

École polytechnique de Louvain

Efficient Processing of Radar Data Through Transformers for Robust Object Detection and Tracking

Author: **Samuel KOUFF**

Supervisors: **Benoît MACQ, Christophe CRAEYE**

Readers: **Gauthier ROTSART DE HERTAING, Nicolas BIOUL, Cristel
PELSSER**

Academic year 2025–2026

Master [120] in Computer Science

Acknowledgements

This thesis marks the end of a three-year Master’s degree in Computer Science at Université Catholique de Louvain. Having completed a Bachelor’s degree in application development, I didn’t know what to expect from a university environment. These three years proved to be a formative experience, significantly expanding my technical knowledge and shaping my professional outlook. In particular, they allowed me to discover the field of artificial intelligence, network and systems.

I would like to express my sincere gratitude to my supervisors, Professors Benoît Macq and Christophe Craeye, for their valuable guidance and support throughout our meetings. I’m also deeply grateful to Gauthier Rotsart de Hertaing for his continuous support and availability, his advice, his assistance in getting to grips with the radar hardware, and for redeveloping the device’s bimodal data acquisition pipeline. I would further like to thank the OpenHub for providing the multi-modal acquisition setup that made the characterisation part of this work possible. Finally, I would like to express my gratitude to my family, my friends, and my girlfriend for their support and encouragement throughout this journey.

Samuel Kouff, May 2026

Abstract

Autonomous driving perception refers to the ability of autonomous driving systems to effectively perceive the surrounding environment, enabling vehicles to understand and respond to various driving scenarios [1]. Among the sensors used in this field, radar has attracted significant attention owing to its robustness to adverse weather conditions and its low cost. However, radar data is sparse and unstructured, making it challenging to train deep learning models for object detection.

This master’s thesis addresses the following research question: How can radar data be efficiently processed for robust object detection and tracking? To this end, RadViT, a Vision Transformer-based architecture, was designed around three guiding principles: (i) enabling fair comparisons between multiple radar input representations, (ii) exploiting the multi-antenna structure of radar data, and (iii) remaining independent of the specific radar hardware, so as to facilitate generalisation across different sensor configurations. In a second part, the K-MD2 FMCW radar sensor was experimentally characterised through measurements conducted in an anechoic chamber, providing a deeper understanding of the physical phenomena and signal artefacts underlying radar data. A crucial step in ensuring true operational robustness for radar object detection and enabling the quantification of the domain shift between the radar used for training and the K-MD2.

Contents

1	Introduction	5
1.1	Context and motivations	5
1.2	Proposed Contribution	6
1.3	Plan of the thesis	7
2	Literature Review	8
2.1	Radar working principle	8
2.2	Radar Processing	8
2.2.1	Frequency Modulated Continuous Wave	9
2.3	Radar data representations	10
2.3.1	ADC signal	11
2.3.2	Radar tensor	12
2.3.3	Point cloud	13
2.4	Object detection task	14
2.5	Tracking task	15
2.6	Autonomous Driving datasets	16
2.6.1	RADial	17
2.7	Existing radar representations and architectural approaches	18
2.7.1	ADC signal	19
2.7.2	Radar tensor	19
2.7.3	Point cloud	20
2.8	Comprehensive Study of FFT-RadNet and T-FFTRadNet	21
2.8.1	FFT-RadNet	21
2.8.2	T-FFTRadNet	23
3	Radar object detection and tracking through Transformers	25
3.1	Vision Transformers	25
3.1.1	Original Vision Transformers	25
3.1.2	Adapting ViTs for dense prediction	27
3.1.3	Motivation for a ViT-based approach in Radar perception	28
3.2	RadViT: A Vision Transformer for radar object detection	28
3.2.1	Architecture overview	29
3.2.2	Radar input processing	29
3.2.3	Radar-adapted patch embedding	30
3.2.4	Transformer Encoder	31
3.2.5	Multi-scale features pyramid	31

3.2.6	Range-Angle decoder	32
3.2.7	Detection header	33
3.3	Design choices and explored alternatives	33
3.4	Extending the RADIAL dataset for classification and tracking	34
3.5	Experimental results	35
3.5.1	Dataset split and label encoding	35
3.5.2	Training configuration	36
3.5.3	Detection metrics used	36
3.5.4	Detection performances	37
3.5.5	Classification performances	38
3.6	Multi-Object Tracking with Kalman Filtering	42
3.6.1	Kalman Filter-Based Algorithm	42
3.6.2	Implementation of the tracking algorithm	43
3.6.3	Tracking metrics used	44
3.6.4	Tracking Results and Analysis	44
3.7	Summary	46
4	Characterisation of K-MD2 radar	47
4.1	Devices characteristics	47
4.2	K-MD2 radar calibration in Anechoic Chamber	48
4.2.1	Computation of the system's intrinsic noise floor	48
4.3	Radar Cross-Section	50
4.3.1	Radar Cross-Section computation	51
4.3.2	Experimental Results	52
4.4	Radar artefacts	55
4.4.1	I/Q Imbalance	55
4.4.2	Range and Doppler aliasing	57
4.5	Summary	58
5	Next steps	60
5.1	Methods to mitigate overfitting	60
5.2	Transfer learning strategies	60
5.3	Multi-modal fusion with camera	61
5.4	Improvement of Tracking pipeline	61
6	Conclusion	62
A	RadViT Architecture Diagrams	64
B	Complementary results	66
C	Detection output visualisation	67
D	Declaration on the use of AI	69

Chapter 1

Introduction

1.1 Context and motivations

The World Health Organisation’s recent report estimates that approximately 1.19 million people die each year in road crashes [2]. In Belgium, there were 35.916 accidents in 2024, 443 of which resulted in fatalities within 30 days [3]. Therefore, there is an urgent need to develop robust autonomous systems that help drivers prevent accidents and respond to potential dangers as they arise. Autonomous driving perception is the ability of autonomous driving systems to effectively perceive the surrounding environment, enabling vehicles to understand and respond to various driving scenarios [1]. Nowadays, more and more vehicles are equipped with Advanced Driver Assistance Systems (ADAS). These systems refer to the range of on-board technologies that assist the driver with driving tasks. They include features such as collision warning or intervention, brake assist, lane-keeping assist, and more [4].

One of the primary challenges in autonomous driving perception lies in ensuring the reliability and accuracy of onboard sensors. This is precisely the objective addressed by the MUSE project, in collaboration with the OpenHub and UCLouvain, of which my master’s thesis is part. In general, three main sensor types are considered in autonomous driving: radar, LiDAR and camera. Cameras provide rich semantic and visual information, yet their performance degrades significantly under challenging conditions such as low illumination, occlusions and adverse weather. Their low manufacturing cost and widespread availability, however, make them the most economically accessible sensing modality. LiDAR, on the other hand, generates a dense, high-resolution three-dimensional point cloud through laser pulses, enabling precise depth and distance estimation for accurate 3D scene reconstruction. Nevertheless, its performance similarly deteriorates in adverse weather, because laser pulses attenuate or bounce when they encounter suspended particles like rain, snow, or smoke [5]. It inherently lacks semantic information, and its high production cost represented a significant barrier to large-scale deployment.

Radar sensors, by contrast, provide reliable velocity, range measurements and perform well under low-light conditions and in occluded environments [6]. Unlike other light-wave-based sensors, radar emits radio waves at wavelengths significantly longer than

the size of atmospheric particles such as water droplets, dust, or smoke. As a result, radar signals are not scattered by these particles. In terms of cost, radar sensors span a broad range, from low-cost devices to advanced imaging systems, making them adaptable to diverse deployment scenarios. With these characteristics, radars are widely used in ADAS systems. However, radar remains the sparsest of the three sensing modalities. This comes at the cost of reduced angular resolution, sparse point cloud density, elevation ambiguity, and susceptibility to clutter. Since 2024, the emergence of 4D radar [7] has extended conventional radar by adding elevation estimation alongside range, azimuth, and radial velocity. This fourth dimension yields denser point clouds with precise height information, similar to LiDAR data, while retaining radar’s velocity measurement capability. However, 4D radar sensors remain significantly more expensive than their 3D counterparts [1].

In this context, where it is necessary to strike the right balance among cost, accuracy, and reliability, determining which sensors to use and how to effectively combine their data streams remains one of the most active research areas in autonomous perception. LiDAR-camera fusion has been extensively studied and has achieved strong performance in both 2D and 3D object detection. Yet both modalities share a common vulnerability to adverse weather conditions, which limits their operational robustness. Radar-camera fusion has therefore attracted growing interest, as the two modalities complement each other well and offer a cost-effective alternative [8]. This specific research question is central to Léonel Ngague’s master’s thesis, which explores the optimal combination of modalities for object detection tasks.

While multi-modal fusion improves overall perception, the combination succeeds only if each sensor achieves sufficient and comparable performance. Object detection from raw camera images is a mature field with well-established architectures such as YOLO [9], Mask R-CNN [10], and DETR [11], which achieve state-of-the-art performance. LiDAR-based detection has similarly reached maturity, with point-based methods such as PointPillars [12] serving as strong baselines. Radar-based detection has also seen significant research activity in recent years, yet it remains challenging due to the data’s inherent sparsity and unstructured nature. It is precisely this challenge that motivates the focus of this master’s thesis.

1.2 Proposed Contribution

This master’s thesis addresses the following research question: How can radar data be efficiently processed to enable robust object detection and tracking? To answer this question, we make the following contributions. Primarily, we designed RadViT, a Vision Transformer-based architecture for radar object detection. RadViT is designed around three guiding principles: (i) enabling fair comparisons between multiple radar input representations, (ii) exploiting the multi-antenna structure of radar data, and (iii) remaining independent of the specific radar hardware, so as to facilitate generalisation across different sensor configurations. In addition, this work make two complementary contributions. First it extends the RADIAL dataset initially designed only for object

detection with vehicle classification labels and multi-object tracking annotations to support a complete evaluation pipeline. Second, a Kalman filter-based multi-object tracker was developed and evaluated, building on the detections produced by RadViT. Finally, the K-MD2 FMCW radar was experimentally characterised in an anechoic chamber, including measurements of the noise floor, radar cross-section, and signal artefacts. This characterisation is essential to provide true operational robustness for radar object detection and to quantify the domain shift between the radar used for training and the K-MD2 radar sensor used in the MUSE project. The code is available at https://github.com/kouffsamuel/master_thesis.

