473
Heavy	0.127	0.165	0.295	0.212
Bicycle	0.078	0.139	0.096	0.114
Motorcycle	0.139	0.202	0.172	0.186

The upcoming NuScenes experiments further investigate the variations in class performance. However, addressing such an issue can be viewed as an entirely separate research topic. In this thesis, to ensure a meaningful comparison with other models, the car class's performance is used as a primary indicator due to its similarity across the NuScenes and KITTI datasets. Focusing on this class allows for a direct assessment of the model's ability to detect and localize cars, which are the most popular class in both datasets. Analyzing the model's performance across different datasets provides insights into its generalization capabilities and its handling of similar object categories. This comparative analysis helps identify whether performance differences are due to dataset-specific challenges or inherent model limitations. Moreover, evaluating the model's performance on the car class serves as a reference point for understanding its overall OD capabilities.

7.2 3D LiDAR model

The second model trained utilizes 3D point cloud data as its input. The input data for the particular model discussed in this section comes from the LiDAR sensor, whereas the Radar-only model will be presented in the following section. In Chapter 2, the general advantages and disadvantages of a LiDAR sensor have been discussed. While it may be challenging to envision the widespread use of LiDAR sensors in mass-production cars due to said reasons, the experiments conducted in this study provide valuable insights into its comparison with other devices within an AD sensor suite. Furthermore, a LiDAR-based 3D OD system serves as a foundational baseline for evaluating and benchmarking other algorithms in the 3D domain.

Unlike the previous model, this one was trained from scratch without leveraging any preexisting weights. The DLA backbone, BiFPN and prediction heads, were initialized randomly using the Xavier method. The 3D LiDAR architecture was subject to the standard training procedures employed in this study. Through it, the architecture components were optimized to learn and extract meaningful features from the LiDAR pointcloud data in a 3D BEV domain. Both the KITTI and Nuscenes datasets provide LiDAR data, allowing for comprehensive training and evaluation of the model's performance on different datasets. This enables a thorough assessment of the model's ability to generalize across various sensors' specifications.

Figure 7.3: The results of LiDAR-only 3D OD model on KITTI test set. The visualization consists of two distinct views for each sample, namely the camera image and the BEV perspective. In the former, blue and magenta bounding boxes represent model predictions and false positives respectively. In the latter BEV, supplemental target labels are marked in green and missed detections in yellow. The LiDAR pointcloud data overlay is present in BEV but bypassed in the camera image for better overall visibility.

OD results obtained from the model training on the KITTI dataset are presented in Figure 7.3. Despite the inherent challenges posed by the 3D prediction domain, the utilization of depth readings and the sparsity of LiDAR pointcloud data provide sufficient information to achieve accurate object detections on a BEV grid. The quantity of sensor detections associated with each object enables the model to extract features that correspond to the object's accurate size and even determine its class based on the learned representation. Additionally, the distance measurements provided by each detection allow for precise localization of an object relative to the host vehicle. As a result, the model excels in the 3D OD task, demonstrating high performance and accuracy.

In the evaluation of KPIs for the 3D OD task, similar main metrics such as precision, recall, F1 score, and mAP, can be used as in the 2D domain. However, considering the unique challenges and benchmarks in the 3D domain, additional metrics are employed to provide a more comprehensive characterization of the solution's performance in terms of predicting various aspects of 3D objects including their location, size, and orientation in the form of Mean Average Translation Error (mATE), Mean Average Size Error (mASE), and Mean Average Orientation Error (mAOE) respectively. On top of that, the NuScenes Detection Score (NDS) is used, which combines these aspects into a single value metric alongside mAP. The results of the KPI evaluation for the 3D LiDAR-only model are presented in Table 7.3, showcasing the model's performance across these metrics for KITTI dataset.

Table 7.3: 3D OD KPI metrics evaluated on KITTI test set based on LiDAR pointcloud data. Labels and predictions matching is done with the use of the 3D association method DIST2. The results of the KPIs reflects overall performance, as well as each individual class one. In addition to general performance KPIs, dedicated 3D metrics are also presented. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑	mATE↓	mASE↓	mAOE↓	NDS↑
All	0.722	0.868	0.819	0.843	0.339	0.469	0.370	0.664
Car	0.875	0.883	0.860	0.872	0.303	0.390	0.222	0.785
Pedestrian	0.649	0.726	0.656	0.689	0.308	0.605	0.879	0.525
Truck	0.708	0.824	0.782	0.802	0.547	0.381	0.098	0.682
Bicycle	0.657	0.775	0.616	0.687	0.198	0.5	0.282	0.665

An important aspect of the KPI evaluation that differs for 3D models is the association method used. While the 2D image association employed the IoU20 method, the more popular distance-based method called DIST2 is utilized for 3D models. This approach involves matching predictions and labels based on the absolute Cartesian distance between the centres of the cuboids. Specifically, the DIST2 method classifies a prediction and label pair as a true positive example when the distance between their centres is smaller than 2 meters in the BEV domain. By employing this distance-based approach, the association process incorporates more relaxed conditions, allowing for a broader range of matching criteria. However, metrics such as mATE, mASE, and mAOE provide additional information regarding the accuracy of the two cuboids. While the distance-based approach provides a general measure of association, these additional metrics provide a more detailed analysis of the quality and alignment of the predicted and ground truth cuboids.

The LiDAR processing architecture, similar to the camera-only model trained on the KITTI dataset, yields highly accurate results across all classes. While the car class exhibits the highest values of the KPI metrics, the performance of other classes is also within satisfying ranges. This experiment validates the effectiveness of the proposed concept and confirms the successful implementation of the model. Furthermore, the experiment establishes a solid foundation for the pointcloud processing sub-model architecture. The voxelization and feature extraction techniques employed in this task demonstrate their effectiveness in generating meaningful representations of sensor data in the form of feature maps. These feature maps play a crucial role in the forthcoming fusion solution, providing valuable information that will contribute to the overall perception system.

The same model architecture was also trained on the LiDAR data from the Nuscenes dataset, and the visualizations of the model predictions can be seen in Figure 7.4. It is important to highlight the difference in data format between the two sensors. The LiDAR sensor in the KITTI dataset provides a much denser pointcloud compared to the one in the Nuscenes dataset, as presented before. This discrepancy has explicit consequences on the OD performance, as the model has less information to work with when determining the exact size, position, and class of each object in the Nuscenes LiDAR data. This could be particularly observed in faraway instances, like the bottom scene far left car, which is less accurate or even completely

Figure 7.4: The results of LiDAR-only 3D OD model on NuScenes test set. The visualization specification matches the one used in Figure 7.3. The major difference could be observed in a LiDAR pointcloud data overlay. NuScenes sensor provides sparser detections within shorter ranges, which has a direct impact on the model performance at long-range detections.

missed as the track ahead in the middle scene. However, the model handles the overall OD task well with fairly accurate predictions. For the car class, they come close to the vision-only mAP score, but in this far more challenging 3D domain. This could be attributed to the fact, that the camera perspective visibility occlusion does not pose a problem when it comes to the labels, as the 3D is their native domain. Nevertheless, the distribution of labels remains a challenge in the NuScenes dataset, as there are significantly more instances of cars compared to other classes.

When evaluating the KPIs on the NuScenes dataset, presented in Table 7.4, a similar trend to the 2D model can be observed. The car class detections exhibit high metric values, indicating frequent and accurate high-confidence predictions across the majority of the test samples. However, the performance of other classes diminishes even further compared to the vision-only model. On one hand, the model's structure proves capable of handling 3D pointcloud data, as evident from the results on the KITTI dataset as well as the car class in the NuScenes dataset. On the other hand, a significant performance drop is observed among the individual classes in the NuScenes dataset. The most probable reason for this is, once again, the class imbalance in the labelled data. This imbalance causes most of the optimization iterations to focus primarily on car targets, resulting in the model being optimized specifically for car detection. Additionally, the sparser

Table 7.4: 3D OD KPI metrics evaluated on NuScenes test set for LiDAR pointcloud processing architecture. Labels and predictions matching is done with the same DIST2 3D association method. The results present selected 3D KPIs values for all objects, as well as each individual class present in the NuScenes dataset. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑	mATE↓	mASE↓	mAOE↓	NDS↑
All	0.254	0.752	0.517	0.613	0.591	0.576	1.139	0.266
Car	0.740	0.794	0.708	0.749	0.524	0.492	0.556	0.608
Pedestrian	0.276	0.333	0.388	0.358	0.299	0.617	1.539	0.318
Truck	0.277	0.448	0.337	0.384	0.862	0.534	0.672	0.293
Heavy	0.126	0.265	0.166	0.204	0.935	0.608	1.363	0.139
Bicycle	0.081	0.091	0.098	0.090	0.445	0.602	1.245	0.199
Motorcycle	0.108	0.187	0.113	0.141	0.336	0.629	1.563	0.226

LiDAR data in the NuScenes dataset leads to fewer detections overall. This becomes particularly harmful for small objects such as pedestrians, bicycles, and motorcycles, as the lower resolution drastically reduces the amount of information available for accurately detecting these objects in the input data. It is likely that the combination of these factors contributes to the poor performance observed in these classes.

Taking these factors into consideration, the experiments conducted with the 3D LiDAR-only model provide valuable insights for the overall AD perception research. They demonstrate the capability of the proposed architecture to effectively process 3D pointcloud and accurately predict objects in the BEV grid. Moreover, the model performs exceptionally well on the KITTI dataset across all classes, showcasing its proficiency in OD tasks. When evaluating the model on the NuScenes dataset, the performance of the car class also proves to be satisfactory. Lacking performance for other classes poses a problem, but for fusion purposes, the focus on car class only already provides a viable means of comparing the results, especially taking into account the Radar-only model findings described in the next section. This concludes the 3D LiDAR-only model, which can be effectively integrated as a component within a fusion architecture solution.

7.3 3D Radar model

The third single-sensor network architecture focuses on utilizing Radar pointcloud data to generate 3D object detections. The structure of this model is identical to the previously discussed LiDAR-only architecture, with the only difference being the source and type of input pointcloud data and corresponding input grid resolution. The training procedure followed the same approach, with the hyperparameters set exactly as before. This consistency ensures a comparable training process and allows for a direct comparison between the LiDAR-only and Radar-based models. Additionally, it's worth noting that the Radar pointcloud data is exclusively available in the NuScenes dataset. Consequently, it is feasible to train the model solely on this dataset, completely disregarding the KITTI dataset. Despite the similarity in pointcloud format, there are significant differences between the two sources. The most notable distinction concerns the number of detections provided in each sample, as previously shown in Figure 6.4. While a LiDAR pointcloud typically

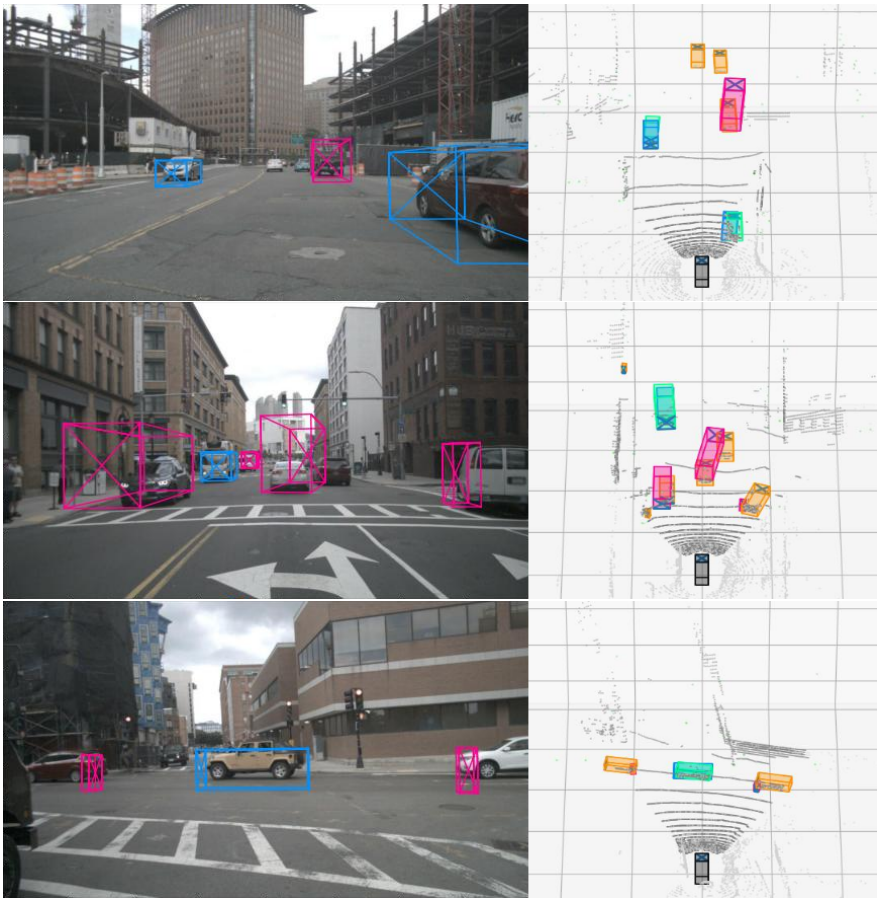

Figure 7.5: The results of Radar-only 3D OD model on NuScenes test set. The visualization overlay follows the same scheme, with labels, positive detections, false detections, and missed ones marked in green, blue, magenta and yellow respectively. Compared to previous experiments, Radar-only model provides significantly lower visual 3D OD results.

contains several thousand points, Radar data tends to be much sparser, often comprising only around 50 detections. This disparity in data density has profound consequences, as the sparse nature of Radar data does not provide sufficient information for accurate 3D object predictions.

Upon visual examination of the results illustrated in Figure 7.5, it becomes evident that the predictions of the Radar-only model pose certain challenges. Comparing the results to previous experiments, there is an increase in false positives and missed object instances. False positive detections, observed in all samples, often occur in locations where actual objects are present, but due to incorrect size and centre position estimations, these detections are not associated with their corresponding target labels accurately. Conversely, some objects, such as two cars in the far distance in the top sample, remain completely undetected. The sparsity of Radar pointcloud data significantly contributes to this issue. With a limited number of detections available, the predictions often rely on just single radar sensor reading. Although it provides accurate distance measurements in the BEV, predicting objects' dimensions and rotation based on a single data point is unrealistic or even impossible in a wide range of real-world scenarios on the road. Consequently, the radar-only model encounters difficulties in accurately predicting target labels. Addressing this challenge, some research suggests aggregating detections from multiple past frames. However, for the fusion use case,

Table 7.5: 3D OD KPI metrics evaluated on NuScenes test set for a Radar-only model. Labels and predictions matching is done with the same DIST2 3D association method as in LiDAR-only experiments. The results present selected 3D KPIs values for all objects, as well as each individual class present in the NuScenes dataset. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑	mATE↓	mASE↓	mAOE↓	NDS↑
All	0.081	0.400	0.307	0.348	0.812	0.670	0.534	0.204
Car	0.324	0.452	0.410	0.430	0.811	0.613	0.395	0.358
Pedestrian	0.046	0.202	0.084	0.119	0.441	0.746	0.611	0.223
Truck	0.116	0.190	0.141	0.162	1.184	0.651	0.597	0.183
Heavy	0.0	0.0	0.0	0.0	-	-	-	0.0
Bicycle	0.0	0.0	0.0	0.0	-	-	-	0.0
Motorcycle	0.0	0.0	0.0	0.0	-	-	-	0.0

a single-frame detector is required. Nevertheless, in theory, fusion with camera image data should help with this issue by providing complementary information.

The KPIs evaluation presented in Table 7.5 confirms that the results of the Radar-only model are significantly lower compared to the LiDAR-only model. Apart from the car class, the metrics for all other classes are close to zero. A possible explanation of this effect could not only be due to the class imbalance issue but also the limitations of Radar sensors in detecting smaller objects like pedestrians. Radar sensors are more effective in detecting firm metal surfaces, such as cars, rather than the biological bodies of pedestrians. Additionally, for the truck class, which should reflect Radar waves consistently, the combination of low data sparsity and large truck sizes causes errors in location and size estimation, leading to missed match associations for this particular class. Such errors in location and size measurements have a significant impact on the Radar-only model's performance, resulting in decreased accuracy.