1.3 Plan of the thesis

After this introduction, Chapter 2 provides a comprehensive literature review. It establishes the theoretical foundation of radar sensing, covering its operating principles and the resulting data structures. This chapter also evaluates the signal processing methods used to transform raw measurements into learnable representations and justifies the selection of the RADIAL dataset for training and evaluation.

Chapter 3 presents the first main contributions from the research conducted: RadViT, a Vision Transformer-based architecture designed for radar object detection. We evaluate its performance on both Range-Doppler and ADC signals, comparing it against state-of-the-art architectures. This chapter further extends the pipeline with a Kalman filter-based multi-object tracking algorithm, whose results are benchmarked against the SORT algorithm.

Chapter 4 presents the second main contribution by characterising the K-MD2 radar. Conducted in an anechoic chamber, this study quantifies the sensor’s noise floor and Radar Cross-Section (RCS) of various targets, providing a necessary foundation for future transfer learning applications.

Chapter 5 explores future research directions, including methods to mitigate overfitting, transfer learning strategies, multi-modal fusion with camera and improvement of the tracking pipeline, before concluding this master’s thesis in 6.

Chapter 2

Literature Review

This chapter provides a review of the relevant literature. It begins with the working principles of radar, covering signal processing and data representations. Object detection in the radar context is then discussed, followed by an overview of autonomous driving datasets and neural architectures dedicated to object detection.

2.1 Radar working principle

Radar stands for "Radio Detection And Ranging". The goal of this sensor is to calculate the range and velocity of targets by transmitting radio waves and receiving their reflections [13]. Radar is composed of one or multiple transmitter and receiver antennas. The time elapsed between the transmission of the original signal and the reception of the echo, known as the Time Of Flight, is used to determine an object's range by calculating the time difference between the transmitted and reflected signals. The object's velocity can be determined by analysing the frequency difference between the transmitted and received waves, a phenomenon known as the Doppler effect [13].

In traditional 3D radar systems (range, Doppler velocity and azimuth angle), the antennas are arranged only along a two-dimensional plane. As a result, targets are detected only in 2D horizontal coordinates, without vertical height information (elevation). With recent advances in radar technology, 4D radar has become possible, with antennas arranged horizontally and vertically, enabling elevation measurements in addition to range, velocity, and azimuth [13].

2.2 Radar Processing

Building on the fundamental working principles of radar systems, this section details the signal processing. This step is crucial for extracting meaningful information from raw radar signals, such as target range and radial velocity. Radar signal processing schemes are generally categorised into analogue continuous wave (CW) and digital modulation schemes. However, analogue continuous waves are currently the most common due to their low transmit power requirements and cost-effectiveness [14]. The following focuses on Frequency Modulated Continuous Wave.

2.2.1 Frequency Modulated Continuous Wave

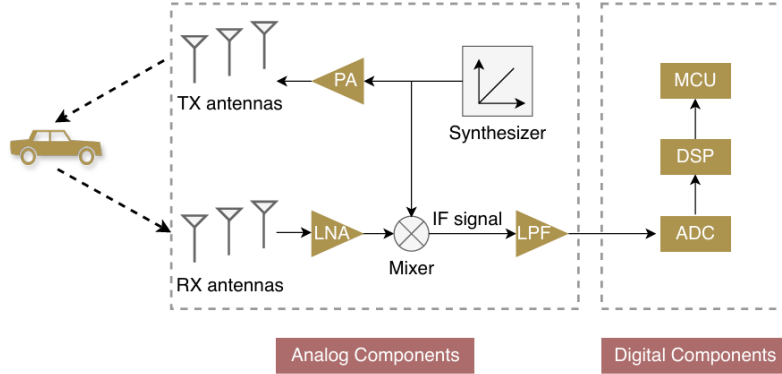


Figure 2.1: Block diagram of an FMCW radar system. The abbreviations used are as follows: PA (Power Amplifier), LNA (Low Noise Amplifier), LPF (Low Pass Filter), ADC (Analogue-to-Digital Converter), DSP (Digital Signal Processing module), and MCU (Micro Controller Unit). [13]

The core of FMCW is the generation of frequency chirps. When the radar starts working, the synthesiser first produces a linear frequency-modulated pulse, called a chirp, which is amplified by the Power Amplifier before being transmitted via the TX antenna. The chirp frequency increases linearly over a bandwidth B during a specific duration T_c . The echo is captured by the RX antenna and amplified with low noise. The mixer then combines the TX and RX signals to produce an Intermediate Frequency (IF) signal, subsequently digitised by the ADC. This IF signal, characterised by the beat frequency f_b , encodes the instantaneous frequency difference between the transmitted and received chirps and contains the information needed to detect and locate objects [13]. This process is illustrated in 2.2(a).

To obtain range, as illustrated in 2.2(c), the received signal is sampled for each chirp. After that, an FFT (Fast Fourier Transform) is applied to obtain a spectrum containing multiple peaks, each corresponding to objects located at different ranges. This operation is called "Range-FTT" [13].

To measure the Doppler velocity, the approach consists of transmitting two chirps separated by a time interval T_c . For each chirp, the spectrum obtained after Range-FFT peaks at the same range position but with different phases. The phase difference $\Delta\phi$ contains the object's velocity. When multiple objects are located at the same range, distinguishing their velocities becomes difficult because their peaks appear at the same position in the spectrum. To solve that, a sequence of N_c chirps is transmitted forming a "Sample-Chirp map", as illustrated in 2.2(d). A Doppler-FFT is then applied to the phasors obtained from the Range-FFT, enabling the separation of targets since the phase differences between consecutive chirps vary according to their velocities [13].

The azimuth angle is estimated using at least two RX antennas. The difference in the propagation paths from the target to each RX antenna introduces a phase shift in the FFT spectrum that corresponds to the Angle of Arrival (AoA). When multiple objects share the same range and velocity, additional RX antennas are needed to resolve them angularly, as shown in 2.2(b). The radar signal is thus described along a Sample-Chirp-Antenna dimension, as illustrated in 2.2(e). An Angle-FFT is then applied to the phasor sequences associated with the peaks obtained from the two-dimensional FFT processing (Range-FFT followed by Doppler-FFT), enabling separation of targets by azimuth angle. Similarly, this process can be applied to estimate the elevation angle [13]. To enhance angular resolution, modern radar systems employ a Multiple Input Multiple Output (MIMO) architecture, which uses N_{Tx} transmitters and N_{Rx} receivers to create a larger virtual array. As an example, the RADIAL [15] dataset combines 12 transmitting and 16 receiving antennas, yielding in $16 \times 12 = 192$ virtual antennas. This architecture, however, affects the Doppler processing described above as the signature of each target appears N_{Tx} times in the Doppler spectrum, separated by the phase shift.

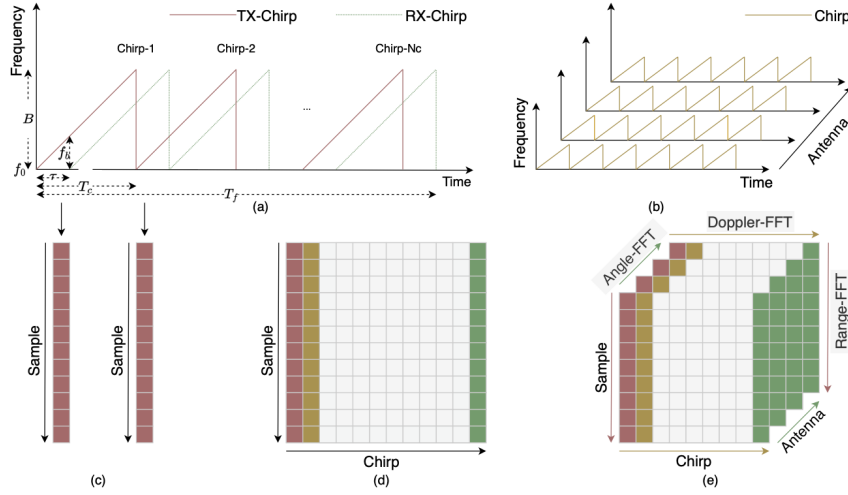


Figure 2.2: (a) Frequency of chirps transmitted by a TX antenna and received by an RX antenna. (b) Frequency of chirps received by multiple RX antennas. (c) Sampling is applied to each chirp. (d) Generation of a sample chirp map from the samples of all chirps. (e) Construction of a sample-chirp antenna tensor from the sample-chirp maps obtained across multiple RX antennas.[13]

2.3 Radar data representations

The signal processing pipeline described previously is essential for extracting physical quantities such as range, Doppler velocity, azimuth, and elevation angle from radar data. These quantities must then be structured into spatial representations, which are crucial for deep learning-based object detection.

Figure 2.3 illustrates a common approach pipeline for the generation of radar data representations. To support this discussion, all representations are derived from the

same sample of the RADial dataset, whose corresponding camera frame is shown in Figure 2.4.

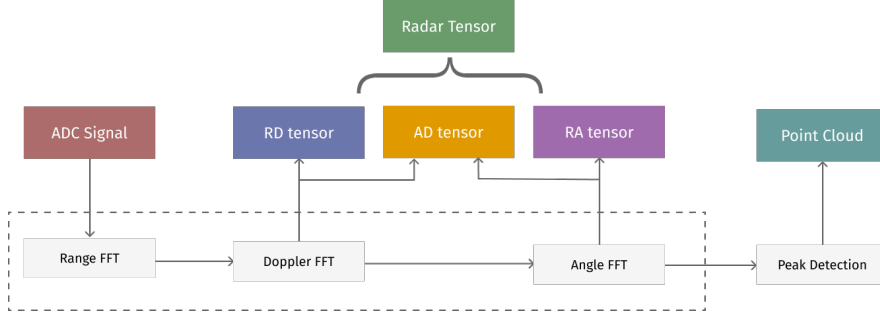


Figure 2.3: Pipeline for generation of radar data representations. The abbreviations used are as follows: RD (Range-Doppler), AD (Azimuth-Doppler), RA (Range-Azimuth).