The conducted experiments reveal the relatively low performance of the Radar-only architecture. However, considering the described limitations of Radar sensors and the adoption of a simple single-frame approach to the OD task, the obtained results are within expectations. Furthermore, visual interpretation of the predictions demonstrates the model's efforts to detect objects in proximity to the actual targets, but the lack of accuracy prevents successful associations with the ground truth, resulting in false or missed detections. In light of these observations, the fusion of Radar data with camera image data holds the potential to significantly enhance the overall quality of perception.

7.4 3D monocular camera model

The last single-sensor architecture experiments circle back to the camera model. While the 2D bounding box detections in the image demonstrate high performance, comparing these results directly with the 3D predictions from LiDAR and Radar networks can be challenging due to different operating domains. Since the fusion results will also predict objects in the 3D BEV, it is essential to enable the evaluation of all proposed methods in the same 3D detection domain. To that end, the camera sensor architecture for 2D image

OD is the only one that lacks such an option. To address this need, a 3D monocular camera architecture, as described in Chapter 5, was developed and subjected to a comprehensive evaluation of KPIs.

This approach utilizes a previously trained model to extract 2D image features. Additionally, the CDSM rotation layer is employed to transform these features onto a BEV grid, enabling the final 3D OD in that domain. The overall model architecture employs the trained EfficientNet backbone and BiFPN in the same manner as the 2D architecture. The addition of the CDSM rotation layer separates the reused components from the subsequent prediction heads, which are replaced with ones specifically designed for the 3D domain, similar to those used in pointcloud networks. Regarding the training process, the backbone and BiFPN weights are transferred from the 2D model, while the remaining parts of the architecture are initialized with random parameters using the Xavier method. Through the fine-tuning process, it was found that refining the backbone and BiFPN parts, along with the rest of the network, leads to improved results. Therefore, these elements were not frozen during training, but rather they underwent a gradient-based optimization process as well. The preparation of targets and their processing follows the same approach as in the 3D models, and the 3D loss function is utilized to obtain the loss value. By implementing and testing this camera-only model, it becomes possible to obtain 3D object detections from the camera sensor, providing a valuable comparison between the camera-based and the fusion system's capabilities in the 3D domain.

Figure 7.6: The results of monocular camera-only 3D OD model on KITTI test set. The visualization overlay follows a previously introduced colour coding scheme, with labels, positive detections, false detections, and missed ones marked in green, blue, magenta and yellow respectively. The pointcloud data is added for enhanced BEV visualization, however, it was not used for obtaining the 3D predictions, which are based on camera image alone.

The visual results of the trained monocular 3D camera model on the KITTI dataset are presented in Figure 7.6. Initial observations indicate a strong performance by the model. It successfully detects the majority of objects across all examples with remarkable precision, considering that it relies solely on image data. The model demonstrates high accuracy in localizing nearby objects, providing precise position estimations. However, challenges arise when dealing with distant objects. Although the network correctly predicts these objects, the depth estimation required for this task introduces a margin of error significant enough to fall above the DIST2 association threshold. Consequently, the model struggles to establish accurate position and correctly predict distant objects. However, it is important to note that these cases are isolated incidents, and overall, the performance of the model can be described as highly satisfactory.

The KPI metrics of the 3D camera model on the KITTI dataset, as shown in Table 7.6, provide valuable insights into its performance. While the metrics are slightly lower than those of the 2D image detection model, they still indicate a strong overall performance. It is worth emphasizing that 3D prediction poses significant challenges due to the added complexity of depth estimation. While examining the performance across different object classes, pedestrians and bicycles stand out as classes with a notable decline in performance compared to the 2D variant of the network. The significant drop in mAP for these classes highlights the difficulty of accurately detecting and localizing smaller objects. The model's ability to estimate depth accurately plays a crucial role in the detection of these objects, and limitations in this aspect result in reduced performance. Despite these challenges, it is important to acknowledge the outstanding ability of the camera-only model in predicting objects in 3D space. Considering the inherent limitations of relying solely on camera data without the aid of other sensors, the achieved results can be regarded as highly impressive. The model demonstrates a commendable level of accuracy and reliability, particularly when detecting larger objects and objects in close proximity. This highlights the potential of monocular camera-based 3D object detection and its significance in perception systems.

Table 7.6: 3D OD KPI metrics evaluated on KITTI test set for a monocular camera model. Labels and predictions matching is done with the DIST2 3D association method. The results present 3D KPIs values for each individual class as well as for all objects together. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑	mATE↓	mASE↓	mAOE↓	NDS↑
All	0.549	0.685	0.693	0.689	0.683	0.593	0.306	0.511
Car	0.719	0.740	0.702	0.721	0.710	0.519	0.262	0.611
Pedestrian	0.485	0.677	0.565	0.616	0.664	0.768	0.752	0.378
Truck	0.699	0.875	0.677	0.763	0.726	0.381	0.044	0.657
Bicycle	0.293	0.404	0.475	0.436	0.634	0.702	0.166	0.396

During the comprehensive evaluation on the NuScenes dataset, the performance of the 3D monocular camera-based OD model demonstrates a decreased level of accuracy compared to the KITTI dataset. Figure 7.7 visually presents the model's predictions on this dataset, revealing a mixed performance. While there are instances of accurate detections, particularly in closer ranges similar to the KITTI dataset, the model

Figure 7.7: The results of monocular camera-only 3D OD model on NuScenes test set. The visualization overlay follows a previously introduced colour coding scheme: labels, positive detections, false detections, and missed ones are marked in green, blue, magenta and yellow respectively. The pointcloud data is added for enhanced BEV visualization, however, it was not used for obtaining the 3D predictions, which are based only on camera image.

encounters significant challenges in accurately estimating the depth of objects across a larger number of instances. Consequently, this leads to the occurrence of false positive detections and missed targets, as evidenced by the distant cars in the first two scenes presented. The third scene represents an interesting scenario where all objects are detected by the model, and from the camera image perspective, they seemingly correspond to actual cars. However, upon further analysis from a BEV perspective projection, it becomes clear that all predictions are misaligned with the actual targets, exhibiting differences exceeding two meters in distance. This misalignment further underscores the limitations of the 3D monocular camera model in accurately estimating the depth of objects and aligning the predicted objects with their true positions within the 3D space. On the other hand, the exploration of monocular 3D OD in the AV domain is a relatively new research direction. Despite all, the results obtained with the CDSM rotation approach are already promising, showcasing a considerable level of performance. There is significant potential for further enhancements and improvements in this area.

When comparing the performance metrics of the previously discussed 2D vision model on the NuScenes dataset to those on the KITTI dataset, it is evident that the model's performance is considerably inferior on

Table 7.7: 3D OD KPI metrics evaluated on NuScenes test set for a monocular camera model. Labels and predictions matching is done with the same DIST2 3D association method. The results present 3D KPIs values for each individual class and for all classes combined together. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑	mATE↓	mASE↓	mAOE↓	NDS↑
All	0.121	0.557	0.305	0.394	0.920	0.745	0.988	0.118
Car	0.445	0.648	0.420	0.510	0.827	0.557	0.315	0.439
Pedestrian	0.036	0.396	0.016	0.031	0.747	0.826	1.458	0.089
Truck	0.141	0.313	0.182	0.230	0.996	0.558	0.794	0.179
Heavy	0.074	0.303	0.112	0.164	1.419	0.839	0.377	0.167
Bicycle	0.014	0.085	0.011	0.020	0.748	0.903	2.083	0.065
Motorcycle	0.015	0.157	0.014	0.027	0.784	0.789	0.902	0.094

NuScenes. This is also the conclusion when examining Table 7.7, which reveals a similar pattern for the 3D monocular model. Only the results for the car class show meaningful predictions, while the performance of other classes is negligible. Taking a closer look at metrics beyond mAP, it is notable that the precision and recall values differ significantly. While the precision values are relatively acceptable, the recall values are close to zero. This indicates that the deficient mAP performance is primarily due to a large number of missed detections, resulting in a low recall rate. This aligns well with the earlier visual findings, where many objects were estimated at too different distances to establish an accurate association with their corresponding labels, resulting in missed detections.

Nonetheless, it is worth noting that while the camera-based model may struggle with accurately estimating object positions in the image, it excels in predicting object size and class. On the other hand, the Radar pointcloud processing network demonstrates strong position estimation but rather fails in object size and class prediction. Consequently, the fusion architecture, which combines the strengths of both sensors, presents a potential solution to compensate for their individual weaknesses and enhance overall performance. The forthcoming experiments focus on the fusion architecture, providing further insights into its capabilities.

Chapter summary:

- This chapter contains results from single-sensor model training experiments for different sensors across KITTI and NuScenes datasets. Each evaluation was done with visualization analysis as well as related KPI metrics calculation.
- Initial 2D camera image model shows remarkable performance for the car class on the two datasets both visually and with high KPIs values. Other classes perform slightly less accurately for KITTI, whereas the performance gap is clearly visible on NuScenes, introducing a class imbalance problem related to the dataset rather than the model itself.

- LiDAR-only architecture is the first one to predict objects in 3D domain. The results are exceptional, which could be attributed to versatile sensor characteristics, both in pointcloud resolution and accurate depth measurements.
- Similar pointcloud architecture trained on NuScenes Radar data performs significantly worse. With much sparser pointcloud readings from the sensor, the network struggles to predict even the cars, with close to zero performance for other classes.
- Finally, the proposed monocular camera for 3D OD with the use of CDSM alignment method is evaluated through experiments on both KITTI and NuScenes. Obtained results show satisfactory performance, given the depth estimation difficulty level for camera-only solution in 3D.

Chapter 8

CDSM fusion results

Chapter highlights:

- *Determination of the best CDSM fusion method*
- *Visualizations and KPI metrics evaluation*
- *Fusion gain comparison*
- *Corner cases analysis*

This part of the thesis describes the most crucial experiments concerning the proposed CDSM architecture in the context of fusion research. Building upon the single-sensor trials, the chapter explores and assesses additional fusion models that combine the camera with LiDAR and camera with Radar setups. These models are precisely trained and evaluated on the selected KPI metrics to measure the fusion gain factor and overall enhancements to the perception system.

This chapter starts with an evaluation of different fusion techniques, which are tested in order to determine the best approach, based on 3D KPI metrics. This selected fusion method is then used for further experiments. Another crucial aspect of this research is the performance comparison between the single-sensor and the fusion models. By examining the results obtained from these different architectures, the advantages and disadvantages of each approach are identified both visually and in terms of KPI metrics. The comparison not only highlights the strengths and weaknesses of individual models but also emphasises the benefits of fusing information from multiple sensors for improved overall system performance. Furthermore, the analysis focuses on the exploration of specific corner cases, which serve as critical instances that justify the adoption and implementation of fusion techniques. By examining these particular scenarios, the efficacy and advantages of fusion methodologies are thoroughly assessed and verified. Lastly, the evaluation extends beyond the proposed models and explores their relation to the SOTA solutions in the fusion domain. By comparing proposed models to existing cutting-edge approaches, their competitiveness can be assessed with respect to the current research state.

8.1 Fusion methods evaluation

Following the successful implementation and validation of single-sensor architectures, it became feasible to train a fusion perception model that incorporates them. This can be accomplished by employing one of the three proposed approaches to feature fusion discussed in Chapter 5: one-to-one, feature-wise, and range-based aggregation methods. To determine the most effective approach, the following experiment was conducted. The fusion network architecture was trained using camera images and Radar pointcloud data from the NuScenes dataset, with the objective of predicting objects in a 3D BEV. The individual sensor submodels were utilized to process the respective sensor input data. To ensure maximum similarity between the final models, these subnetworks were kept frozen during training, allowing optimization solely for the fusion modules. Each fusion method was implemented and appropriately integrated into the network architecture. The training process employed the same hyperparameters that were fine-tuned for the 3D OD network, along with similar 3D detection heads. Trained models were evaluated on the NuScenes test set using the 3D KPI metrics.

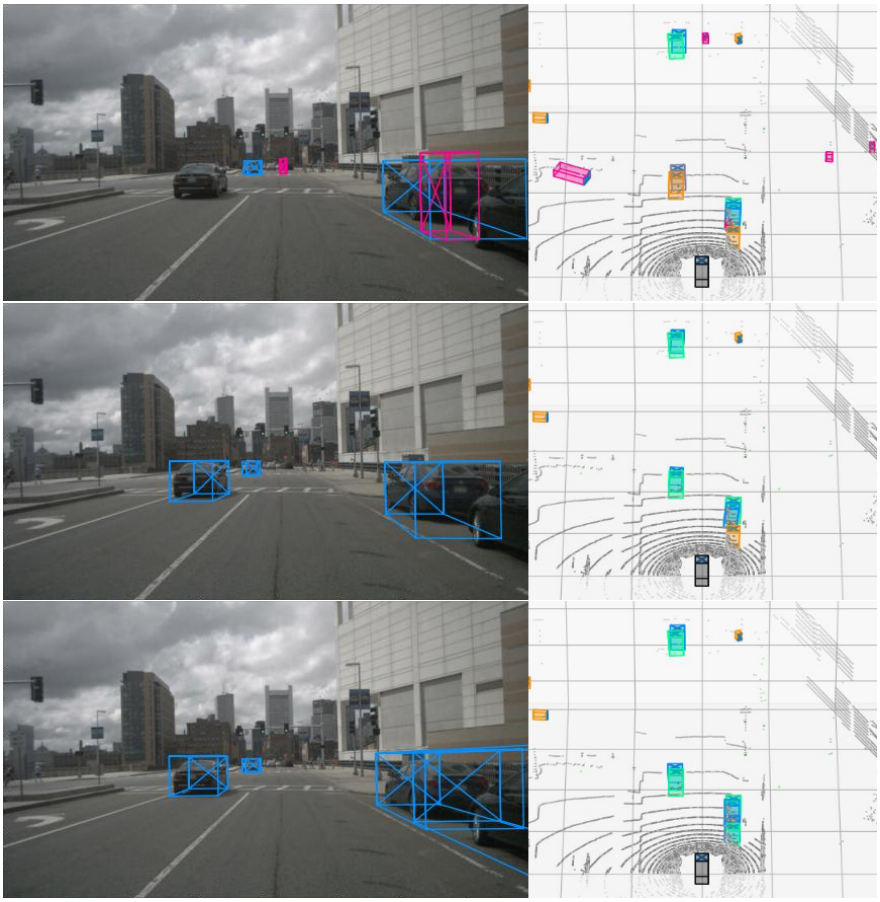

Figure 8.1: The results of different CDSM fusion methods on NuScenes test set. The exact same sample predictions are shown for proposed one-to-one, feature-wise, and range-based, respectively from top to bottom. Labels, positive detections, false detections, and missed ones are marked in green, blue, magenta and yellow respectively.

While the performance scores of the conducted experiment may not be outstanding, it is important to note that maximizing these scores was not the primary objective. Rather, the goal was to determine the most effective fusion strategy among the proposed approaches. For this purpose, additional constraints like frozen single-sensor submodels were introduced, which directly impact model efficiency. The maximization of KPIs will be done, once the most promising fusion method is selected. The results in Figure 8.1 present each method's predictions for the same data sample from the NuScenes test dataset. They exhibit similarities between the proposed methods to some extent, however, each technique has subtle differences that affect final outcome.

The first method, one-to-one fusion, is the simplest of the three. Upon visual inspection, it becomes clear that its performance is comparatively lower than the other two methods. The resulting model demonstrates acceptable true positive predictions, but it also exhibits numerous missed detections and false positive results. This outcome is expected, as the simple concatenation of two aligned feature maps without any additional domain-specific processing is employed. The method serves as a baseline for the subsequent approaches, providing insight into the fusion results achieved solely through CDSM rotation.

The following two methods incorporate aggregation and refinement steps within the fusion process. These steps aid in processing 2D image features in the new 3D domain, preparing them for fusion with pointcloud features. The distinction between these methods lies in how the aggregation is performed, and the presented figure showcases the outcomes of each approach. In both cases, the aggregation and refinement steps contribute to addressing false positives and missed detections, resulting in well-rounded predictions overall. Additionally, the range-based aggregation method also takes into account the camera sensors' FoV, which becomes particularly noticeable at very short ranges in front of the host vehicle. While the feature-wise aggregation ensures aligned image features with constant horizontal density across the entire BEV grid, the range-based approach compresses the features progressively as they get closer to the camera sensor. As a result, the range-based method spatially aligns close-range image features with the corresponding pointcloud features, leading to accurate predictions in close proximity, as exemplified by the car on the right side of the scene in the bottom illustration.