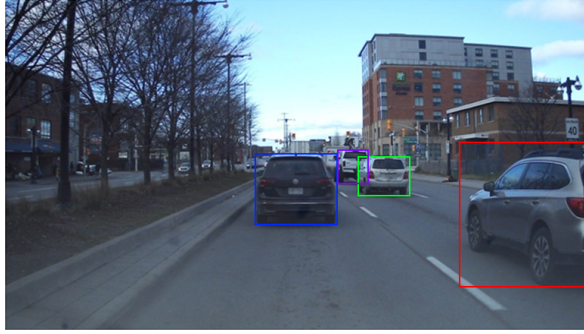


Figure 2.4: Camera frame used

2.3.1 ADC signal

Initially, the radar sensor produces raw data called an ADC signal, which is the digital representation of the analogue waveforms captured by the radar antenna. This signal is complex and high-dimensional, combining reflections from targets with noise and environmental clutter. Environmental clutter refers to unwanted returns from non-target objects in the scene, such as infrastructure or vegetation. ADC signal retains all information from the detections, which is valuable for deep learning applications. However, the signal lacks spatial coherence among its values, since all information is represented only in the time domain. It is also sensitive to factors such as electromagnetic interference, occupant positions, and obstructing objects, which require advanced signal processing methods to extract features [13] and make it difficult for humans to interpret it directly.

The RADial radar transmits 256 chirps per frame, each sampled at 512 points. The $\pm 1\text{M}$ samples visible on the x-axis are decomposed as $2 \times 512 \times 4 \times 256 = 1048576$ samples. Factor 2 accounts for the two components of a complex number, namely its real and imaginary parts. The factor 4 reflects the 4 RX antennas per channel. The

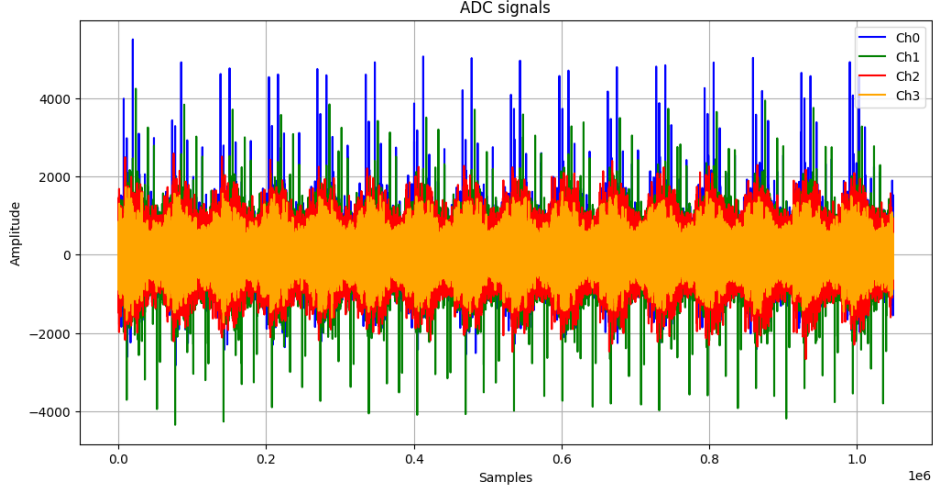


Figure 2.5: Raw ADC signals recorded by the four receive channels of the RADial radar over the current frame.

y-axis represents the raw ADC output as a signed 16-bit integer code. As expected from a raw time-domain signal, no high-level semantic information, such as the presence of cars, can be inferred directly from this representation.

2.3.2 Radar tensor

To address this, researchers utilise 3D FFTs across the sample, chirps, and antenna dimensions to convert the ADC signal into an image-like representation called 3D radar (RAD) tensor. This representation is often huge in memory and complex to manipulate. A common approach is to project the RAD tensor onto lower-dimensional subspaces, yielding the Range-Doppler (RD), Range-Azimuth (RA), and Azimuth-Doppler (AD) tensors. Each point in the tensor corresponds to a specific range, Doppler value, and angle, simplifying the visualisation and analysis of the radar returns.

Figure 2.6 illustrates the Range-Doppler map and Range-azimuth map obtained from the ADC signal. Due to the Multiple Input Multiple Output (MIMO) architecture of the RADial radar, the Doppler spectrum exhibits periodic replicas of each target reflection, spaced 12 bins apart along the Doppler axis. These reflections induce ambiguities, making it difficult to visually identify targets. However, we observe repeated peaks at approximately 7, 11, 22, and 34m, corresponding to the four vehicles in the camera frame.

In the Range-Azimuth map, the nearest vehicle on the right side of the camera frame appears at approximately -50° . The negative sign reflects the radar coordinate convention, in which targets to the right of the radar are assigned negative azimuth angles. A second peak is visible at 0° , corresponding to the vehicle directly ahead. Two other cars can be identified at approximately -20° and -15° , corresponding to the vehicles located at 22m and 24m, respectively.

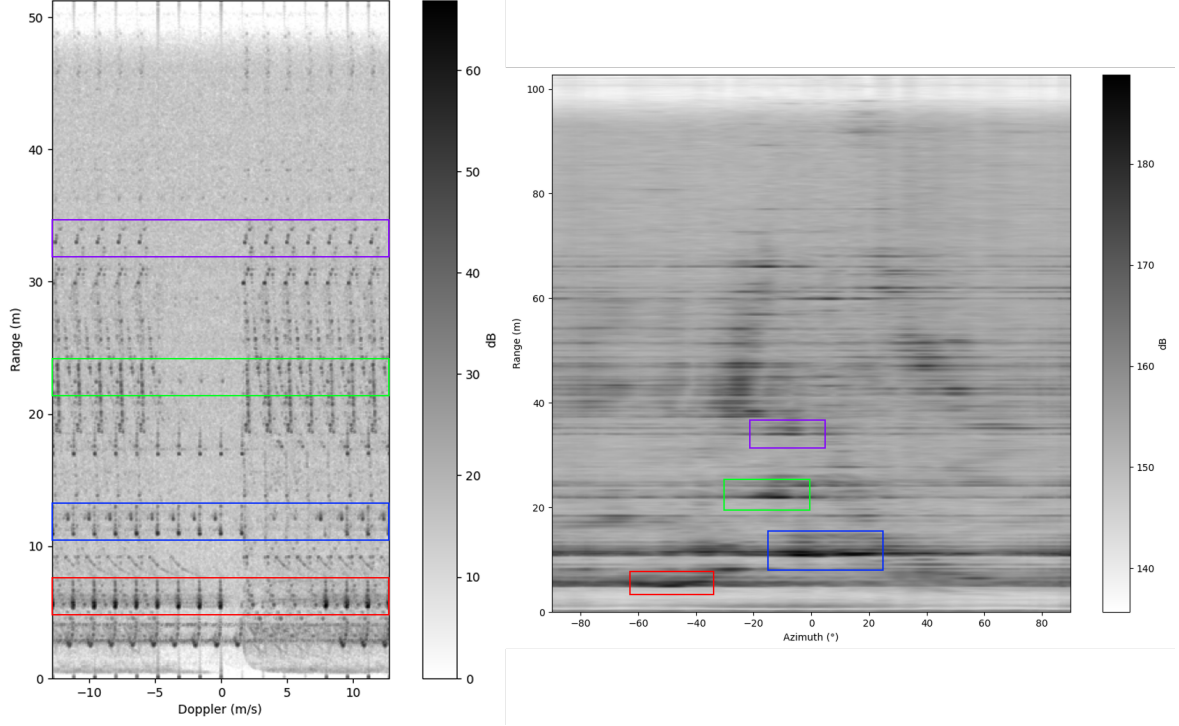


Figure 2.6: Range-Doppler map on the left and Range-Azimuth map on the right.

2.3.3 Point cloud

Peak detection is applied on the radar tensor to remove clutter, producing a sparse, point-like structure known as the point cloud. Each point cloud corresponds to a potential object in the environment, with its coordinates indicating the object's range, azimuth, and elevation (X, Y, Z). But also Doppler speed and reflected power. Point cloud representation provides a 3D spatial structure of the surrounding environment. Because radar point clouds are processed using CFAR, they benefit from reduced noise and a lightweight data structure. Obviously, some information contained in the raw data is lost during this process. Moreover, small objects or objects with weak reflections may fail to appear in the resulting point cloud.