Table 8.1: 3D OD KPI metrics evaluated on NuScenes test set for different methods of CDSM fusion. Labels and predictions matching is done with the DIST2 3D association method. The results present KPIs values for the car class only, as it is the most representative one in the dataset. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Method	mAP↑	Precision↑	Recall↑	F1↑	mATE↓	mASE↓	mAOE↓	NDS↑
One-to-one	0.501	0.615	0.477	0.537	0.745	0.601	0.399	0.459
Feature-wise	0.509	0.698	0.516	0.593	0.733	0.586	0.401	0.467
Range-based	0.523	0.713	0.577	0.637	0.703	0.551	0.393	0.486

In order to assess the best-performing method in a more comprehensive manner, KPI metrics were evaluated across all test samples. The results of this evaluation are presented in Table 8.1. Confirming the initial visual examination, the one-to-one fusion approach exhibits the poorest performance in terms of KPIs.

In contrast, the other two methods demonstrate higher scores across all metrics. Both feature aggregation and refinement methods outperform the one-to-one fusion, resulting in higher mAP scores. However, the degree of improvement varies significantly between the feature-wise and range-based approaches. While both show considerable enhancements in precision scores, with an improvement of almost 10 percentage points over the baseline, the recall value also plays a crucial role in determining the overall mAP. The feature-wise aggregation achieves a recall of 0.516, compared to the one-to-one method's recall of 0.477, indicating a significant improvement. However, the range-based aggregation surpasses it even further, achieving a recall of 0.577, signifying a remarkable improvement of 10 percentage points. This notable difference can be attributed to the range-based aggregation's ability to extract close-range object detections more effectively from the vision feature maps, considering the camera's FoV.

Based on these findings, the range-based aggregation method is considered the most effective approach for fusing aligned image and pointcloud features. Consequently, it will be utilized in each fusion architecture described in the subsequent experiments.

8.2 Fusion performance

The concluding experiments in this research address the fusion architectures' model training to achieve optimal performance in terms of KPI metrics for 3D OD using combinations of different sensors. Specifically, fusion models incorporating a camera and LiDAR, as well as camera and Radar sensors, were trained on both the KITTI and NuScenes datasets. These models' architectures utilize pretrained single-sensor networks as initial components within the overall architecture to process and extract image and pointcloud feature maps. Additionally, the selected range-based CDSM fusion method is employed to merge the two sets of feature maps and generate the final fusion representation of the input data. Based on that representation, 3D prediction heads, used previously in other models yield definitive results.

All models were trained in the exact same manner, with the only difference being the input data source, which changes between different datasets and LiDAR or Radar sensors. The initial parameter set for the image processing submodel was obtained from the pretrained 2D image model weights. Similarly, the corresponding pretrained LiDAR or Radar submodel was utilized to extract pointcloud features from the input. The CDSM fusion module, BiFPN, and prediction heads were all initialized randomly using the Xavier method. The training process followed the same scheme as in the previously described experiments, including the chosen hyperparameter values and optimization techniques. The single-sensor submodels were optimized further, without freezing the pretrained parts, to better fit the fusion goal. The trained model predicts objects in the 3D BEV grid, where appropriate KPI metrics were employed for evaluation purposes.

This section primarily focuses on the individual fusion models for different sensor setups and presents the results obtained from each experiment. Detailed comparison to single-sensor solutions, fusion gain analysis, comparison to SOTA approaches, and discussions on corner cases of fusion solutions are elaborated in the subsequent sections.

8.2.1 Camera and LiDAR fusion

While the primary focus of this research is on exploring fusion solutions for perception systems, the sensor configuration involving the camera and LiDAR presents a less favourable setup. LiDARs already provide excellent detection performance independently. However, due to cost and technological readiness considerations, it is rather uncommon to include it in the sensor suite of production vehicles. As a result, the practical usefulness of fusing camera and LiDAR data for perception systems is limited. Nonetheless, for research purposes and out of sheer curiosity, a proposed model architecture has been trained to process camera images and LiDAR pointcloud data on both KITTI and NuScenes datasets. This allows for further verification of the presented approaches and provides insights into camera-LiDAR fusion capabilities in AV perception.

Figure 8.2: The results of CDSM fusion 3D OD model on KITTI test set. The visualization overlay employs a colour coding scheme that was introduced earlier, where labels are marked in green and predictions in blue.

The camera and LiDAR CDSM fusion results on KITTI data, shown in Figure 8.2, showcase precise predictions for objects at both short and long ranges. The fusion model performs exceptionally well, exhibiting minimal false positives and missed detections. By combining the strengths of both sensors, the fusion approach enhances the already reliable results from LiDAR OD, resulting in impressive overall performance.

The evaluation of KPI metrics, as shown in Table 8.2, confirms the excellent performance observed in the visualizations. Each metric surpasses the performance of the single sensor counterpart. However, when compared to the results obtained from the LiDAR-only approach, the overall enhancement is relatively modest. This can be attributed to the high resolution of the KITTI LiDAR sensor, which already provides redundant information compared to the one from the camera image. As a result, the fusion approach does not

Table 8.2: 3D OD KPI metrics evaluated on KITTI test set for a proposed CDSM fusion architecture working with camera image and LiDAR pointcloud data. Labels and predictions matching is done with the DIST2 3D association method. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑	mATE↓	mASE↓	mAOE↓	NDS↑
All	0.737	0.854	0.870	0.862	0.336	0.464	0.318	0.682
Car	0.884	0.897	0.875	0.886	0.290	0.376	0.177	0.801
Pedestrian	0.671	0.748	0.712	0.730	0.296	0.602	0.743	0.561
Truck	0.784	0.934	0.716	0.811	0.533	0.369	0.103	0.724
Bicycle	0.611	0.795	0.629	0.702	0.223	0.510	0.249	0.641

contribute substantial improvements beyond the already excellent LiDAR performance. On the other hand, the obtained metrics do not exhibit any signs of degradation when compared to single-sensor architectures. This proves that the proposed fusion functions as intended. Furthermore, despite the significant disparity in the contributions of the two sensors, the fusion process does not interfere with the performance of either. Instead, it leverages the strengths of both sensors to build upon their individual capabilities, ultimately achieving improved results.

Figure 8.3: The results of CDSM fusion 3D OD model on NuScenes test set. The visualization overlay follows the previously introduced colour coding scheme: labels, positive detections, false detections, and missed ones are marked in green, blue, magenta and yellow respectively.

The camera and LiDAR CDSM fusion model was also trained using NuScenes data. The predictions of the trained model on the test set are displayed in Figure 8.3. Overall, the results are favourable over single-sensor ones. The fusion model demonstrates decent accuracy even for distant objects, as exemplified by the bus in the middle example and the leftmost car in the bottom illustration. In such cases, where the LiDAR-only model faced challenges due to the sparser NuScenes LiDAR pointcloud format at far ranges, the fusion model manages to provide accurate predictions based on supplementary camera image features. However, when compared to the KITTI results, there are occasional instances of missed detections in estimated objects. For example, the missed detection on the far right of the bottom scene, which can be attributed to the limited number of LiDAR points capturing that object, coupled with occlusion from another car obstructing the corresponding area in the camera image. Although rare, these occurrences, where occlusion happens in both input sensor modalities, can result in lower overall performance. Nevertheless, this issue is related to the inherent challenge of limited information in such cases, which restricts the visibility and availability of essential input data required for accurate predictions.

Table 8.3: 3D OD KPI metrics of a proposed CDSM fusion architect for camera images and LiDAR pointcloud data. The KPI metrics evaluation is done on the NuScenes test set. Labels and predictions are matched with the DIST2 3D association method. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑	mATE↓	mASE↓	mAOE↓	NDS↑
All	0.298	0.733	0.570	0.641	0.573	0.592	0.890	0.306
Car	0.743	0.784	0.743	0.763	0.487	0.488	0.530	0.620
Pedestrian	0.356	0.380	0.461	0.417	0.282	0.596	1.583	0.365
Truck	0.301	0.629	0.313	0.418	0.852	0.523	0.521	0.334
Heavy	0.196	0.306	0.274	0.289	0.925	0.630	0.980	0.175
Bicycle	0.141	0.201	0.168	0.182	0.382	0.685	0.753	0.267
Motorcycle	0.190	0.451	0.291	0.353	0.320	0.722	0.834	0.282

The evaluation of the model based on 3D KPIs is presented in Table 8.3. Although the metric values are lower compared to the KITTI dataset, this pattern was also observed in the previous camera-only and LiDAR-only experiments. It can be concluded that the NuScenes dataset poses a greater challenge overall. This can be attributed to factors such as sparser pointcloud data, class imbalance, and inconsistent visibility of labels in the camera sensor, as discussed earlier. Despite the challenges, the KPI results of the fusion model surpass those of each individual single-sensor model. Similarly to the fusion model evaluation on the KITTI dataset, the performance improvement is only slight when compared to the LiDAR-only model. Nonetheless, the fusion model performs better across all metrics. Moreover, while the improvement in accuracy for the car class is slight, there is a significant enhancement in the results of previously overlooked classes. The earlier situation of high precision but low recall has been improved, as the increased recall values indicate a higher rate of true positive detections.

8.2.2 Camera and Radar fusion

The fusion architecture designed for integrating camera images and Radar pointcloud data holds significant importance within this thesis, as it offers a promising market potential by leveraging widely used sensors, that are already present in production vehicles. Moreover, the presented literature review highlights the lack of similar solutions, with only a handful of fusion methods adapted to work with such sensor suites.

The proposed model was exclusively trained and tested on the NuScenes dataset, as KITTI lacks radar data. However, in terms of KPI metrics for 3D OD tasks, both the camera and Radar solutions fell short when compared to LiDAR-only models. This can be attributed to the specific limitations of each sensor. The camera-based model faces challenges in accurately estimating depth information, while the low density of Radar pointcloud data prevents the precise detection of object attributes. On the other hand, both sensors possess unique strengths that can complement each other. By implementing a proper fusion setup, the system can leverage these strengths and mitigate their respective weaknesses. The fusion architecture allows the system to benefit from the robustness of radar in detecting object presence and the rich visual information captured by the camera. This synergy could enable the system to achieve enhanced performance by leveraging the strengths of both sensors.

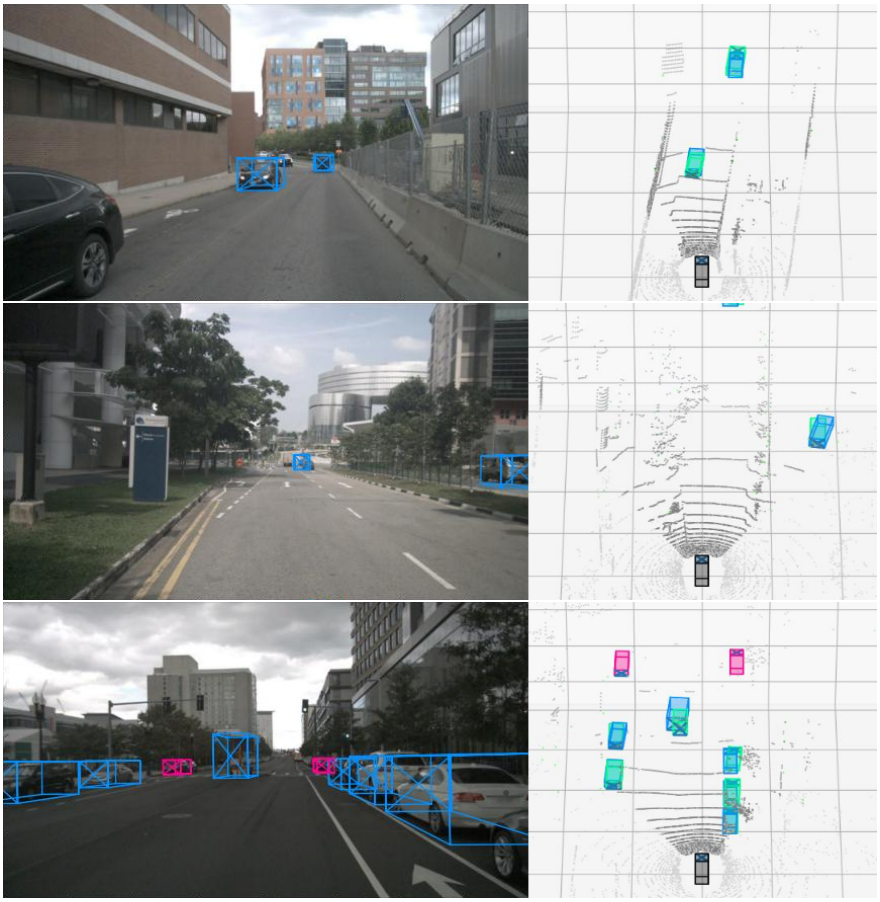

Figure 8.4: The results of the CDSM fusion model for 3D OD using camera and Radar data on the NuScenes test set. The visualization overlay follows the established colour scheme, where labels are represented in green, positive detections in blue, false detections in magenta, and missed detections in yellow.

The visualization of the trained fusion model results, presented in Figure 8.4, confirms the successful synergy achieved between the camera and Radar sensors. The predictions of the fusion model surpass those of each individual single-sensor solution. The fusion demonstrates remarkably high accuracy, even at considerable distances, which is visible in the first two scenes. This remarkable level of accuracy surpasses that of any previous model. Such performance is made possible through the combination of a long-range radar sensor, which provides robust detection capabilities at a distance, and the object features extracted from the camera. The model also exhibits exceptional precision, resulting in minimal missed object detections, even in densely populated scenes like the last example. Moreover, the only instances of false positive detections, observed in the third scene, are valid detections of actual cars. These false positive indicators only exist due to the lack of corresponding labels, which can be attributed to either visibility filtering or missed annotations in the NuScenes dataset. Ultimately, the results align with expectations, as the fusion model effectively leverages the strengths of each sensor. The fusion approach combines the best attributes of both the camera and radar sensors, leading to highly accurate and reliable object detection results.

Table 8.4: 3D OD results of the proposed CDSM fusion architecture utilising camera images and Radar pointcloud data. The KPI metrics calculation was performed on the NuScenes test set. The labels and predictions are matched using the DIST2 3D association method. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Class	mAP↑	Precision↑	Recall↑	F1↑	mATE↓	mASE↓	mAOE↓	NDS↑
All	0.208	0.653	0.572	0.610	0.806	0.660	0.913	0.207
Car	0.681	0.707	0.711	0.709	0.623	0.521	0.390	0.584
Pedestrian	0.178	0.289	0.276	0.282	0.492	0.776	0.997	0.211
Truck	0.241	0.449	0.322	0.375	1.056	0.621	0.720	0.230
Heavy	0.055	0.133	0.147	0.140	1.218	0.695	1.704	0.078
Bicycle	0.0	0.0	0.0	0.0	-	-	-	0.0
Motorcycle	0.091	0.238	0.250	0.243	0.642	0.686	0.755	0.198

The evaluation of KPI metrics for the camera and Radar fusion model is presented in Table 8.4. While the overall metrics for all classes exhibit a mixed performance, there is a noticeable improvement compared to the results obtained from single-sensor models. Despite the persisting issue of class imbalance in the NuScenes dataset, the fusion model excels in predicting the car class, surpassing the performance of both camera-only and Radar-only models. The accuracy achieved in 3D OD for the car class is comparable to that of LiDAR sensors, showcasing the effectiveness of the fusion approach in leveraging the strengths of both sensors. However, it is important to note that the performance of other classes still falls behind. This discrepancy can be attributed to the fact that both single-sensor submodels, used for preprocessing each input data type, primarily excel in detecting cars. On the other hand, while this problem occurs on NuScenes datasets for camera and LiDAR setup as well, that exact same model trained on KITTI does not suffer from such conditions. However, since the KITTI dataset does not provide Radar pointcloud data, it is not possible to verify this hypothesis and attribute the issue solely to the NuScenes dataset.

Nevertheless, the CDSM fusion method proves to be a valuable approach to integrating information from multiple sources. By merging data from different sensors and perspectives, the fusion model achieves improved performance compared to individual single-sensor solutions. Additionally, this fusion leverages widely used sensors that are already present in production cars, offering a practical and cost-effective alternative to integrating additional LiDAR sensors. The lightweight nature of the CDSM fusion approach also minimizes the additional computational resources required. Considering these advantages, the next section will provide a detailed comparison between the single-sensor architectures and the fusion architecture. This comparison will extend beyond KPI metrics to include visual improvements, evaluation compared to SOTA techniques, and an in-depth analysis of fusion in corner cases. This comprehensive examination will further demonstrate the benefits and potential of the fusion approach.

8.3 Fusion gain

Upon training and evaluating all the proposed single-sensor architectures, alongside the fusion models combining different sensor setups separately, a thorough comparison can be conducted. This comparison aims to evaluate the performance of each experiment's results in relation to one another and determine the fusion gain achieved by the fusion solutions. Although the results on the KITTI dataset are very promising, the lack of radar pointcloud data within the dataset poses a significant disadvantage. This limitation restricts the possible input sensor configurations to just camera-only and LiDAR-only single-sensor models, along with only one fusion architecture that combines both camera images and LiDAR pointcloud data. To address this drawback and provide a more general overview of the advantages and disadvantages of each sensor's contribution to the AV perception system, the models' comparison is done on the NuScenes dataset. This dataset is created with a comprehensive sensor suite, allowing for training and comparing every single-sensor modality, including camera, LiDAR, and Radar. Furthermore, with this dataset, additional fusion setups can be explored, namely camera with LiDAR and camera with Radar configurations, offering a broader understanding of the various sensor combinations' performance.

The summary of evaluation metrics on the NuScenes dataset for all mentioned experiments is presented in Table 8.5. It is important to note that for a consistent comparison between each method, the most representative car class results were used. While other classes exhibit mixed performance, varying for each approach, they overall pose a challenge in all experiments conducted on the NuScenes data. Addressing class imbalance and label visibility problems are crucial aspects for a production-ready solution. However, for research purposes and fusion gain evaluation, the car class results are satisfactory and can already provide comprehensive insights. Moreover, it's worth noting that the aforementioned problems are not connected to the approach, as the same architectures trained on the KITTI data do not exhibit the same behaviour across different classes. Therefore, the presented conclusions should generalize to all classes once those unrelated issues are addressed.

Table 8.5: The comparison of the KPI metrics for all single-sensor and fusion models trained on the NuScenes dataset. The results present the performance on car class only. Different solutions use different sensor setups, where C stands for the camera, L for the LiDAR, and R for the Radar. Each model has a prediction domain indicator, as well as the association method used for matching the predictions and labels for all KPI calculations. ↑ and ↓ indicate whether higher or lower metrics' values respectively are more desirable.

Method	Sensor	Domain	Association	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓
Vision model	C	2D	IOU20	0.741	-	-	-	-
Vision model	C			0.445	0.439	0.827	0.557	0.315
Pointcloud model	L	3D	DIST2	0.733	0.608	0.524	0.492	0.556
Pointcloud model	R			0.324	0.358	0.811	0.613	0.395
CDSM Fusion	C+L			0.743	0.620	0.487	0.488	0.530
CDSM Fusion	C+R	3D	DIST2	0.523	0.486	0.703	0.551	0.393
CDSM Fusion (FT)	C+R			0.681	0.584	0.623	0.521	0.390

The table presents the KPI metrics evaluation for the single-sensor camera, LiDAR, and Radar models. The results primarily focus on the 3D domain, except for the initial experiments with 2D object detection in the images. While the 2D performance is quite high, it lacks 3D-specific KPIs and is not directly comparable to other results due to domain differences. Among the 3D single-sensor results, the LiDAR-only architecture achieves the best performance, which is expected since the LiDAR sensor excels in providing dense and information-rich pointcloud data, as well as accurate range detection readings. The camera and LiDAR fusion architecture shows minor improvements over the pure LiDAR solution, with all metrics slightly higher. This indicates that visual information from the camera affects final results by a minor margin, hence the benefits of the fusion in the given setup are insignificant.

Both the camera and Radar single-sensor networks demonstrate decent performance in the 3D OD task. However, when compared to the LiDAR-only approach, there is a significant performance gap. This observation opens up the possibility for a camera and Radar fusion architecture to build upon the single-sensor methods and provide better overall performance. By combining rich visual camera image features with accurate Radar depth detections, the fusion model achieves a remarkable mAP of 0.523, an improvement from the camera and Radar solutions which scored 0.445 and 0.324, respectively. A closer examination of the detailed metrics reveals that the significant improvement can be attributed to a much lower mATE, resulting in more accurate detections and a higher number of true positive associations within the predicted objects. Furthermore, when the subsequent sensors' feature extraction submodels were optimized together with the fusion architecture, rather than being frozen from initial experiments, the fine-tuned model, denoted in the table as "FT", demonstrated even better performance. All the metrics show improvements, with the mAP reaching an impressive 0.681, closer to the LiDAR performance rather than the camera or Radar ones. Based on this analysis and the presented results, the conclusion of the metrics evaluation is that the camera and Radar fusion achieved a major improvement over both single-sensor methods. This confirms that the CDSM fusion approach can be successfully utilized for LLF, significantly enhancing 3D OD performance.

Figure 8.5: The comparison of the results for the same scene in the NuScenes test dataset, showcasing the outputs of the camera-only, radar-only, and CDSM fusion models from top to bottom. Both the images and corresponding 3D views are provided, along with a visualization overlay that follows the established colour scheme. In this colour scheme, labels are shown in green, positive detections in blue, false detections in magenta, and missed detections in yellow. The BEV also includes a LiDAR point cloud for reference. The presented image highlights the performance gain achieved through the fusion approach when compared to the single-sensor models.

The visual results shown, in Figure 8.5, illustrate the comparison of the camera and Radar single-sensor predictions with the CDSM fusion architecture predictions. Upon closer examination, the conclusions drawn from the KPI analysis become apparent. In the top camera-only prediction, the bounding box position and size are relatively proper, indicating that the camera sensor is effective at detecting objects in the scene. However, the 3D depth estimation is slightly mispredicted, leading to failed associations and resulting in false positive detections. This limitation is expected, as camera sensors may struggle with accurately estimating the depth of objects in the scene, especially at longer distances, which was the conclusion from the camera-only results interpretation. Subsequently, in the middle Radar-only example, the 3D position estimation is very accurate, resulting in a true positive detection of the car in front of the host vehicle. The Radar sensor's ability to provide accurate depth readings proves advantageous in this scenario. However, cars on the left and right sides are predicted with hugely mismatched bounding box sizes, proving that the Radar sensor alone may struggle with accurately determining the size of objects in the scene. Finally, in the

bottom image, the CDSM camera and Radar fusion architecture leverages readings from both sensors. By combining Radar depth readings with accurate object size features from the camera, the fusion predictions match the targets much more accurately. This combination of information from both sensors results in improved overall performance, which is reflected in the KPIs. The fusion model addresses the limitations of each individual sensor and utilizes their strengths to achieve more reliable and precise detections, leading to a significant improvement in the final predictions.

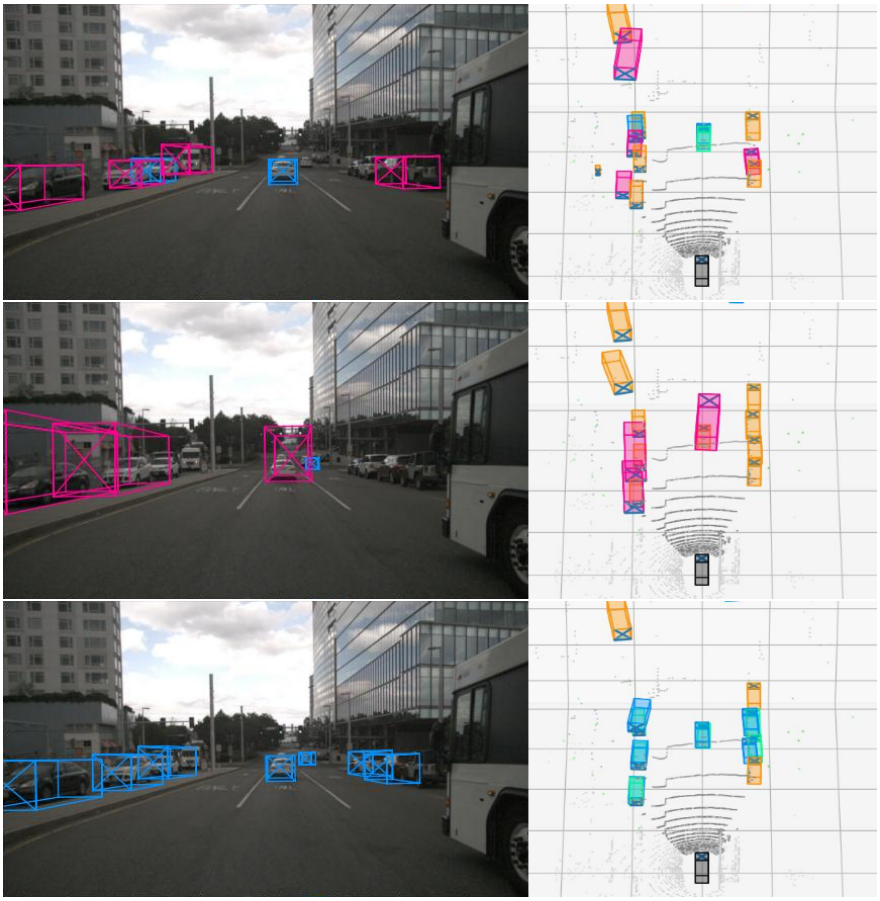

Figure 8.6: The comparison of another scene from the NuScenes test dataset, with predictions from the camera-only, radar-only, and CDSM fusion models respectively from top to bottom. The visualization overlay follows the described colour scheme: labels are shown in green, positive detections in blue, false detections in magenta, and missed detections in yellow. The image demonstrates the detection accuracy gain of the fusion approach with respect to both single-sensor models.

The previous example demonstrated how the fusion of Radar detections improved poorly estimated depth in object predictions. In contrast, the scenes presented in Figure 8.6 highlight how camera features enrich the Radar sensor readings in the fusion outcome. In the top camera-only predictions, while still lacking precision in objects' positions, the processing of image features provides a clear division into separate cars. The camera sensor excels in identifying distinct objects, even in cluttered scenes such as the presented one. Conversely, the Radar detections in the middle column are relatively sparse, making it challenging to distinguish clear boundaries between one large object and two smaller ones. This difficulty is noticeable in the column of parked cars on the left-hand side of the scene. Multiple Radar readings reflected from cars are

used as object indicators, but they are grouped together, resulting in large bounding boxes corresponding to truck or bus objects, which is an inaccurate prediction. In the fusion architecture results, both sensors' features are combined, allowing the model to properly divide Radar pointcloud readings among corresponding objects identified by image features. This fusion enables the model to improve final predictions significantly. Although there are still some missed targets, the objects in this overall difficult scene are now predicted with much greater quality compared to single-sensor results.

8.4 Corner cases

In the preceding section, the benefits of the CDSM fusion architecture were discussed in the context of enhancing one sensor's results using information extracted from the other source. This was demonstrated through examples of improving camera object position accuracy with accurate Radar depth detection and also by associating Radar detections to specific objects based on a visual camera features representation. However, there are instances where one sensor may fail to detect an object entirely. In such corner cases, the fusion also proves to be valuable, serving as an additional safety layer in the perception system.

Figure 8.7: The example of a corner case scenario where the single sensor models for camera and Radar in the top and middle images, respectively, fail to produce accurate results. In contrast, the fusion model in the bottom image successfully combines the information from both sources and provides improved detection performance. The figure includes both the images and corresponding 3D views, accompanied by a visualization overlay that follows the established colour scheme.

The scene pictured in Figure 8.7 illustrates a corner case where the Radar-only system struggles to detect parked cars on the right side. These objects are labelled and classified as missed detections, which implies that there are Radar pointcloud detections within the objects' bounding boxes. Ideally, the model should detect them, but it fails to do so. Conversely, the top camera-only model successfully detects all three cars, though with slightly poor accuracy for the car in front of the host vehicle, leading to false positive detection. Nevertheless, the parked cars are properly detected. The fusion model, which combines inputs from both sensors, is capable of correcting the predictions made by the single-sensor models. The car in front is now detected with sufficient accuracy, resulting in a true positive. More importantly, the fusion model is able to detect parked cars, which were previously missed by the Radar-only model, with even higher accuracy.

Figure 8.8: The illustration of another corner case scenario. The first two scenes from the top show camera and Radar single sensor models results, which encounter difficulties in accurately detecting any object at all. However, the fusion model depicted in the bottom image effectively integrates the information from both sensors, leading to improved detection performance. Notably, the fusion model successfully identifies objects that were not detected by either of the individual single-sensor solutions.

Another example of a corner case where CDSM fusion model outperforms single-sensors solutions is presented in Figure 8.8. Several interesting observations can be made from this scenario. Firstly, both the camera-based and Radar-based models miss objects in their respective predictions. The camera-based model fails to detect the car in the far distance, which aligns with the previously discussed accuracy problem for long-range predictions of that model. Likewise, the Radar-based solution misses the car on the right, likely

due to the occlusion from a street lamp pole. The Radar signal is well reflected by the pole, restricting the detection of the car behind it. In contrast, the fusion model once again demonstrates superior performance and the ability to merge and refine information from both sensors. In the situation where there are no positive detections in either of the single-sensor models, the fusion model provides correct predictions for all objects with enhanced accuracy. It even detects a pedestrian, which was not visible in any previous predictions. This case can be explained by a prediction confidence threshold, which is not met by either of the single-sensor networks. However, when fusing information from both sensors, confidence is accumulated from both sources, satisfying the threshold condition and resulting in a positive detection outcome.

8.5 SOTA comparison

The final fusion method evaluation includes a comparison with SOTA results in the 3D OD domain. The evaluation, shown in Table 8.6, is based on the car class model predictions using the NuScenes dataset. While the main focus and contribution of this thesis revolve around the camera and Radar fusion approach, we also include additional SOTA solutions for camera-only and LiDAR-only models, as well as the CDSM camera and LiDAR fusion results for a comprehensive comparison. The key distinction in these results is the association method utilized for calculating the matching pairs of predictions and labels, which is essential for evaluating various metrics. Following the official NuScenes leader board (*NuScenes ranking* 2020), a distance-based association approach is adopted, where different threshold ranges of 0.5m, 1m, 2m, and 4m are applied. This results in four different mAP values, which are then aggregated by summing them up and computing their mean to determine the final mAP score. This approach ensures an assessment of the model's performance across different distance ranges and is widely used to compare the results on the NuScenes dataset.

Table 8.6: The results of KPI comparison between proposed CDSM fusion approach and various SOTA solutions using selected 3D performance metrics. The modalities considered in the comparison include camera (C), LiDAR (L), and Radar (R) sensors. The detection domains are either 2D image space or a 3D-enhanced BEV grid. The evaluation employed the official NuScenes association method to ensure consistency and fairness in the comparisons. ↑ indicates the higher metric value is more desirable.

Method	Sensor	Domain	Association	mAP_{car} ↑
FCOS3D	C		Average mAP	0.524
PointPillars	L	3D	over DIST	0.684
CDSM Fusion	C+L		0.5,1,2 and 4	0.695
CRFNet	C+R	2D	IOU20	0.559
CenterFusion	C+R		Average mAP	0.509
FUTR3D	C+R	3D	over DIST	0.530*
CDSM Fusion	C+R		0.5,1,2 and 4	0.535

**FUTR3D paper provides only general mAP for C+R. Based on the comparison of C+L (single-beam) general and car performance, an estimation of C+R car class mAP has been made to be somewhere between 0.52-0.54.*

Among the camera and Radar fusion solutions, the CRFNet approach reports the best mAP metric. However, it is important to note that CRFNet methodology involves early-level fusion, where 2D images are enhanced with mapped and encoded Radar detections to predict 2D bounding boxes in the image space. As a result, this task is considerably simpler than full 3D OD, making direct comparisons inappropriate. For full 3D OD with a camera and Radar fusion approach, the most related SOTA solutions are CenterFusion and FUTR3D. The proposed CDSM fusion method demonstrates a very similar mAP value to FUTR3D, the newer of the two, and both outperform CenterFusion by a small margin. These superior results suggest that the CDSM fusion method is effective and capable of delivering high-quality results. Moreover, it introduces a novel and innovative approach to the fusion of camera and pointcloud feature maps, which can be applied to any given single-sensor architecture, provided it produces intermediate feature maps at different levels. The CDSM fusion concept offers a final advantage in that it is highly lightweight and requires minimal computational overhead compared to both single-sensor preprocessing submodels.