In figure 2.7, a CFAR processing with a threshold of 2dB and an estimation window of 9x9 cells is applied to the Range-Doppler spectrum. Due to the MIMO configuration, the CFAR is applied on a reduced Doppler spectrum of size 512x16, obtained by summing the full 256-bin spectrum in groups of 16 to coherently combine the energy of each Tx replica. The effective Doppler resolution is therefore $16 \times 0.1 = 1.6 \text{ m/s}$, resulting in a sparser point cloud. However, we can now identify the targets in the Range-Doppler map. The three vehicles to the right of the ego-vehicle appear at approximately 1.6 m/s in the Doppler axis. Since the ego-vehicle is moving at approximately 7 m/s, static targets appear to be moving at approximately -6.4 m/s rather than 0 m/s. Static targets are approaching the radar as the vehicle moves forward. The offset from -7 m/s to -6.4 m/s is a direct consequence of the coarse Doppler quantisation on the reduced spectrum: $7/1.6 \approx 4$ bin and $4 \times 1.6 = 6.4 \text{ m/s}$. Finally, the vehicle directly ahead is

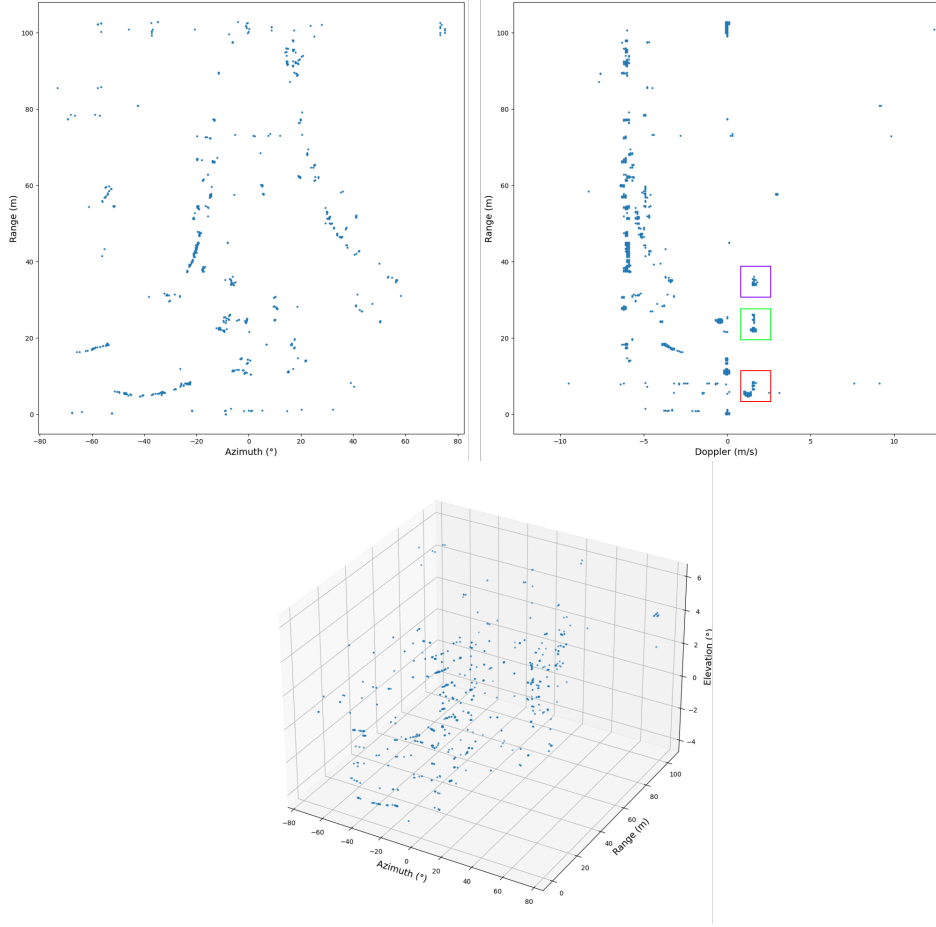


Figure 2.7: (a) Point cloud projected onto the Range-Azimuth plane, (b) point cloud projected onto the Range-Doppler plane, (c) three-dimensional point cloud representation in Range-Azimuth-Elevation space.

not detected by the CFAR because its SNR falls below the threshold. In Figure 2.7(a), individual targets are harder to isolate compared to the RD spectrum in 2.7(b), and it is further harder in three-dimensional space with added elevation 2.7(c)

2.4 Object detection task

Radar data representations constitute the raw input space on which perception algorithms operate. However, radar data representations alone are insufficient for autonomous driving applications as a system must be able to interpret this data, identify which regions of the scene correspond to physical objects, and assign them a semantic meaning. The following section formalises this task and discusses the specific challenges it raises when applied to radar data.

One approach to object detection involves estimating their bounding boxes and predicting their semantic classes. A general formulation of object detection can be

written as [16] :

$$\beta = f_{det}(\mathcal{I}_{sensor})$$

where $\beta = \{B_1, ..., B_n\}$ denotes the set of N objects in a scene, f_{det} represents the object detection model, and $\mathcal{I}_{sensor}$ corresponds to one or more sensor inputs. Object detection tasks can be divided into two categories: 2D and 3D. For this master’s thesis, the scope was limited to 2D object detection.

Given their structural similarity to images, radar tensors are well-suited to image-based detection architectures such as ResNet [17], Faster R-CNN [18], and Vision Transformers [19], which have been applied to both 2D representations (e.g., RA and RD tensors) and full 3D RAD tensors. However, unlike natural images, radar tensors lack a physical interpretation, making it non-trivial to transfer features learned in the image domain to radar data. Furthermore, applying algorithms to radar tensors in real-time applications remains challenging due to their high dimensionality and the presence of noise, interference, and clutter.

Having formalised the object detection task, we will now categorise object detection algorithms. The choice of detection algorithm is a critical design decision that directly conditions both accuracy and computational efficiency. Detection architectures can be categorised along two axes: the number of processing stages and the use of predefined anchor boxes. A two-stage detector (e.g, R-CNN) will first generate region proposals using selective search or a Region Proposal Network, and then perform object classification and bounding-box refinement on each proposed region. This method achieves high accuracy because the second stage refines the results, but it’s generally slower due to the two-step process. A single-stage detector (e.g., YOLO) directly detects objects in a single pass over the image, skipping proposal generation. It is typically faster and better suited to real-time applications, but may have lower accuracy than two-stage methods [20]. Anchor-based detection is an approach that places multiple predefined reference boxes (anchors) at each spatial location, then trains the network to classify which anchors contain objects and refine their coordinates. As opposed to anchor-free, where the network predicts directly from feature points [21]. For radar object detection, the choice between one and two-stage methods depends on the performance we want to achieve. However, for anchor selection, it is preferable to use anchor-free methods, as object sizes may vary considerably.

2.5 Tracking task

While object detection provides an instantaneous, comprehensive understanding of the scene at each time step, autonomous driving systems require a consistent understanding of the environment over time. This is the task of object tracking. The goal of object tracking is to associate detections across successive frames and estimate each object’s trajectory. Given our set of detection objects β_t at each time step t , the tracker needs to maintain a set of tracks $\mathcal{T} = \{T_1, ..., T_n\}$, where each track T_i encodes the state of an object, including its position and velocity.

Depending on the number of targets to be tracked, object tracking can be divided into two categories: Single-Object Tracking (SOT) and Multi-Object Tracking (MOT). SOT initialises a target model from a manually annotated first frame, then matches this model against subsequent frames using similarity measures. To handle appearance changes due to pose or environmental variations, the target model is updated online at each frame. MOT introduces additional challenges compared to SOT, including object occlusion, appearance similarity, and data association across an unknown number of targets. The dominant MOT paradigm is Tracking-by-Detection, which consists of a first detection step followed by an association step [8].

In the radar domain, tracking is essentially a state estimation problem, with filtering techniques such as the Kalman filter [22]. Radar MOT can be divided into two modes: Detection-Based tracking, which generates detections per frame before constructing trajectories, and Detection-Free Tracking, which operates directly on raw multi-frame measurements, offering better performance for low-SNR targets at the cost of higher computational complexity [8].

2.6 Autonomous Driving datasets

Having established the object detection and tracking tasks that an autonomous driving system must perform, this section examines radar datasets for these tasks. These datasets are specifically designed for autonomous driving applications, capturing complex real-world driving scenarios and providing multimodal sensor data suitable for perception, detection, and decision-making tasks.

Liu et al. (2024) [23] studied more than 265 autonomous driving datasets. Although the number is vast and growing, several limitations remain within this domain. First, the inherent complexity of these datasets arises from the requirement to integrate and synchronise multiple sensors, including cameras, radars, and LiDARs. Second, the annotation process is often time-consuming and error-prone. Labelling the nuScenes dataset took about 7,937 hours and cost 100 000 \$. Second, inconsistencies such as missing or inaccurate bounding boxes still appear in KITTI datasets [24][23]. Third, a geographical imbalance may introduce bias into the data. A significant proportion of datasets originates from a limited number of regions, notably the United States (20.94%), Germany (12.57%), and Europe (12.57%). This means a system trained in Germany might fail to recognise objects common in other cultures, such as the high density of electric scooters in China. Finally, class imbalance constitutes an additional challenge inherent to driving scenarios. For example, TJ4RadSet [25] has more than 15,000 cars annotated, compared to 6,000 trucks, 8,000 cyclists, and 4,000 pedestrians. This disproportion can bias the model toward dominant classes and make learning underrepresented ones more difficult.

Each dataset is designed for specific applications, such as object detection, semantic segmentation, object tracking, pedestrian detection and trajectory prediction, to name just a few. In this master’s thesis, we focus specifically on datasets designed for object

detection and tracking tasks. Unfortunately, many of these datasets provide only a single representation of radar data. For example, RADIATE [26], CARRADA [27], RADDet [28] and K-RADAR [7] provide radar tensors, whereas nuScenes [29], RadarScenes [30], TJ4RadSet [25] and WaterScenes [31] focus on point clouds. RADICAL [32] is the first dataset providing raw ADC signal data, specialised for object detection. To enable comparisons across different radar data representations, datasets that provide multiple representations are required. This is the case with RADIAL [15], which offers not only ADC signals but also processed data derived from them, including radar tensors and point clouds.

2.6.1 RADIAL

The RADIAL [15] dataset contains 91 sequences, each about 1-4 minutes long, for a total duration of 2 hours. The sequences cover three types of driving scenarios: highway (38%), countryside (32%), and city driving (31%). In total, the dataset includes around 25 000 synchronised frames from three sensors: camera, LiDAR, and HD radar. Out of which 8,252 are annotated, covering 9,550 vehicles. Each vehicle annotation consists of two-dimensional bounding boxes defined in the image plane, along with their corresponding real-world distance, angle, and Doppler velocity relative to the sensor. In addition to object detection, the dataset also provides segmentation. Rather than providing semantic segmentation of individual objects, the dataset introduces the concept of free-space segmentation, which aims to identify the traversable area in front of the vehicle, that is, the region within which safe navigation can be guaranteed by separating it from obstacles such as pedestrians, other vehicles and infrastructure.

Vehicle detection labels were first generated automatically using supervision from the camera and LiDAR modalities. A RetinaNet model [33] was used to extract object proposals from the camera. These proposals were subsequently validated when both radar and LiDAR agreed on the object’s position from their respective point clouds. Finally, manual verification was performed to reject or validate the labels. [15]

The free-space annotations were generated automatically from camera images using a DeepLabV3+ [34] model, pre-trained on the Cityscapes dataset and fine-tuned on a small manually annotated subset of the RADIAL dataset for a two-class segmentation task: free space and occupied. The model segmented each video frame and obtained a segmentation mask, which was projected from the camera’s coordinate system to the radar’s coordinate system using known calibration. Frames with preexisting vehicle bounding boxes were excluded from the free-space mask.

Radar specifications

This section describes the sensor specifications of the high-definition radar. A comprehensive description of the remaining sensors is available in the official repository. <https://github.com/valeoai/RADIAL>. The RADIAL HD radar employs a MIMO antenna configuration of 16 receiving and 12 transmitting antennas, resulting in $16 \times 12 = 192$ virtual antennas. This large virtual array is what qualifies the sensor as high-definition,

achieving an azimuth resolution of 0.1° and an elevation resolution of 1° . The 180° azimuth field of view provides full-frontal coverage of the driving scene, while 103m and 0.2m ranges and resolutions offer sufficient spatial precision for vehicle detection at highway speeds. The velocity resolution of 0.1m/s enables fine-grained Doppler separation between targets. One notable constraint is the 5 fps frame rate, which limits temporal resolution. Table 2.1 summarises the complete sensor specifications.

HD Radar		
FOV	Range	103 m
	Azimuth	180°
	Elevation	12°
Resolution	Range	0.2 m
	Azimuth	0.1°
	Elevation	1°
	Velocity	$0.1 \text{ m}\cdot\text{s}^{-1}$
Frame rate		5 fps
Height above ground		80 cm

Table 2.1: Main specifications of the HD radar used in the RADial dataset.¹

Justification of the selected RADial dataset

The selection of RADial as the primary dataset for this master’s thesis is motivated by the fundamental principle that robust tracking presupposes accurate detection, and accurate detection presupposes effective processing of the radar signal. Addressing the full chain requires a dataset that exposes multiple radar data representations simultaneously. RADial is, to the best of our knowledge, the only public dataset that provides raw ADC signals alongside RAD tensors and point clouds, enabling a principled study of how representational choices impact detection performance. Even though RADial presents some constraints, such as the number of annotated frames and the class imbalance included in this dataset. However, one aspect of this thesis was to investigate whether a transformer-based architecture can effectively process radar representations for object detection. In this context, a focused, well-understood dataset with multiple representations is more valuable than a larger dataset with a single representation. The absence of tracking annotations initially reflects the prioritisation of establishing robust object detection.

2.7 Existing radar representations and architectural approaches

Now that we have selected the dataset, this section surveys deep learning architectures proposed to address object detection across radar data representations. For each

¹<https://github.com/valeoai/RADial>

representation, we introduce state-of-the-art approaches to situate the broader landscape. Particular attention is given to FFT-RadNet and T-FFTRadNet, which operate on radar tensors and serve as the direct architectural baseline for this master’s thesis. Unless stated otherwise, each architecture is evaluated on the dataset for which it was originally designed

2.7.1 ADC signal

Two main directions have emerged from the literature. The first consists of designing individual end-to-end learnable radar perception models, such as ADCNet [35], T-FFTRadNet [36] and CubeLearn [37]. This approach applies deep learning to ADC, offering an alternative to traditional signal-processing procedures and effectively replacing computationally expensive FFT-based preprocessing steps. The second focuses on improving the resolution of radar data, which is essential for accurate object localisation. For example, the ADC Super-Resolution model (ADC-SR) [38] exploits raw ADC signals to improve azimuth resolution by predicting signals from unseen receivers. The resulting hallucinated ADC signals can be used to produce higher-quality RAD tensors, enhancing object detection performance.

End-to-end ADC-based architectures demonstrate competitive detection performance. However, a key limitation of operating directly on raw ADC signals is their sensitivity to electromagnetic interference and obstruction, which requires advanced signal processing to extract features. [13]

2.7.2 Radar tensor

Radar tensor is categorised into 2D,3D and 4D representations. 2D tensors include Range-Azimuth, Range-Doppler, and Azimuth-Doppler maps, while 3D tensors combine all three dimensions into a full RAD cube. 4D tensors further extend this representation by incorporating elevation information.

2D tensor-based methods apply image-based network architectures, such as ResNet [17], Faster R-CNN [18], or U-Net [39], directly to RA or RD maps. By operating on a simplified representation, these methods reduce computational complexity and exploit correlations between two dimensions to improve detection and localisation performance. For example, we can cite DAROD [40], which takes a Range-Doppler map as input and employs a Faster R-CNN backbone with a Region Proposal Network to identify regions in the spectrum that may contain targets. However, such methods exploit only a subset of the available spatial information, which may limit accuracy in complex scenarios involving multiple objects or occlusions [13].

3D tensor-based methods process range, Doppler and azimuth simultaneously, providing richer spatial information to the network. Major et al. [41] demonstrated the effectiveness of a deep learning-based object detection that operates on RAD tensor and proved that the Doppler dimension helps increase detection performance [13]. 4D tensor-based methods offer the most complete representation. K-Radar [42] compared

the performance of the 3D RAD tensor with the 2D RA tensor, demonstrating the importance of elevation information in radar perception. However, it comes at the cost of significantly higher computational complexity and memory requirements.

Despite their richness, radar tensors retain noise and clutter alongside object information, which limits the ability to capture subtle object characteristics. Furthermore, 3D and 4D tensors require considerable memory and bandwidth, posing challenges for real-time deployment on embedded systems.

2.7.3 Point cloud

A radar point cloud is sparse, providing only a coarse indication of an object’s location and is unable to describe the object’s outline. Point cloud-based methods can be categorised into three groups: clustering-based, point-based, and voxel-based.

Clustering-based methods group radar points into clusters using algorithms such as DBSCAN and perform classification based on spatial attributes such as position, velocity, and RCS. While computationally lightweight and memory-efficient, these methods are sensitive to parameter settings and lack the representational capacity of deep learning approaches, limiting their performance in complex scenes.

Point-based methods process radar point clouds at the individual point level, extracting spatial features directly from raw coordinates. Architectures such as PointNet [43], originally developed for LiDAR data, have been adapted to radar data by applying learned convolutional filters to individual points and treating spatial coordinates as integral features within the network. Point-based methods offer fine-grained localisation and classification capabilities, but they remain susceptible to noise and occlusion, which degrade feature-extraction accuracy.

Voxel-based methods discretise the 3D space into a regular grid of voxels, transforming a point cloud into a volumetric representation that can be processed by 3D convolutions or sparse convolutional neural networks, such as VoxelNet [44] or SECOND [45]. This structured representation facilitates efficient feature extraction but introduces quantisation errors during voxelisation, leading to a loss of fine-grained spatial information. Furthermore, the limited resolution of voxel grids reduces detection accuracy for small or distant objects.

Overall, while point clouds offer an intuitive 3D spatial structure and benefit from noise filtering techniques, they inevitably discard a large portion of the information present in the raw signal. Objects with weak reflection or small physical cross-sections may not produce sufficient returns to be rendered as point clouds. Moreover, most radar point cloud detection algorithms are currently adapted directly from LiDAR-based methods, which were designed for dense point clouds that capture object outlines. Radar point clouds are considerably sparser, making such direct adaptations suboptimal.

For object detection tasks, point cloud-based approaches remain less used than tensor-based methods in the radar literature. The sparsity of radar point clouds means that many objects produce too few returns to be reliably detected, leading to low recall regardless of localisation accuracy. This is confirmed by PIXOR [46] evaluated on RADial, which achieves a strong precision but a low recall, leading to an inferior F1 score compared to tensor-based approaches. Detailed performances are reported in Section 3.5.4. For these reasons, this thesis focuses on Range-Doppler tensor and raw ADC signal representations, which retain richer spatial and spectral information and have demonstrated stronger detection performance.

2.8 Comprehensive Study of FFT-RadNet and T-FFTRadNet

Building on the architectural landscape established in the previous section, this section provides a comprehensive study of FFT-RadNet and T-FFTRadNet, the two architectures that form the baseline of this thesis.

2.8.1 FFT-RadNet

Introduced by Rebut et al.(2022) [15] alongside the RADial dataset, FFT-RadNet is a deep learning architecture designed for efficient HD (high-definition) radar perception. The main motivations for this approach were to avoid the high processing overhead required to generate RA representations and the memory usage required to store the full RAD tensors. The architecture operates directly on the RD spectrum produced by the 16 receiver antennas. A trainable MIMO pre-encoder jointly processes the multi-channel RD data, reorganising and compressing the antenna contributions into a compact representation suitable for angle estimation. This representation is then fed into a shared feature pyramidal network (FPN) to extract multi-scale features. These features are then used in a range-angle decoder to build a range-azimuth latent representation. Finally, this latent representation is processed by a dedicated detection and segmentation head for multi-task learning. Figure 2.8 illustrates the architecture pipeline.