Chapter summary:

- In this chapter the main innovation proposed in the thesis, the CDSM fusion method, was evaluated and tested. At first, the three proposed fusion techniques (one-to-one, feature-wise and range-based) were compared and the best-performing range-base method was selected for further experiments.
- Next, camera and LiDAR data fusion model was trained. The presented results prove the fusion solution of that modalities works as intended, however, considering the LiDAR-only architecture performance, the KPI metrics are improved only slightly.
- On the other hand, camera and Radar fusion improves on both related single-sensor models by a large margin. Especially for the car class, obtained results show that when each sensor struggles with certain scenarios, the complementary fusion solution can provide better results.
- Further analysis was performed on camera and Radar fusion, with a detailed comparison to both corresponding single-sensor results. Specific scenes were investigated highlighting the fusion gain. Similarly, a more comprehensive comparison was done for the KPI metrics, generalizing presented findings across the whole dataset.
- The corner cases were presented, where single-sensor models failed to accurately predict selected objects. Those examples showcased the robustness of the fusion solution and the impact it could have on perception systems.
- Lastly, summary of CDSM fusion results in relation to current SOTA solutions was presented, showing similar KPI metrics scores achieved. Especially promising results were obtained for camera and Radar configuration.

Chapter 9

Explainable AI analysis

Chapter highlights:

- *Grad-CAM adaptation to multi-class and multi-scale models*
- *Grad-CAM adaptation for pointcloud-based models*

In the previous chapter, the results and corresponding KPIs provided essential quantitative measures for evaluating model performance. However, as those models present novelty and technological advancements, the need for XAI methods to ensure their robustness and safety arises, as discussed in Chapter 4. Understanding the inner workings of a blackbox Neural Network is not only essential for transparency and interpretability but also plays a crucial role in identifying potential issues and improving the overall trustworthiness of the system. This chapter goes beyond the KPI evaluation and focuses on the adaptation of XAI methods, particularly Grad-CAM, to the proposed model architectures. By exploring the application of such techniques, it showcases the potential to enhance model understanding and offer insights into the decision-making processes. The presented work is an extension of the earlier research documented in the article (Dworak and Baranowski 2022), where the author of this thesis played a significant role as the main contributor. Based on this foundation, two adaptations of the original Grad-CAM method are discussed.

The first part of this chapter focuses on extending the Grad-CAM method to accommodate the complexity of multi-class and multi-scale network architectures, employed in the previous single-sensor models and fusion solutions. A specific example showcases the application of this method during the design process of the CDSM fusion method. By leveraging the Grad-CAM results, a range-based aggregation was developed, demonstrating the practical utility of the adapted technique. In the second part, a novel adaptation of the Grad-CAM method is presented, enabling effective visualization and explanation of pointcloud architectures. This is a completely new domain for applying Grad-CAM, however, given the presented fusion process, such XAI is necessary for pointcloud-based approaches as well. The research addresses unique challenges in this domain and focuses on improving the quality of generated visualizations. By exploring both of those adaptations of the Grad-CAM technique, this research aims to provide a deeper understanding of the inner workings of complex Neural Networks, such as proposed CDSM fusion architectures.

9.1 Multi-scale Grad-CAM

In order to apply the Grad-CAM to the Neural Network architectures proposed in Chapter 5, certain modifications are necessary to ensure proper functionality. As explained in the evaluation methods section, the Grad-CAM approach consists of two main elements: target layer activations and weights that determine their significance in the final Class Activation Maps. While the activations are obtained in a similar feed-forward manner regardless of the network design, adapting the Grad-CAM method to the proposed models poses a challenge related to the grid output format of the network. The output of the network plays a crucial role in calculating activation weights via the backpropagation of gradients to a target activation layer. Given the model's purpose and architecture design, specifically linked to OD in the form of a SSD, the prediction format becomes significantly more complex than in a simple classification problem, as depicted in Figure 9.1. This complexity introduces unique considerations that must be addressed for the Grad-CAM technique to yield meaningful results in the context of the proposed models.

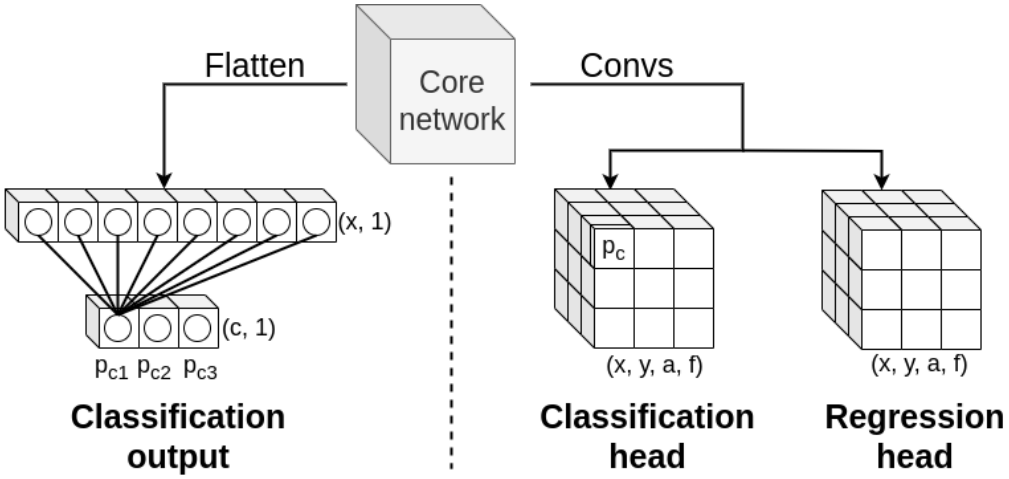

Figure 9.1: A comparison between different types of model outputs in a context of the Grad-CAM method. In the case of the classification network, the output is a vector of class probabilities, which can be directly used as a class score. On the other hand, the grid tensor output is a more complex structure with multiple dimensions (x, y, a, f) for each prediction head. Within this structure, class probability values are embedded into the feature vector of the classification head.

The primary objective of processing the output in the context of gradient-based XAI is to obtain a class score value, denoted as S_c, which serves as the starting point for the backpropagation algorithm. In the original Grad-CAM approach for a classification problem, the model's output is a vector containing probability predictions $p(c_i)$ for each class. The class score value for a specific class c_i can be extracted as $S_{c_i} = p(c_i)$. In contrast, the SSD OD model produces a multi-dimensional tensor as its output. The first two dimensions of this tensor correspond to a 2D grid of cells, dividing the entire RoI. Optionally, a third dimension represents anchor boxes of different sizes if they are utilized in the model. In the last dimension, estimated features for specific grid positions and anchors are predicted as a vector.

Unlike the approach described in the referenced research paper, where separate class heads were used to predict both confidence and object features, the architectures proposed in this thesis employ shared classification and regression heads across all classes. In this setup, the class probability value, relevant to the class score, is obtained solely from the classification head, while the regression head could be entirely disregarded. The single prediction vector from the classification head contains classification probabilities for each class. Taking into account this output format, the proposed calculation of class score value for presented models is given by the following formula:

$$S_{c_i} = \sum_{w_i=1}^{w} \sum_{h_i=1}^{h} \sum_{a_i=1}^{a} s_{c_i}(y_{pred}(w_i, h_i, a_i, c_i)) \quad \text{where} \quad s_{c_i}(p(c_i)) = \begin{cases} 0 & p(c_i) \leq th \\ p(c_i) & p(c_i) > th \end{cases} \tag{9.1}$$

where the final class score S_{c_i} is determined by summing up the selected c_i class probability predictions $p(c_i)$ for each cell in the output grid along the first two spatial grid dimensions x, y, and optionally the anchor dimension a of model's classification head prediction y_{pred}. To ensure higher confidence predictions and filter out unwanted noise, the predicted probabilities are further refined by applying a threshold $th = 0.1$. This threshold allows only the more confident predictions to contribute to the final class score.

The class score value serves as the starting point for the backpropagation algorithm, which computes the gradients for a selected convolutional layer with respect to the score. In the Grad-CAM technique, these gradients form the basis for obtaining weights of activations at the given layer. Various verified methods exist for calculating these weights from the gradient values. For the adaptation research purpose, both the original Grad-CAM and Grad-CAM++ formulas were tested, and both yielded satisfactory results. Considering the simplicity and sufficiency of the original method, a straightforward approach of calculating the global mean of gradient values is employed as the weight for each layer activation. The final CAMs for the proposed model architectures are obtained by multiplying the activations with their corresponding weights.

The presented solution addresses one of the challenges related to the high-dimensional tensor format in the output. Another obstacle arises from the use of BiFPN as a features refinement block in all proposed models, leading to multiple output tensors, one for each scale. Creating a single CAM for a multi-scale model proves to be challenging since the output, activation, and gradient tensors differ in size for each scale as well. To address this issue, the suggested approach is to utilize the Grad-CAM method independently for each scale, as discussed earlier. This strategy of generating independent visualizations for each scale can offer more informative insights than a single merged heatmap. Furthermore, rescaling and merging CAMs is more feasible in an image pixels domain. Therefore, when a single map is required for a multi-scale model, the approach involves creating CAMs for each scale and then merging them as image masks. This method allows for a more manageable process of obtaining a comprehensive CAM for the entire multi-scale model.

Throughout the research, the proposed output-altering adaptations of Grad-CAM were examined on a camera-only multi-scale model. Figure 9.2 showcases the generated CAMs for three different scales of the same scene model's predictions. In the presented figure, an intermediate correlation between an object's

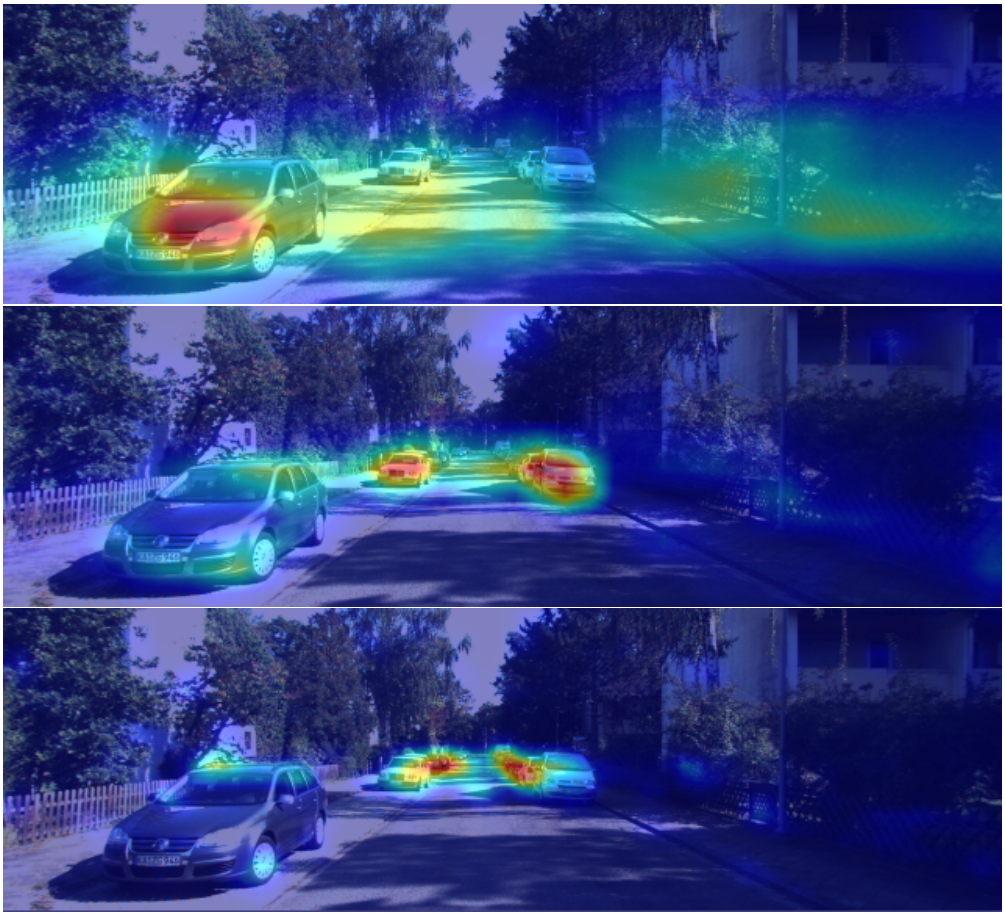

Figure 9.2: Visualization of the proposed Grad-CAM method applied to a vision-only OD network featuring a multi-scale output grid. The illustrated CAM visualizations for the car class are shown in three scales: large, medium, and small, respectively from top to bottom. These visualizations reveal the varying distances of detected objects from the camera sensor at different scales.

position relative to the sensor and the scale in which that object is predicted can be observed. Further investigation reveals that this correlation is due to the anchor sizes assigned to each label based on the IoU score, which, in turn, depends on the particular grid cell size. Smaller anchors are generated on fine-grained grids of smaller scales, while larger anchors are generated on larger scales. Consequently, the model learns to predict small objects on the corresponding small scale, as it is optimized during the training process to match predicted targets to those labels.

Expanding upon this discovery, an exemplary instance of how the Grad-CAM method investigation aids in model comprehension can be demonstrated. Through the combined information of CAMs and known label positions, the detection ranges for each output scale can be deduced for a camera-only model. This insight led to one of the most significant contributions of this thesis, the proposal of the range-based CDSM fusion method, described in Chapter 5. Remarkably, the range-based CDSM fusion method emerged as the most effective among the three proposed solutions for integrating camera image features with pointcloud data. This not only validates the intended functionality of the proposed solution but also serves as a firsthand demonstration of the benefits of employing XAI methods during model analysis.

9.2 Pointcloud Grad-CAM

Another area where Grad-CAM adaptation is certainly required is the Class Activation Maps visualization of the models that process pointcloud data. As the method proves to be a very viable analysis tool, the proposed LiDAR-only and Radar-only models as well as the fusion solutions could benefit significantly from such adaptation. The previously discussed output processing, which in that case also includes multi-dimensional tensors and multiple scales of the output, can be leveraged in a similar manner to obtain respective CAM overlays. However, pointcloud processing with a Neural Network approach requires architectural solutions distinctive to this type of data. Specifically, the initial 3D pointcloud detections undergo a VFE process, which transforms the data processing domain into BEV perspective. These changes greatly impact the application of the Grad-CAM method, as the target layer and resulting CAMs are now generated within the BEV perspective. This poses a challenge when attempting to visualize and combine 3D pointcloud input and BEV CAMs into one coherent representation.

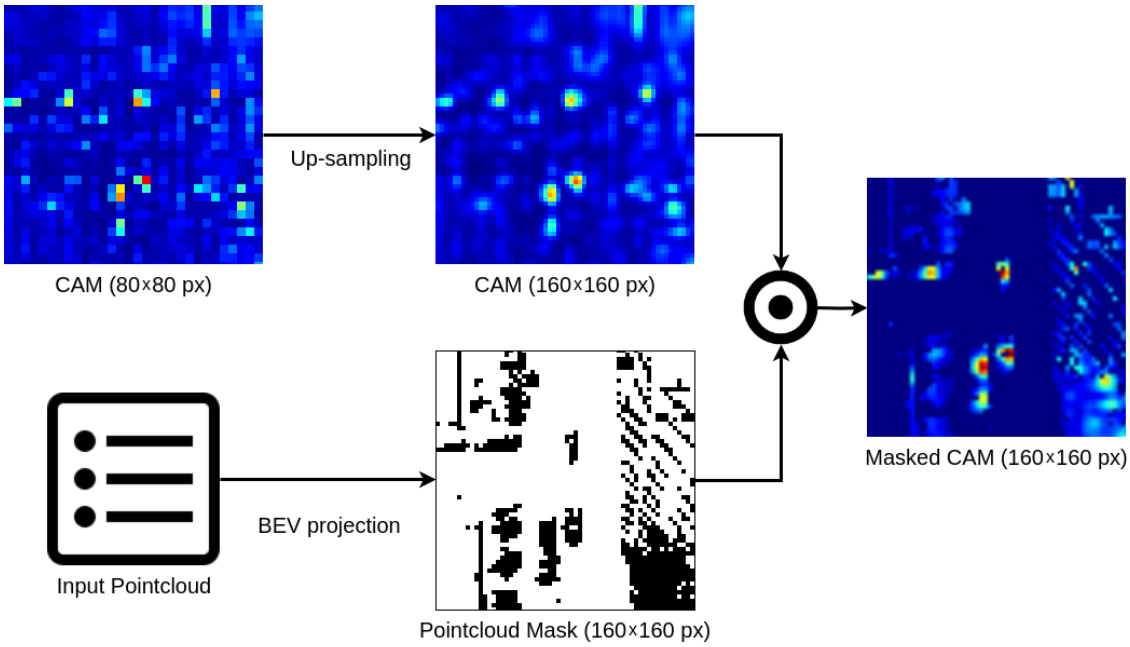

Figure 9.3: An example of generated CAMs and input pointcloud data combination. The process involves projecting detections onto a BEV grid and then multiplying the resulting pointcloud mask with an up-sampled CAM. This operation yields a high-resolution heatmap where related CAM values are concentrated only near the actual inputs in the BEV perspective.

To mitigate the effect of a mismatch between 3D pointcloud data and 2D CAMs, a fused visualization method is proposed, as illustrated in Figure 9.3. This method involves creating an input pointcloud mask by projecting each point onto a BEV with a significantly higher resolution than that of the CAM. This possibility arises from the fact that CAMs are limited to the size of the target Neural Network layer. In contrast, casting pointcloud detections can be performed with arbitrary precision, ranging down to the sensor resolution. Simultaneously, the CAMs undergo up-sampling using cubic interpolation to match the higher resolution of the projected pointcloud. Finally, element-wise multiplication of the mask and the CAM

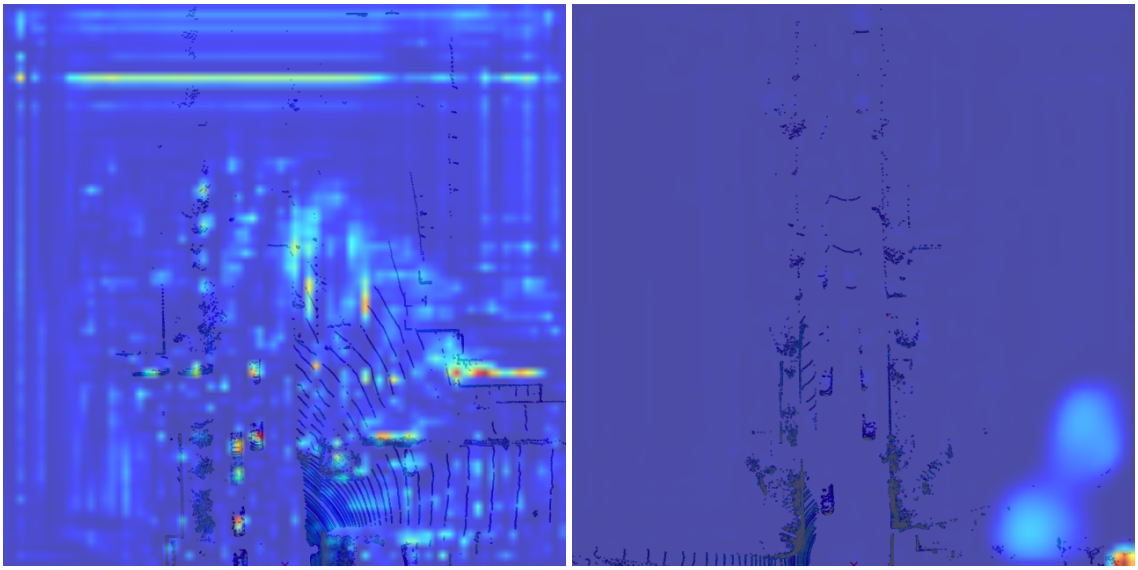

Figure 9.4: Initial Grad-CAM adaptation results for a LiDAR pointcloud data using proposed class score calculation for a grid tensor output and masked BEV visualization. Half-transparent CAM is presented in a figure together with a dark blue pointcloud BEV projection for reference. The colour coding ranges from blue to red, where warmer colour indicates a higher influence on an input point on the final predictions. While the car objects are appropriately highlighted, significant noise is observed in the right image. Moreover, the level of details remains low when compared to casted LiDAR pointcloud detections due to the low resolution of CAM. The left image showcases the example of CAM dominated by activation layer noise in the bottom right corner.

is calculated, resulting in high-resolution BEV heatmaps with more detailed information regarding actual pointcloud readings. The size of the mask influences the enhancement outcome, thus further experiments explore various mask resolutions used. For the initial experiments, however, no mask is being used, to present the results that resemble the original method as closely as possible.

In the initial experiment, the Grad-CAM method is applied to a LiDAR pointcloud processing network with minimal modifications. The activations are obtained during feed-forward inference, while the weights are calculated using the global mean average formula on gradients from the back-propagation algorithm, based on the accumulated class scores value. The consequent CAM is then merged with the pointcloud data for visualization purposes. However, it's important to note that during this experiment, the CAM generation does not involve masking, as these outcomes form a baseline for any further improvements. The first results for pointcloud Grad-CAM are presented in Figure 9.4. The generated visualization properly highlights objects in the scene with red-coloured areas around the cars in the related pointcloud data, indicating that those input points are indeed used for determining objects' features. Nevertheless, the visualization lacks detailed accuracy near the objects compared to the sensor pointcloud resolution. Additionally, it is burdened with significant noise, which adversely affects the overall quality of the results.

In the next experiments, the pointcloud mask described earlier is applied to enhance the level of detail in the CAM visualization, addressing the low-resolution problem. Both the chosen activation layer and the generated original CAM have a resolution of 80×80, while the pointcloud BEV projection can be performed

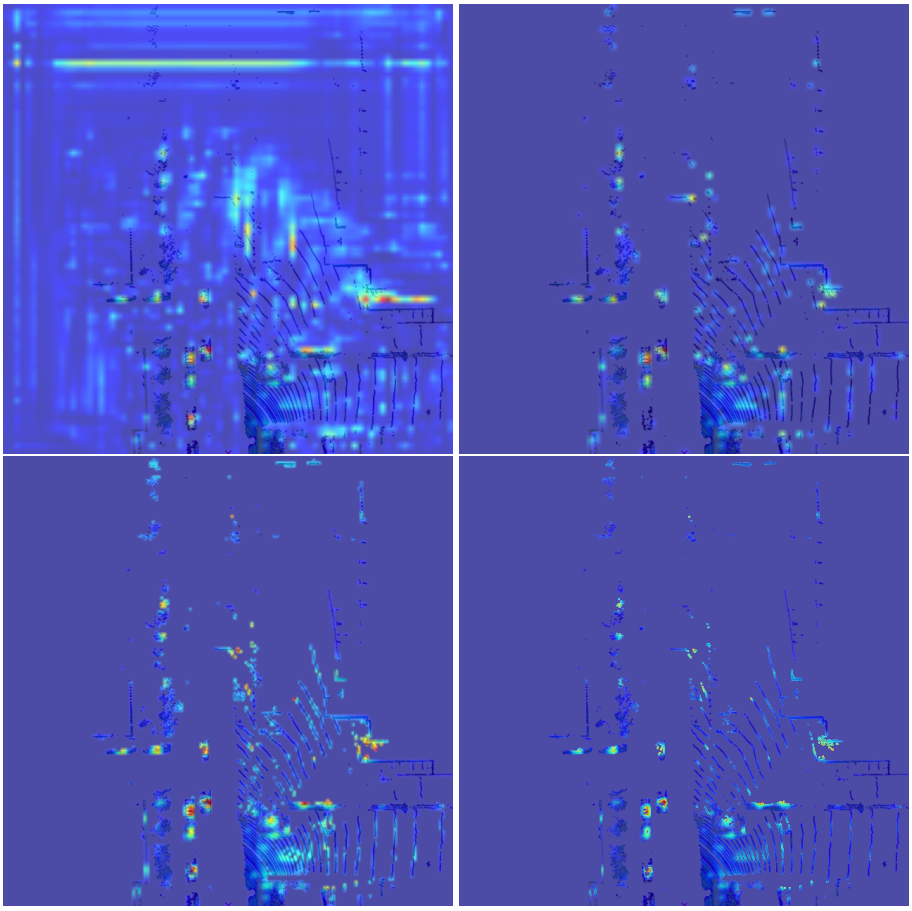

Figure 9.5: Comparison of different input pointcloud mask resolutions used for CAM filtration in BEV. The illustration shows the case with no mask, 80×80, 160×160, and 640×640 mask sizes in pixels, respectively, from left to right and top to bottom. Without a mask, the generated CAM is noisy and imprecise. The application of the mask filters out the noise and enhances the level of detail near the input pointcloud data. However, at higher mask resolutions, CAMs become too grained, rendering them unsuitable for interpretation.

at any given grid size. Additionally, the resolution of the supplementary pointcloud overlay presented in the results is 640×640, where each pixel corresponds to a 12.5cm by 12.5cm area in VCS. Subsequently, various experiments involve using pointcloud binary masks of different sizes, ranging from 80×80 to 640×640 pixels, as shown in Figure 9.5. The main observation regarding pointcloud masks is that at higher resolutions, CAMs become excessively filtered to the detections, leading to a loss of object instance integrity and an unclear visualization. Conversely, using a resolution of 80×80 does not yield any significant improvement in precision. As a result, a mask resolution of 160×160 is chosen as a compromise between readability and level of detail. Applying such a mask also provides other advantages, which may not be immediately apparent. In the CAM domain, before visualization, the values from each weighted activation are relatively small due to gradients' weights, forcing a normalization process to visualize them as an image format. However, by applying the mask before normalization, certain parts of the activation are excluded, resulting in a wider RGB spectrum range for relevant CAM values. This effect is observable in the visualization backgrounds of different masks and near highly confident detections.

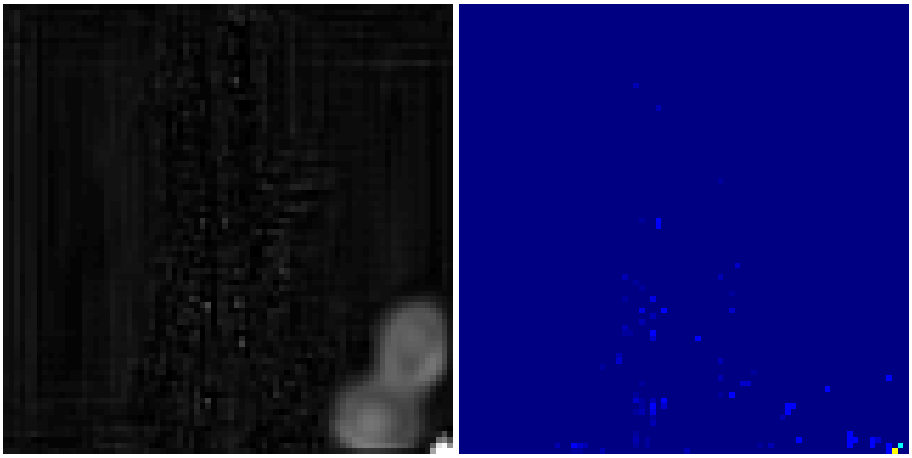

Figure 9.6: Visualization of activation from a single target layer (left) and the corresponding CAM (right) were generated using a model trained with standard 2D convolutional layers in the architecture design. Noticeable noise is observed in the activation map, particularly in the bottom left corner, and this noise propagates to the generated CAM.

The majority of noise in the generated CAMs is filtered out by utilizing the pointcloud mask, but a complete solution to the issue is not provided. When sensor detections are present on background objects, the masking process may not be effective in filtering them out. Upon examining all elements of Grad-CAM method, the root cause of the issue was identified during activation maps examination. In Figure 9.6, a visualization of a single activation is presented. It was chosen based on a high gradient weight value, which indicates that this activation carries significant importance for the CAM creation. Despite the crucial object indicators appearing in the middle of the activation map, the noise in the corner outweighs the rest, leading to a corrupted outcome. Similar cases were found throughout the dataset, and a common aspect of all of them is that the noise occurs in a part of the grid where input data is empty, due to the sparse nature of voxel-wise pointcloud processing.

The proposed solution to address the sparse nature of pointcloud data is to retrain the model with a specialized version of a convolution operation layer called Sparsity Invariant Convolution (Uhrig et al. 2017), proposed by researchers in the LiDAR depth-completion domain. The overall layer processing pipeline is illustrated in Figure 9.7. In essence, the new method builds upon the standard convolution operation. An additional mask is introduced to keep track of empty data cells, allowing some parts of the input to be omitted during the application of the convolutional filter. Moreover, each utilized input component of the convolution is weighted according to the positive mask values. This ensures that the results of the operation are normalized regardless of the number of non-empty inputs, which also supports the training process. Sparsity Invariant Convolution has gained popularity in depth-completion NN architecture as it significantly improves the processing of sparse LiDAR pointcloud data, resulting in an overall performance boost. Additionally, it affects intermediate activations used in the Grad-CAM method, indicating its potential to address the noise problem.

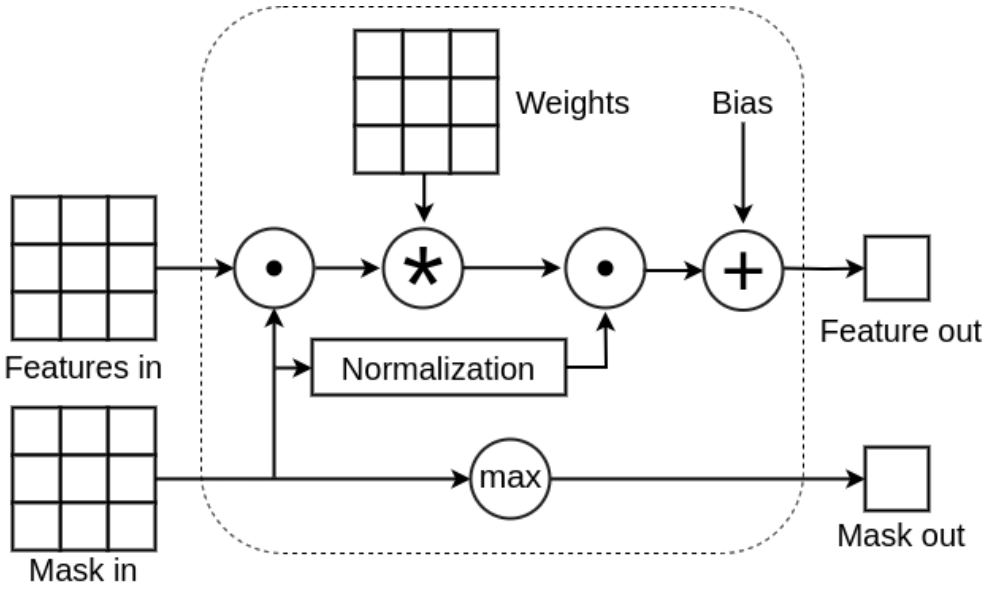

Figure 9.7: Sparsity Invariant Convolution operation diagram. This layer is specifically designed to handle convolutions with highly sparse input data. It employs a mask, propagated alongside the features, to monitor empty inputs within the network structure. The mask is subsequently used to filter out irrelevant values before conducting multiplication with layer-trainable weights. Moreover, the results are scaled according to the mask values before incorporating the bias.

Following the replacement of standard convolutional layers and subsequent retraining, the new point-cloud processing model undergoes the same activation analysis. The results, shown in Figure 9.8, reveal a significant improvement over the previous version. Sparsity invariant convolutions have effectively reduced the noise generated by empty input data, leading to a clearer CAM with improved distinction of important regions.