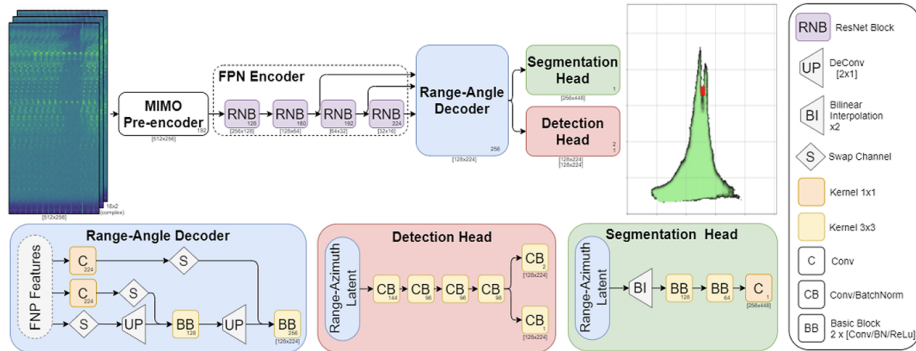


Figure 2.8: FFT-RadNet architecture

The MIMO pre-encoder constitutes the central component of the architecture. In a MIMO radar, when a signal reflects off an object, that object’s signature appears N_{Tx} times, one per transmitter, for every receiver. While the actual radial distance R remains the same, these signatures are captured at different Doppler velocities (D) due to a phase shift (Λ) induced in the transmitted signal. Specifically, the object is measured at the coordinates $(R, (D + k\Lambda)[D_{max}])_{k=1, \dots, N_{Tx}}$, where D_{max} is the maximum measurable Doppler value. To enable efficient angle estimation, the MIMO pre-encoder first reorganises the RD spectrum using Atrous convolution. This layer is used to gather Tx and Rx information at the right Doppler positions. The size of its kernel is $1 \times N_{Tx}$, and its dilation amounts to $\delta = \frac{\Delta B_D}{D_{max}}$, where B_D correspond to the number of Doppler bins. A second 3x3 convolution layer then learns to combine these receiver antenna channels, effectively compressing the signal into a more compact and meaningful representation.

The compact representation produced by the MIMO pre-encoder is then fed into a Feature Pyramidal Network (FPN). The goal of this backbone is to extract a feature pyramid through successive convolutions, yielding multiple resolutions. The architecture uses four blocks. The lower-level blocks maintain high spatial resolution to preserve small-object signatures, whereas the higher-level blocks capture more complex semantic information at a lower resolution. These four blocks are composed of 3, 6, 6, and 3 residual units, based on Bottleneck blocks. Bottleneck blocks are residual blocks that are composed of 3 convolutional layers. Figure 2.9 illustrates the bottleneck block as described in [47].

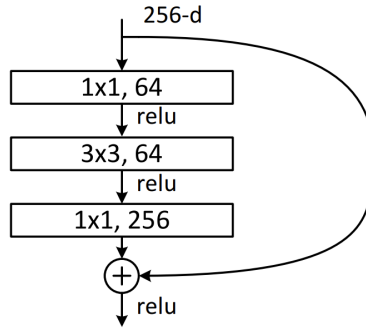


Figure 2.9: Bottleneck block [47]

The four multi-scale feature maps are subsequently processed by a Range-Angle decoder to produce a high-resolution representation. This upscaling is achieved through multiple deconvolution layers, whose outputs are combined with previous feature maps to preserve spatial details, producing a range-azimuth latent representation. The input tensor dimensions correspond to range, Doppler, and azimuth angle. Since the feature maps sent to the subsequent task should correspond to the Range-Azimuth representation, they swap the Doppler and azimuth axes to align with the correct dimensions.

The resulting Range-Azimuth latent representation is passed to a two-branch detection head and processed using a first common sequence of four Conv-BatchNorm layers with 144, 96, 96 and 96 filters, respectively. The branch is divided into confidence and regression branches. The confidence branch uses a convolutional layer with a sigmoid activation to generate a probability map. Each pixel determines if it is occupied by a vehicle. To reduce computational complexity, a coarse Range-Azimuth map is predicted, with each cell having a resolution of $1/4$ and $1/8$ of the native range and azimuth resolutions, respectively. The regression branch focuses on precision, using a single 3×3 convolution layer to produce two feature maps that yield refined range and azimuth estimates for detected objects.

In parallel to detection, a dedicated segmentation head operates on the same Range-Azimuth latent representation to identify free driving space. The Range-Azimuth latent representation is first interpolated to reach the target resolution, then processed through two consecutive stages of Conv-BatchNorm-ReLU blocks, producing 128 and 64 feature maps, respectively. A final 1×1 convolution produces a 2D feature map, followed by a sigmoid activation that estimates the probability that each location is drivable.

Overall, FFT-RadNet demonstrates that operating directly on the Range-Doppler spectrum, rather than on pre-computed RA or RAD representation, yields a computationally efficient pipeline without sacrificing detection performance.

2.8.2 T-FFTRadNet

FFT-RadNet takes the Range-Doppler map as its network input. Even if this representation is less costly than generating a Range-Azimuth tensor, it still involves pre-processing overhead, as it requires several successive Fourier transforms followed by a normalisation step. To overcome these costs, Giroux et al. (2023) [36] proposed a neural network capable of operating directly on raw ADC signals via the utilisation of complex transformation layers. They also introduced hierarchical Swin Vision transformers into the radar object detection literature. Figure 2.10 illustrates the architecture pipeline.

Conventional DFTs have some inherent computational costs and require a normalisation step before the representation can be fed into a network. Fourier-Net operates directly on raw, non-normalised time-domain signals and learns a transformation suitable for the task. Zhao et al. [37] introduced a complex-valued linear layer that incorporates prior knowledge of the Fourier transform structure into their design. Complex-valued linear layers (CPLX) consist of an $M \times M$ matrix, where each column acts as a learnable transformation over a slice of input. First, ADC signals are processed through a range-transformation layer, followed by a Doppler-transformation layer. Permutations (P) between these layers ensure the multiplications are correctly oriented to mimic a DFT over the respective axes. The weights of these layers are initialised with values corresponding to a standard DFT. The weights are also shifted to center zero Doppler velocity.

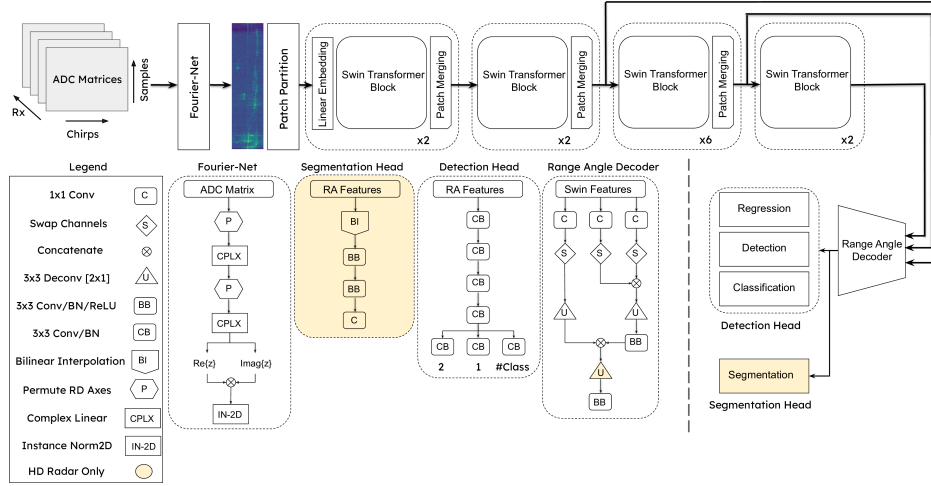


Figure 2.10: T-FFTRadNet architecture

The Swin Transformer constructs a hierarchical feature representation at varying resolutions, which is fed to the network backend, the same as that used in FFT-RadNet. On the front end, it splits an input image into small, non-overlapping patches. As the network gets deeper, it uses patch-merging layers to combine neighbouring patches, reducing the number of tokens and increasing the receptive field. This is performed using 2×2 patches, smaller than 3×3 convolutional windows. In Swin blocks, self-attention is computed in local neighbourhoods. The Swin Transformer maintains linear complexity relative to image size, avoiding the quadratic overhead of standard self-attention transformers that compare all patches globally. To maintain a broader context, consecutive layers utilise shifted windows that overlap the boundaries of the previous layer, establishing connections between neighbouring windows.

On the RADial dataset, T-FFTRadNet achieves a higher recall and F1 score than FFT-RadNet, while FFT-RadNet retains an advantage in precision and range error, suggesting that Transformer-based architectures offer a competitive and complementary alternative to CNN-based approaches. Detailed performance figures are reported in Section 3.5.4 alongside the contributions of this master's thesis.

These two architectures are selected as the architectural baseline of this master's thesis for three reasons. First, both were originally developed and evaluated on the RADial dataset, enabling direct comparison with the contributions presented in the following chapter. Second, they constitute a controlled CNN-versus-Transformer comparison within an identical pipeline. Third, T-FFTRadNet supports both ADC and RD inputs within the same network architecture, enabling comparisons between raw and preprocessed radar representations under identical experimental conditions.

Chapter 3

Radar object detection and tracking through Transformers

Building on the literature reviewed in the previous chapter, this chapter presents the first main contribution of this thesis. We begin by introducing Vision Transformers and discussing their limitations for dense prediction tasks, before motivating the adoption of a Transformer-based approach for radar object detection. RadViT architecture is then described in detail, followed by a discussion of the key design choices and explored alternatives. This chapter continues with an experimental evaluation covering both detection and classification performance. Finally, a Kalman filter-based multi-object tracking pipeline is presented, along with its evaluation results.

3.1 Vision Transformers

Inspired by the success of Transformers architectures in natural language processing, Vision Transformers (ViT), introduced by Dosovitskiy et al. [19], extend this paradigm to image understanding. Rather than processing images through convolutional operations, which rely on sliding a local filter over a spatial neighbourhood, ViT treats an image as a sequence of non-overlapping patches. Each patch is flattened and projected into a fixed-dimensional embedding via a linear layer, producing a sequence of tokens that is then fed into a standard Transformer encoder. These methods allow the model to capture long-range spatial dependencies from the earliest layers, in contrast to CNNs, in which global context is only progressively aggregated through stacked local operations.

3.1.1 Original Vision Transformers

The standard Transformer architectures operate on 1D sequences of token embeddings. For 2D images, the input image $x \in \mathbb{R}^{H \times W \times C}$, is reshaped into a sequence of N flattened 2D patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where (P, P) denotes the patch resolution and $N = HW/P^2$ is the resulting number of patches. Each patch is then projected into a D -dimensional embedding space via a trainable linear layer, ensuring a fixed dimensionality across all model layers. Figure 3.1 illustrates the working principle of Vision Transformers.

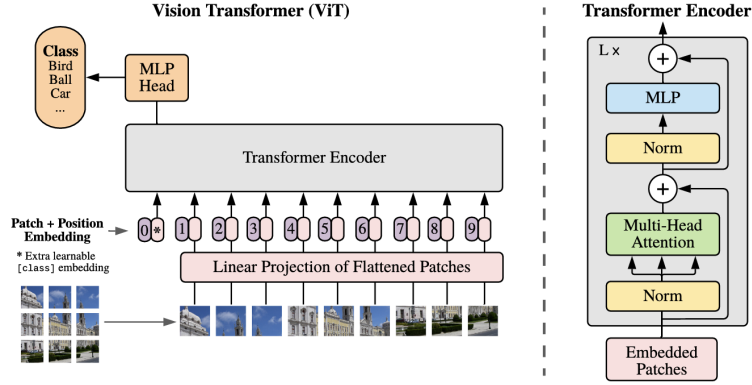


Figure 3.1: ViT architecture overview

Similar to BERT [48], they add a learnable `[class]` token into the sequence of embedded patches. This special token does not correspond to any image region. Rather, it aggregates global contextual information across all patches through the self-attention mechanism. Its final state serves as the image-level representation that is fed to the classification head.

Since the self-attention operation is permutation-invariant, spatial information would be lost. To address this, a standard learnable 1D position embedding is added to the patch embedding. The resulting sequence is passed through a Transformer Encoder, consisting of alternating blocks of Multi-Head Self-Attention (MHSA) and Multi-Layer Perceptron (MLP) layers. Layer normalisation (LayerNorm) is applied before each block, and residual connections are added after each block, following the pre-norm formulation introduced in [49].

Multi-Head Self-Attention (MHSA), introduced by Vaswani et al. [50], extends the self-attention mechanism by performing k self-attention operations in parallel, enabling the model to jointly attend to information across different representation subspaces.

For each token $\mathbf{z}_i \in \mathbb{R}^D$ in the input sequence, three vectors are computed through independent linear projections: a Query $\mathbf{q}_i = W^Q \mathbf{z}_i$, a Key $\mathbf{k}_i = W^K \mathbf{z}_i$, and a Value $\mathbf{v}_i = W^V \mathbf{z}_i$, where $W^Q, W^K, W^V \in \mathbb{R}^{D \times d_k}$ are learnable weight matrices. The Query represents the current element focus information, the Key represents the elements being matched against, and the Value carries the information to be aggregated.

Packing the full input sequence into a matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, the attention computation over the entire sequence can be computed by matrix multiplications:

$$\mathbf{Q} = \mathbf{X}W^Q, \quad \mathbf{K} = \mathbf{X}W^K, \quad \mathbf{V} = \mathbf{X}W^V \quad (3.1)$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} \right) \mathbf{V} \quad (3.2)$$

where the scaling factor $\sqrt{d_k}$ prevents the dot products from growing too large, which would push the softmax into regions of very small gradients.

Multi-Head Self-Attention is basically the concatenation of multiple attention heads. The outputs of all heads are then concatenated and projected back to the original dimensionality through a learnable matrix $W^O \in \mathbb{R}^{hd_k \times D}$: formulated as:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (3.3)$$

$$\text{where } \text{head}_i = \text{Attention}(\mathbf{X}W_i^Q, \mathbf{X}W_i^K, \mathbf{X}W_i^V) \quad (3.4)$$

This parallel structure allows each head to specialise in capturing different types of relationships between tokens, such as local versus global dependencies or different semantic aspects of the input.

Each attention block is followed by a Multi-Layer Perceptron (MLP) block, applied independently to each token. It consists of two linear layers with a GELU activation function.

LayerNorm [51] is applied before each block, following the pre-norm formulation. When combined with residual connections, this stabilises training and mitigates vanishing-gradient issues in deep networks. The full encoder block can thus be written as:

$$\mathbf{z}'_\ell = \text{MHSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1} \quad (3.5)$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell \quad (3.6)$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \quad (3.7)$$

where $\mathbf{z}_\ell$ denotes the output of the ℓ -th encoder layer.

3.1.2 Adapting ViTs for dense prediction

The original ViT is perfectly suited for image classification, where a single global representation such as the `[class]` token is sufficient. For the object detection task, we need spatially rich feature maps at multiple scales because objects can vary in size. The standard ViT processes all patches at a fixed resolution throughout the encoder, producing a single-scale feature map at the output. This constitutes a fundamental limitation for detection, as small objects may not be captured at a coarse patch resolution, and large objects require sufficient global context. Convolutional architectures naturally address this through Feature Pyramid Networks (FPN) [52], which aggregate features from multiple stages of the backbone at different spatial resolutions. Several adaptations have emerged in the literature to address this limitation. Hierarchical Swin Transformer [53], already discussed in section 2.8.2, introduces multi-scale processing directly within the Transformer by progressively merging tokens across stages. Another solution is proposed by Ranftl et al. [54], who recover spatial structure through a convolutional decoder that reassembles the flat token sequence into image-like structure at various resolutions.

To conclude, when applying ViTs to detection, a common challenge is recovering spatial structure and multi-scale information that are absent from the flat, single-resolution token sequence produced by the original architecture.

3.1.3 Motivation for a ViT-based approach in Radar perception

Vision Transformers have demonstrated strong capabilities in image understanding, particularly in capturing long-range spatial dependencies through the self-attention mechanism. This property is especially relevant in the context of radar perception, where objects of varying sizes must be detected across a wide field of view. For instance, in a Range-Doppler representation, a single vehicle can produce multiple signatures at different Doppler bins due to micro-Doppler contributions from its wheels and body, and the spectral signatures of two nearby objects can overlap and interact. Unlike CNNs, which build global context only progressively through stacked local operations and may therefore struggle to capture such non-local interactions, ViTs attend to the entire token sequence from the first encoder layer, making them inherently better suited to relate distant spatial regions without relying on network depth.

Moreover, in 2.8.2, we discussed that T-FFTRadNet with a Swin Transformer backbone achieves better performance than FFT-RadNet with a CNN and FPN backbone on the RADIAL dataset. These considerations motivate the development of RadViT which is described in the next section.

3.2 RadViT: A Vision Transformer for radar object detection

This section describes one of the main contributions of this thesis: RadViT, a Radar Vision Transformer designed for object detection on radar data. Rather than adopting the hierarchical Swin Transformer used in T-FFTRadNet, a plain ViT backbone is deliberately chosen for two reasons. First, it provides a simpler and more interpretable baseline, avoiding confounding effects introduced by the multi-scale inductive biases of hierarchical architectures. Second, it enables a fair and controlled comparison between input representations, ensuring that any observed performance differences can be attributed to the choice of representation itself rather than to architectural factors. Specifically, this thesis compares raw ADC signals and Range-Doppler maps within the same architectural network. Point cloud representations were deliberately excluded, as they require a fundamentally different network architecture and have been shown to yield inferior detection performance due to their inherent sparsity, as discussed in Section 3.5.4.

Beyond serving as a comparison tool, RadViT is designed to be largely radar-independent, meaning its core components are not tied to any specific radar configuration and can operate with both MIMO and SIMO systems. This is what distinguishes RadViT from existing architectures in the literature. To preserve this generality, the adaptations

required to accommodate a different sensor are intentionally localised within a single module of the architecture, as discussed in Section 3.2.5. Furthermore, RadViT is designed to exploit the multi-antenna structure of radar data.

Finally, RadViT extends the detection pipelines of FFT-RadNet and T-FFTRadNet with a dedicated classification head, enabling vehicle category prediction alongside confidence, range, and azimuth regression. To the best of our knowledge, none of the existing architectures evaluated on RADial incorporates such a classification branch, making this an additional contribution of this work.

3.2.1 Architecture overview

As we explained before, the network can take either raw ADC signals or Range-Doppler maps as input. When operating in ADC mode, the raw signal is first passed through the FFT-Net module from T-FFTRadNet 2.8.2, which mimics the action of a standard discrete Fourier transform to produce a Range-Doppler representation. From this point, both input modes follow a common processing pipeline as illustrated in Figure 3.2.

The Range-Doppler map is partitioned into non-overlapping patches and processed through a radar-adapted patch embedding module (Section 3.2.3). For each antenna, the real and imaginary parts are independently partitioned into patches, then concatenated, and projected into a fixed-dimensional embedding space via a trainable linear layer. Each antenna is processed independently by the same Transformer encoder block with shared weights, and the resulting token sequences are aggregated by averaging across antennas before being forwarded to the subsequent stages.

The aggregated encoder output is reshaped into a spatial feature map and passed through the multi-scale feature module (Section 3.2.5), which simulates a feature pyramid by deriving four representations at different spatial resolutions from the single encoder output, in the absence of a naturally hierarchical backbone. These are then fed into the Range-Angle Decoder (Section 3.2.6), which reconstructs a Range-Azimuth latent representation of the scene. Finally, the Detection Header (Section 3.2.7) produces three output channels: a confidence map, a regression map predicting range and azimuth offsets, and a classification map assigning a vehicle label to each detected object.