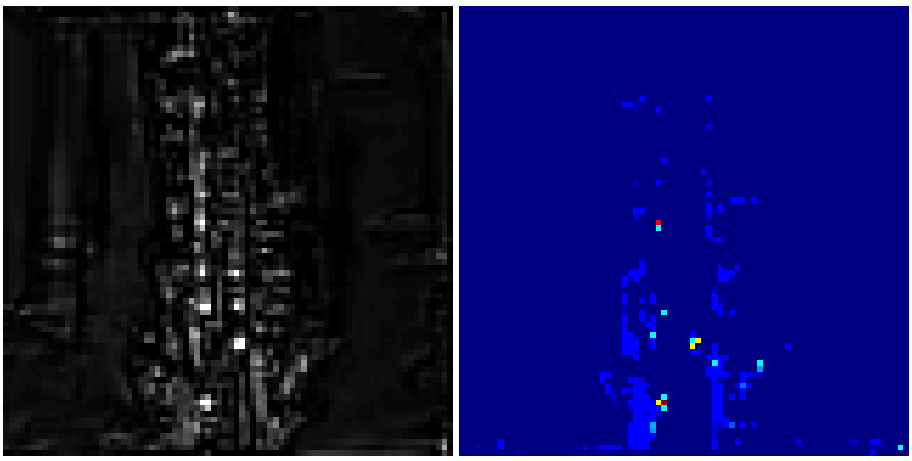

Figure 9.8: Visualization of the exact sample activation and associated CAM as in Figure 9.6. However, these outcomes are achieved using a model that incorporates Sparsity Invariant Convolutions. The example demonstrates that the sparsity invariant operation outperforms the standard convolution approach, particularly concerning empty input cells. Consequently, the generated CAM is more focused on actual detections with higher confidence.

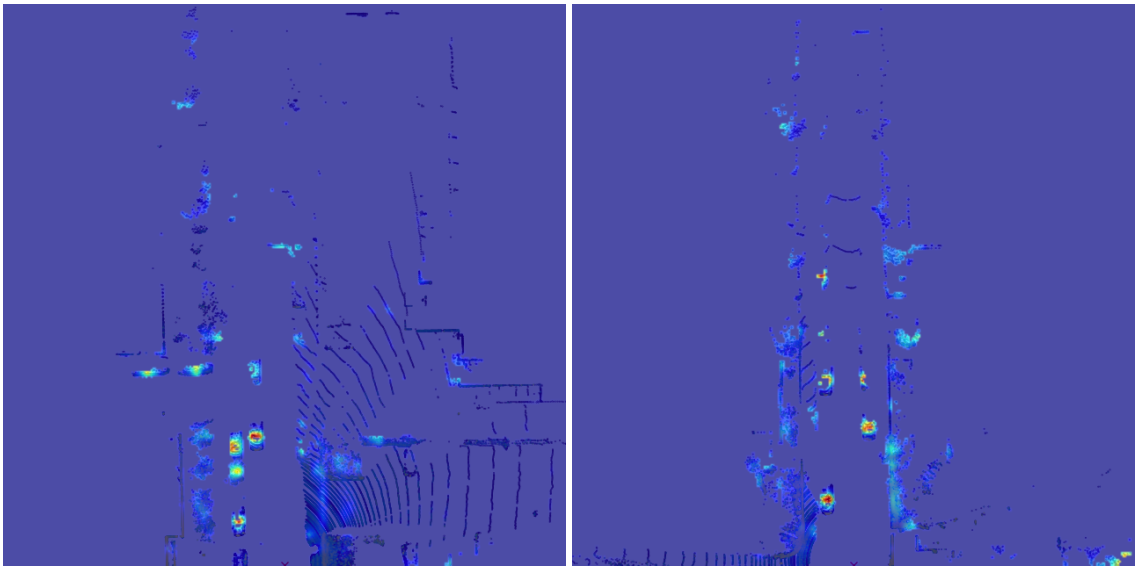

Figure 9.9: The ultimate Grad-CAM adaptation results for a LiDAR pointcloud OD network. By consolidating all experiments and findings, the final pipeline for CAM creation is developed. The results are a high-resolution, clear, and noise-free CAM heatmaps, which effectively highlight essential areas in the input pointcloud overlay.

The final result of the Grad-CAM adaptation for the LiDAR pointcloud network (Figure 9.9) combines voxel-wise processing and 2D Sparsity Invariant Convolutions to infer model output and intermediate activations from a target convolutional layer. Based on the proposed algorithm, class scores are determined from a multi-head, multi-feature model output tensor. Backpropagation is used to obtain gradients for the target layer with respect to the class score, which are then utilized to calculate activation weights for CAMs creation. Finally, the obtained CAMs are fused with the LiDAR pointcloud mask to enhance them with a higher level of detail, resulting in a high-resolution, detailed heatmap for object detection on a LiDAR sensor pointcloud, free of any undesirable noise that could disrupt human visual analysis.

Chapter summary:

- In the first part of this chapter, the adaptation of Grad-CAM technique for multi-scale architectures was presented. Using the described method, an application of XAI to proposed single-sensor models was showcased on the camera image processing network. Obtained results were described in the context of determining each scale features placement during CDSM range-based fusion.
- The second part described the research done regarding an unexplored topic of Grad-CAM adaptation to pointcloud models. The method addresses different input and CAM domains and provides a novel technique for visualizing them together. Additionally, the problem of noise in the activations was solved by in-depth analysis and application of Sparsity Invariant Convolutions.

Chapter 10

Conclusions

This thesis extensively explores various aspects of Autonomous Vehicle perception system, focusing on the fusion of data from automotive sensors to enhance the accuracy and robustness of Deep Learning perception models. The thesis starts with a general introduction to the AD automation levels classification, the definition of perception systems, and the role of sensors such as camera, LiDAR, and Radar in AV sensor suite setup. The concept of sensor data fusion is also discussed, covering different fusion levels and stages. Furthermore, an exhaustive survey of Object Detection Neural Networks used in perception systems is presented. This literature review encompasses both single-sensor models and fusion architectures. Additionally, evaluation methods, including perception KPI metrics and Explainable AI techniques like Grad-CAM, are introduced to assess model performance and provide insights into predicted outcomes. Following a theoretical introduction, the core element of this thesis, the Cross-Domain Spatial Matching fusion method is presented, providing the general idea, methodology, and implementation details behind it. This fusion architecture utilizes DL techniques for a feature-stage Low-Level Fusion. It comprises a novel CDSM domain alignment method and a set of distinct fusion strategies, such as proposed feature-wise and range-based approaches. Data and training details, single-sensor features extraction, and fusion methods assessment are described in a series of experiments performed on two automotive open-source datasets. Those experiments provide both visual and numerical evaluation of each solution and enable the comparison between them. The analysis of the obtained results leads to the final thesis conclusions.

The obtained conclusions firmly confirm the thesis's objective, demonstrating that DL-based LLF solutions hold the capacity to advance AV perception systems. The experiments' results demonstrate the successful functionality of the CDSM solution, with the fusion consistently outperforming single-sensor model architectures. The fusion of camera and LiDAR, while still superior, showcases only marginal improvement over LiDAR-only results. This confirms the distinct advantages of LiDARs and renders additional camera data rather insignificant in the fusion process. In contrast, the fusion of camera and Radar, a more desirable sensor setup for the automotive industry, presents significant enhancements. Evidenced by KPI metrics, such as a high mAP score for the car class in OD task, the analysis of outcomes reveals a synergistic compensation for each sensor's weaknesses, whether in estimating depth or enhancing data resolution. Notably, presented

instances of corner cases underline that the LLF strategy empowers the final model architecture to establish cross-sensor feature representations, yielding predictions for objects that remain evasive to individual single-sensor models.
For most notable accomplishments presented throughout the thesis, the author considers the following:

- The development and implementation of the CDSM fusion architecture stands out as a main contribution to the AV perception research domain. This LLF method offers a unique approach to aligning sensor data features from different domains, which aside from the fusion, could be used in other solutions, such as the presented 3D monocular camera architecture.
- The CDSM architecture introduces novel fusion techniques, enabled by the domain alignment component. Among those techniques, the range-based approach, which employs FoV-based features aggregation and refinement, proves to be the best in terms of KPI performance.
- The complete CDSM fusion architecture yields improved perception outcomes and could potentially serve as a viable alternative to current SOTA approaches.
- The successful adaptation of the Grad-CAM analysis technique to pointcloud models. This adaptation significantly contributes to overcoming a difficulty in the visualization and analysis of complex pointcloud-based LiDAR and Radar models. By providing graphical insights into the decision-making processes, it enhances the interpretability of such solutions, thus augmenting their applicability, enhancing the development process, and leveraging the advantages associated with the Grad-CAM.
- The research efforts, which resulted in scientific publications and patent applications. This achievement highlights the practical relevance and industrial implications of the research conducted in the domain of ML and AV perception.

Looking ahead, the trajectory of this study suggests several promising avenues for further exploration:

- Although not explored in the thesis, data augmentation has the potential to significantly boost model performance. The augmentation process introduces variability into the training data, enabling models to better generalize and adapt to diverse scenarios. When it comes to fusion models, implementing consistent augmentation across all input data domains can be rather challenging and far more complex than just for a single modality. Nonetheless, considering the benefits, a closer examination of data augmentation's feasibility within the realm of fusion is certainly justified.
- The utilization of raw Radar data for fusion, particularly by incorporating range-azimuth-Doppler readings format, offers a promising avenue for enhancing the fusion. Such an approach would require a suitable dataset with unprocessed Radar readings. Utilization of such data with DL approach should address the low-resolution issue, providing better single-sensor results and consequently enhancing the fusion even further.

- The extension of fusion XAI visualization techniques offers another intriguing direction for future research. By extending the Grad-CAM approach to generate simultaneous visualizations for each fusion input, comprising both images and pointclouds CAMs, researchers can provide more informative insights into the distinct contributions of individual sensors within fusion solutions.

In summary, the future landscape of research within the scope of this thesis is rich with possibilities. By addressing the aforementioned problems, subsequent research can build upon the foundations laid in this work, aiming to further innovate the domain of AV perception systems.

References

1 Daniel Dworak, Filip Ciepiela, et al. “Performance of LiDAR object detection deep learning architectures based on artificially generated point cloud data from CARLA simulator”. In: *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*. 2019, pp. 600–605.

2 Daniel Dworak. “BlurNet: Keeping Collected Data Private with a Neural Network Based Pipeline”. In: *Advanced, Contemporary Control*. Springer International Publishing, 2020, pp. 1237–1248.

3 Jerzy Baranowski et al. “Analiza danych i optymalizacja w Przemyśle 4.0 — Data analysis and optimization in Industry 4.0”. In: *Wydział Elektryczny AGH – Wczoraj, Dziś i Jutro*. 2022, pp. 43–52.

4 Daniel Dworak and Jerzy Baranowski. “Adaptation of Grad-CAM Method to Neural Network Architecture for LiDAR Pointcloud Object Detection”. In: *Energies* 15.13 (2022).

5 Filip Ciepiela, Mariusz Karol Nowak, et al. “Automotive Radar Detection Level Modeling with Neural Networks”. In: *Advanced, Contemporary Control*. Cham: Springer Nature Switzerland, 2023, pp. 254–265.

6 Mateusz Komorkiewicz et al. “Vehicles, systems, and methods for determining an entry of an occupancy map of a vicinity of a vehicle”. EP3832531A1, Patent application. 2021.

7 Filip Ciepiela, Mateusz Komorkiewicz, et al. “Method and system for determining an output of a convolutional block of an artificial neural network”. EP3885996A1, Patent application. 2021.

8 Mateusz Wójcik et al. “Method and system for interpolation and method and system for determining a map of a surrounding of a vehicle”. EP3975105A1, Patent application. 2022.

9 Ori Maoz et al. “Methods and systems for determining candidate data sets for labelling”. EP3985560A1, Patent application. 2022.

10 Darsh Parekh et al. “A Review on Autonomous Vehicles: Progress, Methods and Challenges”. In: *Electronics* 11.14 (2022).

11 Walter Morales-Alvarez et al. “Vehicle Automation Field Test: Impact on Driver Behavior and Trust”. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–6.

12 Marco Galvani. “History and future of driver assistance”. In: *IEEE Instrumentation & Measurement Magazine* 22.1 (2019), pp. 11–16.

13 Pawel Skruch et al. “Safety of Perception Systems in Vehicles of High-Level Motion Automation”. In: *2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. 2022, pp. 561–566.

14 Mohammad Javad Shafiee et al. “Deep Neural Network Perception Models and Robust Autonomous Driving Systems: Practical Solutions for Mitigation and Improvement”. In: *IEEE Signal Processing Magazine* 38.1 (2021), pp. 22–30.

15 Jelena Kocić, Nenad Jovičić, and Vujo Drndarević. “Sensors and Sensor Fusion in Autonomous Vehicles”. In: *2018 26th Telecommunications Forum (TELFOR)*. 2018, pp. 420–425.

16 Matthias Pollach, Felix Schiegg, and Alois Knoll. “Low Latency And Low-Level Sensor Fusion For Automotive Use-Cases”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 6780–6786.

17 Zhiqing Wei et al. “MmWave Radar and Vision Fusion for Object Detection in Autonomous Driving: A Review”. In: *Sensors* 22.7 (2022).

18 Philipp Lindner et al. “Multi level fusion with confidence measures for automotive safety applications”. In: *2007 10th International Conference on Information Fusion*. 2007, pp. 1–7.

19 Thomas Herpel et al. “Multi-sensor data fusion in automotive applications”. In: *2008 3rd International Conference on Sensing Technology*. 2008, pp. 206–211.

20 László Lindenmaier et al. “Comparison of Sensor Data Fusion Algorithms for Automotive Perception System”. In: *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 2022, pp. 000089–000096.

21 Árpád Takács et al. “Highly Automated Vehicles and Self-Driving Cars [Industry Tutorial]”. In: *IEEE Robotics & Automation Magazine* 25.4 (2018), pp. 106–112.

22 Matthew Barth, Kanok Boriboonsomsin, and Guoyuan Wu. “The potential role of vehicle automation in reducing traffic-related energy and emissions”. In: *2013 International Conference on Connected Vehicles and Expo (ICCVE)*. 2013, pp. 604–605.

23 Society of Automotive Engineers. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE International, 2018.

24 Dominique Gruyer et al. “Perception, information processing and modeling: Critical stages for autonomous driving applications”. In: *Annual Reviews in Control* 44 (2017), pp. 323–341.

25 Ashish Pandharipande et al. “Sensing and Machine Learning for Automotive Perception: A Review”. In: *IEEE Sensors Journal* (2023), pp. 1–1.

26 Sandip Ray. “Safety, Security, and Reliability: The Automotive Robustness Problem and an Architectural Solution”. In: *2019 IEEE International Conference on Consumer Electronics (ICCE)*. 2019, pp. 1–4.

27 Andre Kohn et al. “Fail-operational in safety-related automotive multi-core systems”. In: *10th IEEE International Symposium on Industrial Embedded Systems (SIES)*. 2015, pp. 1–4.

28 Aptiv. *Smart Vehicle Architecture*. 2022. URL: `https://www.aptiv.com/` (visited on 07/30/2023).

29 O. Eytan and E. Belman. “High-resolution automotive lens and sensor”. US 2019/0 377 110 A1, Patent application publication. 2019.

30 Alen Luštica. “CCD and CMOS image sensors in new HD cameras”. In: *Proceedings ELMAR-2011*. 2011, pp. 133–136.

31 Junichi Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras*. USA: CRC Press, Inc., 2005.

32 Paweł Pawłowski, Karol Piniarski, and Adam Dąbrowski. “Highly Efficient Lossless Coding for High Dynamic Range Red, Clear, Clear, Clear Image Sensors”. In: *Sensors* 21.2 (2021).

33 Han-Wen Huang, Chuan-Ren Lee, and Hung-Pang Lin. “Nighttime vehicle detection and tracking base on spatiotemporal analysis using RCCC sensor”. In: *2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. 2017, pp. 1–5.

34 Kamil Lelowicz, Michał Jasiński, and Adam Krzysztof Piłat. “Discussion of Novel Filters and Models for Color Space Conversion”. In: *IEEE Sensors Journal* 22.14 (2022), pp. 14165–14176.

35 Korbinian Weikl, Damien Schroeder, and Walter Stechele. “Optimization of automotive color filter arrays for traffic light color separation”. In: vol. 2020. Nov. 2020.

36 Naveen Kuruba et al. “A Generic Method to Estimate Camera Extrinsic Parameters”. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 1571–1575.

37 Kamil Lelowicz. “Camera model for lens with strong distortion in automotive application”. In: *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*. 2019, pp. 314–319.

38 Diego Gonzalez-Aguilera, Javier Gomez-Lahoz, and Pablo Rodriguez-Gonzalvez. “An Automatic Approach for Radial Lens Distortion Correction From a Single Image”. In: *IEEE Sensors Journal* 11.4 (2011), pp. 956–965.

39 Mial E Warren. “Automotive LIDAR Technology”. In: *2019 Symposium on VLSI Circuits*. 2019, pp. C254–C255.

40 Ricardo Roriz, Jorge Cabral, and Tiago Gomes. “Automotive LiDAR Technology: A Survey”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2022), pp. 6282–6297.

41 Rajeev Thakur. “Scanning LIDAR in Advanced Driver Assistance Systems and Beyond: Building a road map for next-generation LIDAR technology”. In: *IEEE Consumer Electronics Magazine* 5.3 (2016), pp. 48–54.

42 Ching-Pai Hsu et al. “A Review and Perspective on Optical Phased Array for Automotive LiDAR”. In: *IEEE Journal of Selected Topics in Quantum Electronics* 27.1 (2021), pp. 1–16.

43 Daniel Bastos et al. “An Overview of LiDAR Requirements and Techniques for Autonomous Driving”. In: *2021 Telecoms Conference (ConfTELE)*. 2021, pp. 1–6.

44 Fredrik Schalling, Sebastian Ljungberg, and Naveen Mohan. “Benchmarking LiDAR Sensors for Development and Evaluation of Automotive Perception”. In: *2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*. 2019, pp. 1–6.

45 Andrew M. Wallace, Abderrahim Halimi, and Gerald S. Buller. “Full Waveform LiDAR for Adverse Weather Conditions”. In: *IEEE Transactions on Vehicular Technology* 69.7 (2020), pp. 7064–7077.

46 Zhuoqun Dai et al. “Requirements for Automotive LiDAR Systems”. In: *Sensors* 22.19 (2022).

47 Gunzung Kim, Jeongsook Eom, and Yongwan Park. “An Experiment of Mutual Interference between Automotive LIDAR Scanners”. In: *2015 12th International Conference on Information Technology - New Generations*. 2015, pp. 680–685.

48 Gor Hakobyan and Bin Yang. “High-Performance Automotive Radar: A Review of Signal Processing Algorithms and Modulation Schemes”. In: *IEEE Signal Processing Magazine* 36.5 (2019), pp. 32–44.

49 Sujeet Milind Patole et al. “Automotive radars: A review of signal processing techniques”. In: *IEEE Signal Processing Magazine* 34.2 (2017), pp. 22–35.

50 Christian Waldschmidt, Juergen Hasch, and Wolfgang Menzel. “Automotive Radar — From First Efforts to Future Systems”. In: *IEEE Journal of Microwaves* 1.1 (2021), pp. 135–148.

51 J. Wenger. “Automotive radar - status and perspectives”. In: *IEEE Compound Semiconductor Integrated Circuit Symposium, 2005. CSIC ’05.* 2005.

52 Volker Winkler. “Range Doppler detection for automotive FMCW radars”. In: *2007 European Radar Conference*. 2007, pp. 166–169.

53 Wolfgang Menzel and Arnold Moebius. “Antenna Concepts for Millimeter-Wave Automotive Radar Sensors”. In: *Proceedings of the IEEE* 100.7 (2012), pp. 2372–2379.

54 D. Kok and J.S. Fu. “Signal processing for automotive radar”. In: *IEEE International Radar Conference, 2005.* 2005, pp. 842–846.

55 Wanyou Yu et al. “Velocity Estimation of Wheeled Vehicles with Micro-Doppler Phenomenon for Automotive Radar”. In: *2018 International Conference on Sensor Networks and Signal Processing (SNSP)*. 2018, pp. 205–212.

56 Eugin Hyun, Young-Seok Jin, and Jong-Hun Lee. “Moving and stationary target detection scheme using coherent integration and subtraction for automotive FMCW radar systems”. In: *2017 IEEE Radar Conference (RadarConf)*. 2017, pp. 0476–0481.

57 Marcel Sheeny et al. “RADIATE: A Radar Dataset for Automotive Perception in Bad Weather”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. May 2021, pp. 1–7.

58 Nicolas Scheiner, Ole Schumann, et al. “Off-the-shelf sensor vs. experimental radar - How much resolution is necessary in automotive radar classification?” In: *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*. 2020, pp. 1–8.

59 Masahiro Umehira et al. “Inter-radar interference in automotive FMCW radars and its mitigation challenges”. In: *2020 IEEE International Symposium on Radio-Frequency Integration Technology (RFIT)*. 2020, pp. 220–222.

60 Faruk Uysal and Sasanka Sanka. “Mitigation of automotive radar interference”. In: *2018 IEEE Radar Conference (RadarConf18)*. 2018, pp. 0405–0410.

61 L. Wald. “Some terms of reference in data fusion”. In: *IEEE Transactions on Geoscience and Remote Sensing* 37.3 (1999), pp. 1190–1193.

62 Keli Huang et al. “Multi-modal Sensor Fusion for Auto Driving Perception: A Survey”. In: Feb. 2022.

63 Bharanidhar Duraisamy et al. “Track level fusion of extended objects from heterogeneous sensors”. In: *2016 19th International Conference on Information Fusion (FUSION)*. 2016, pp. 876–885.

64 Ullrich Scheunert et al. “Early and Multi Level Fusion for Reliable Automotive Safety Systems”. In: *2007 IEEE Intelligent Vehicles Symposium*. 2007, pp. 196–201.

65 Jing Ren, Hossam Gaber, and Sk Sami Al Jabar. “Applying Deep Learning to Autonomous Vehicles: A Survey”. In: *2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD)*. 2021, pp. 247–252.

66 C.M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, USA, 1995.

67 Kenji Suzuki. *Artificial Neural Networks*. Rijeka: IntechOpen, Apr. 2011.

68 Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. Cambridge, MA, USA: MIT Press, 2016.

69 Josh Patterson and Adam Gibson. *Deep Learning: A Practitioner’s Approach*. Beijing: O’Reilly, 2017.

70 Charu C. Aggarwal. *Neural Networks and Deep Learning. A Textbook*. Cham: Springer, 2018, p. 497.

71 Martin Kaloev and Georgi Krastev. “Comparative Analysis of Activation Functions Used in the Hidden Layers of Deep Neural Networks”. In: *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. 2021, pp. 1–5.

72 Y. Le Cun et al. “Handwritten Digit Recognition with a Back-Propagation Network”. In: *Advances in Neural Information Processing Systems 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, pp. 396–404.

73 Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105.

74 Ethem Alpaydin. *Introduction to Machine Learning*. 2nd. The MIT Press, 2010.

75 Mathukumalli Vidyasagar. *Learning and Generalization: With Applications to Neural Networks*. 2nd. Springer Publishing Company, Incorporated, 2010.

76 Abbas Zohrevand and Zahra Imani. “An Empirical Study of the Performance of Different Optimizers in the Deep Neural Networks”. In: *2022 International Conference on Machine Vision and Image Processing (MVIP)*. 2022, pp. 1–5.

77 Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

78 Simon S. Haykin. *Neural networks and learning machines*. Third. Upper Saddle River, NJ: Pearson Education, 2009.

79 S. Ravichandiran. *Hands-On Deep Learning Algorithms with Python*. Packt Publishing, 2019.

80 Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

81 Christian Szegedy et al. "Going deeper with convolutions". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9.

82 Ashish Vaswani et al. "Attention is All You Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010.

83 Sanghyun Woo et al. "CBAM: Convolutional Block Attention Module". In: *CoRR* abs/1807.06521 (2018). arXiv: 1807.06521.

84 Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-Excitation Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7132–7141.

85 Ross Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 580–587.

86 Ross Girshick. "Fast R-CNN". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1440–1448.

87 Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1137–1149.

88 Manuel Carranza-García et al. "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data". In: *Remote Sensing* 13.1 (2021).

89 Joseph Redmon, Santosh Divvala, et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788.

90 Joseph Redmon and Ali Farhadi. "YOLO9000: Better, Faster, Stronger". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6517–6525.

91 Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: *CoRR* abs/1804.02767 (2018). arXiv: 1804.02767.

92 Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: *CoRR* abs/2004.10934 (2020). arXiv: 2004.10934.

93 Shu Liu et al. "Path Aggregation Network for Instance Segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8759–8768.

94 Zhaohui Zheng et al. "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression". In: *CoRR* abs/1911.08287 (2019). arXiv: 1911.08287.

95 Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2999–3007.

96 Kaiwen Duan et al. "CenterNet: Keypoint Triplets for Object Detection". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6568–6577.

97 Zhi Tian et al. "FCOS: Fully Convolutional One-Stage Object Detection". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9626–9635.

98 Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *CoRR* abs/1905.11946 (2019). arXiv: 1905.11946.

99 Mingxing Tan, Ruoming Pang, and Quoc V. Le. "EfficientDet: Scalable and Efficient Object Detection". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 10778–10787.

100 Mingxing Tan and Quoc V. Le. "EfficientNetV2: Smaller Models and Faster Training". In: *CoRR* abs/2104.00298 (2021). arXiv: 2104.00298.

101 Fisher Yu, Dequan Wang, and Trevor Darrell. "Deep Layer Aggregation". In: *CoRR* abs/1707.06484 (2017). arXiv: 1707.06484.

102 Kai Han et al. "A Survey on Vision Transformer". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2023), pp. 87–110.

103 Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021.

104 Ze Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9992–10002.

105 Eduardo Arnold et al. "A Survey on 3D Object Detection Methods for Autonomous Driving Applications". In: *IEEE Transactions on Intelligent Transportation Systems* 20.10 (2019), pp. 3782–3795.

106 Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. "Tracking Objects as Points". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 474–490.

107 Tai Wang et al. "FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection". In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2021, pp. 913–922.

108 Tai Wang et al. "Probabilistic and Geometric Depth: Detecting Objects in Perspective". In: *Conference on Robot Learning*. 2021.

109 Hansheng Chen et al. "EPro-PnP: Generalized End-to-End Probabilistic Perspective-n-Points for Monocular Object Pose Estimation". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 2771–2780.

110 Zhiqi Li et al. "BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers". In: *European Conference on Computer Vision*. 2022.

111 R. Qi Charles et al. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 77–85.

112 Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 770–779.

113 Yin Zhou and Oncel Tuzel. "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4490–4499.

114 Alex H. Lang et al. "PointPillars: Fast Encoders for Object Detection From Point Clouds". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 12689–12697.

115 Shaoshuai Shi, Chaoxu Guo, et al. "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 10526–10535.

116 Heejae Han et al. "Object classification on raw radar data using convolutional neural networks". In: *2019 IEEE Sensors Applications Symposium (SAS)*. 2019, pp. 1–6.

117 Li Wang, Jun Tang, and Qingmin Liao. "A Study on Radar Target Detection Based on Deep Neural Networks". In: *IEEE Sensors Letters* 3.3 (2019), pp. 1–4.

118 Youngwook Kim et al. “Human Detection with Range-Doppler Signatures Using 3D Convolutional Neural Networks”. In: *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. 2020, pp. 2440–2442.

119 Jihoon Kwon, Seungeui Lee, and Nojun Kwak. “Human Detection by Deep Neural Networks Recognizing Micro-Doppler Signals of Radar”. In: *2018 15th European Radar Conference (EuRAD)*. 2018, pp. 198–201.

120 Sangtae Kim et al. “Moving Target Classification in Automotive Radar Systems Using Convolutional Recurrent Neural Networks”. In: *2018 26th European Signal Processing Conference (EUSIPCO)*. 2018, pp. 1482–1486.

121 Bence Major et al. “Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors”. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 924–932.

122 Nicolas Scheiner, Florian Kraus, et al. “Object detection for automotive radar point clouds – a comparison”. In: *AI Perspectives* 3 (Nov. 2021).

123 Baowei Xu et al. “RPFA-Net: a 4D RaDAR Pillar Feature Attention Network for 3D Object Detection”. In: *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. 2021, pp. 3061–3066.

124 Andras Palffy et al. “Multi-class Road User Detection with 3+1D Radar in the View-of-Delft Dataset”. English. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4961–4968.

125 Alexander Popov et al. “NVRadarNet: Real-Time Radar Obstacle and Free Space Detection for Autonomous Driving”. In: *CoRR* abs/2209.14499 (2022). arXiv: 2209.14499.

126 Huu-Sy Le, Tan Duy Le, and Kha-Tu Huynh. “A Review on 3D Object Detection for Self-Driving Cars”. In: *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*. 2022, pp. 398–403.

127 Sourabh Vora et al. “PointPainting: Sequential Fusion for 3D Object Detection”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4603–4611.

128 Charles Ruizhongtai Qi et al. “Frustum PointNets for 3D Object Detection from RGB-D Data”. In: June 2018, pp. 918–927.

129 Luca Caltagirone et al. “LIDAR-camera fusion for road detection using fully convolutional neural networks”. In: *Robotics and Autonomous Systems* 111 (Nov. 2018).

130 Florian Wulff et al. “Early Fusion of Camera and Lidar for robust road detection based on U-Net FCN”. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1426–1431.

131 Felix Nobis et al. “A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection”. In: *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. 2019, pp. 1–7.

132 Xiaozhi Chen et al. “Multi-view 3D Object Detection Network for Autonomous Driving”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6526–6534.

133 Jason Ku et al. “Joint 3D Proposal Generation and Object Detection from View Aggregation”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 1–8.

134 Danfei Xu, Dragomir Anguelov, and Ashesh Jain. “PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 244–253.

135 Gregory P. Meyer, Jake Charland, et al. “Sensor Fusion for Joint 3D Object Detection and Semantic Segmentation”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 1230–1237.

136 Gregory P. Meyer, Ankita Gajanan Laddha, et al. “LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 12669–12678.

137 Jin Yoo et al. “3D-CVF: Generating Joint Camera and LiDAR Features Using Cross-view Spatial Feature Fusion for 3D Object Detection”. In: Nov. 2020, pp. 720–736.

138 Ming Liang et al. “Multi-Task Multi-Sensor Fusion for 3D Object Detection”. In: June 2019, pp. 7337–7345.

139 Michael Meyer and Georg Kuschk. “Deep Learning Based 3D Object Detection for Automotive Radar and Camera”. In: *2019 16th European Radar Conference (EuRAD)*. 2019, pp. 133–136.

140 Jyh-Jing Hwang et al. “CramNet: Camera-Radar Fusion With Ray-Constrained Cross-Attention For Robust 3D Object Detection”. In: *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*. Tel Aviv, Israel: Springer-Verlag, 2022, pp. 388–405.

141 Ramin Nabati and Hairong Qi. “CenterFusion: Center-based Radar and Camera Fusion for 3D Object Detection”. In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2021, pp. 1526–1535.

142 Xuanyao Chen et al. “FUTR3D: A Unified Sensor Fusion Framework for 3D Detection”. In: *arXiv preprint arXiv:2203.10642* (2022).

143 Zhijian Liu et al. “BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird’s-Eye View Representation”. In: (May 2022).

144 Xuyang Bai et al. “TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection with Transformers”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 1080–1089.

145 Holger Caesar et al. “nuScenes: A Multimodal Dataset for Autonomous Driving”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11618–11628.

146 Riccardo Guidotti et al. “A Survey of Methods for Explaining Black Box Models”. In: *ACM Computing Surveys* 51 (Feb. 2018).

147 Amina Adadi and Mohammed Berrada. “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)”. In: *IEEE Access* 6 (2018), pp. 52138–52160.

148 Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. “Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models”. In: *CoRR* abs/1708.08296 (2017). arXiv: 1708.08296.

149 Daniel Omeiza et al. “Explanations in Autonomous Driving: A Survey”. In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–21.

150 Bolei Zhou et al. “Learning Deep Features for Discriminative Localization”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2921–2929.

151 Ramprasaath R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 618–626.

152 Aditya Chattopadhay et al. “Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks”. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2018, pp. 839–847.

153 Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for autonomous driving? The KITTI vision benchmark suite”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 3354–3361.

154 Ilya Loshchilov and Frank Hutter. “SGDR: Stochastic Gradient Descent with Restarts”. In: *ArXiv* abs/1608.03983 (2016).

155 *NuScenes ranking*. 2020. URL: https://www.nuscenes.org/object-detection (visited on 07/04/2023).

156 Jonas Uhrig et al. “Sparsity Invariant CNNs”. In: *2017 International Conference on 3D Vision (3DV)*. 2017, pp. 11–20.