3.2.2 Radar input processing

Several signal preprocessing methods are applied to the Range-Doppler map. First, the direct-conversion (DC) offset is removed by subtracting the mean of the ADC signal. This offset arises from TX-RX antenna leakage (Section 4.2), which produces a near-zero beat frequency that manifests as an unwanted constant component in the baseband signal after mixing (Section 4.2). After that, a Hamming Window is applied to the range and Doppler dimensions to reduce the signal’s sidelobes. Additionally, an FFT shift is applied along the Doppler axis to centre the zero-velocity component at the centre of the spectrum. While this operation does not affect network training, centring

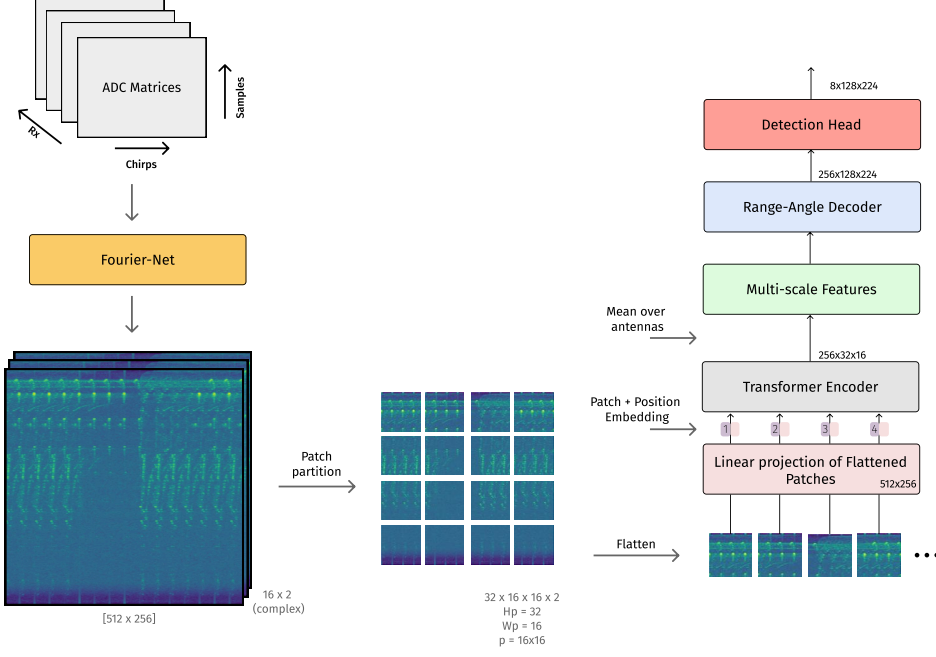


Figure 3.2: RadViT architecture: RD inputs replace the Fourier-Net with Range-Doppler FFTs.

the Doppler spectrum has been shown to yield better detection results (see Section 3.5.4). Finally, we normalise the range Doppler map channel by channel according to:

$$\hat{x}_i = \frac{x_i - \mu_i}{\sigma_i}$$

where μ_i and σ_i denote the mean and standard deviation of channel i , computed over the entire dataset. This per-channel normalisation ensures that no individual antenna channel dominates the learning due to amplitude differences, stabilising training and accelerating gradient convergence.

3.2.3 Radar-adapted patch embedding

Once we have preprocessed the Range-Doppler map, we can partition it into non-overlapping patches. For each antenna, the real and imaginary parts are extracted independently, and each is partitioned into 16×16 patches, leading to $Hp = 512/16 = 32$ patches along the range axis and $Wp = 256/16 = 16$ patches along the Doppler axis, for a total of 512 patches per antenna. The results are concatenated, producing a patch vector of dimension $2 \times 16^2 = 512$, which is projected into the D -dimensional embedding space via a trainable linear layer.

Positional encodings are then added to the resulting token sequence, following the sinusoidal formulation introduced by Vaswani et al. [50]. For each position pos in the sequence, a vector of dimension D is generated by alternating sine and cosine functions at varying frequencies:

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i/D}}\right)$$

$$PE_{(pos, 2i+1)} = \cos(\frac{pos}{10000^{2i/D}})$$

where i denotes the dimension index. Even dimensions receive a sine encoding and odd dimensions a cosine encoding.

3.2.4 Transformer Encoder

Once positional encoding has been applied, the resulting token sequence is passed through the Transformer encoder as illustrated in Figure A.1. As described in the previous section, the encoder processes each antenna independently using shared weights. This weight sharing reduces the number of learnable parameters and encourages the model to learn antenna-invariant representations, which contributes to the radar-independent design of RadViT.

The encoder follows the pre-norm formulation adopted by Dosovitskiy et al. [19], rather than the post-norm variant of the original Transformer [50]. Each encoder block first applies LayerNorm to the input x , then applies MHSA, and finally adds a residual connection. A second LayerNorm is then applied before the MLP block, and its output is added via a second residual connection. A dropout rate of 0.2 is applied after both the MHSA and MLP outputs to mitigate overfitting, which is particularly important given the limited size of the RADial dataset. The encoder stack consists of 6 successive blocks, with parameters $D = 256$, 8 attention heads, and an MLP hidden dimension of 1024, leading to a compact architecture. The justification for this choice is further discussed in Section 3.3.

Once all antennas have been processed, their respective output token sequences are averaged to produce a single unified representation, which is then reshaped into a spatial feature map with dimensions (256, 32, 16).

3.2.5 Multi-scale features pyramid

Because standard ViT lacks a hierarchical backbone that produces feature maps at multiple resolutions, we need to construct a simulated feature pyramid from the single encoder output to remain compatible with the Range-Angle decoder inherited from FFT-RadNet. This module is illustrated in Figure 3.3.

Four feature maps are derived through the following operations: the spatial feature map is upsampled along the range dimension by factors of 2 and 4 via bilinear interpolation, downsampled by a factor of 2 via average pooling, and kept at its original resolution. Each resulting feature map is then passed through a 1x1 convolutional projection to match the channel dimension expected by the Range-Angle decoder (e.g, 256, 160, 192 and 16), which will leverage this dimension to reconstruct the azimuthal representation. These dimensions were chosen by FFT-RadNet to encode the azimuth angle over the entire distance range, at best.

When using a sensor with a different Range-Doppler resolution, only this module needs to be adapted, as the interpolation factors and average-pooling kernel sizes depend directly on the input feature dimensions. All other components of RadViT remain unchanged. As a concrete example, when using the network with the K-MD2 radar (see Chapter 4), which produces 256×256 resolution Range-Doppler maps, the encoder output after 16×16 patch partitioning is already at the coarsest target resolution, eliminating the need for average pooling. Three bilinear interpolations are instead applied to progressively upsample to the resolutions expected by the Range-Angle Decoder, demonstrating that sensor adaptation requires only minimal modification to this single module.

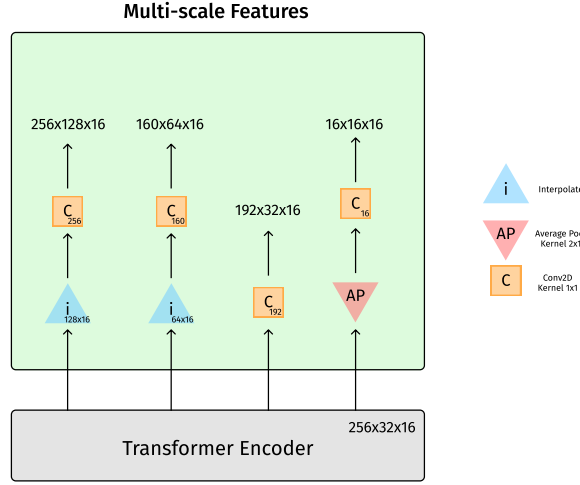


Figure 3.3: Multi-scale features block

3.2.6 Range-Angle decoder

The four feature maps are passed through the Range-Angle decoder, illustrated in Figure A.2, which reconstructs a Range-Azimuth representation from the pyramidal features. The first operation consists of a 1×1 convolutional projection that maps the channel dimension to 224, corresponding to a quarter of the full azimuth resolution ($896/4 = 224$). The channel and Doppler axes are then swapped via a transpose operation, transforming a tensor of shape $(B, 224, R, 16)$ into $(B, 16, R, 224)$, effectively converting the channel dimension into a spatial azimuth axis.

The deepest feature map, corresponding to the coarsest spatial resolution, is then progressively upsampled along the range axis through three transposed convolutions with a stride of $(2, 1)$, doubling the range dimension at each step, from 16 to 32, 64, and finally 128. At each upsampling step, the output is concatenated with the corresponding feature map from the pyramid via skip connections, before being refined by two Conv-BatchNorm-ReLU layers. The final output is a Range-Azimuth latent representation of shape $(B, 256, 128, 224)$, corresponding to the coarse prediction resolution of FFT-RadNet, a quarter of the full resolution in both range and azimuth, which was shown to be sufficient to dissociate two close objects [15].

3.2.7 Detection header

The Range-Azimuth latent representation is then passed through the Detection Head, which first applies a shared sequence of four Conv-BatchNorm layers that progressively reduce the channel dimension from 256 to 144, 96, 96, and 96. The resulting feature map is then split into three parallel output heads. The confidence and regression heads are already discussed in Section 2.8.1. The classification head, which is specific to this thesis, produces a five-channel map that assigns label categories to each detected object. The five classes correspond to car, truck, bicycle, bus and person, following the COCO [55] dataset taxonomy. The three outputs are concatenated along the channel dimension, leading to a final tensor of shape $(B, 8, 128, 224)$.

3.3 Design choices and explored alternatives

Building RadViT required several architectural decisions that are not immediately apparent in the final design. This section details the key design choices made during development, discusses the alternatives that were explored, and explains why certain directions were ultimately abandoned in favour of the adopted solution.

During development, a Multi-Channel Vision Transformer (MViT) was explored as an alternative, inspired by Hussein et al. [56]. The idea was to assign a dedicated encoder to each antenna, resulting in 16 independent encoders in total, allowing each to specialise in learning antenna-specific patterns. However, this architecture proved impractical on the RADial dataset due to the large number of parameters induced. Reducing the number of encoders from 16 to 1 did not improve generalisation, as overfitting persisted. The dataset size (6.231 training samples) was the primary limiting factor, as Transformer-based architectures are known to require a large amount of data to generalise effectively. To mitigate overfitting as much as possible, we adopted a deliberately compact architecture, as described in Section 3.2.4.

To address the overfitting observed during training, we investigated the implementation of Masked Autoencoders (MAE) [57]. This self-supervised learning consists of reconstructing the original image from partial observation, theoretically enabling the encoder to learn a robust spatial representation of the input without the need for manual detection labels. However, this approach proved more complicated than expected, likely because radar data is unstructured and lacks the evident semantic redundancy found in natural images. The refinement of this technique is left for future research direction.

The radar-adapted patch embedding was chosen over a classical convolutional patch embedding for two reasons. First, it explicitly preserves the complex-valued nature of radar by jointly encoding the real and imaginary components within each patch vector, allowing the model to retain phase information that would otherwise be discarded. Second, a classical convolutional patch embedding, which uses a convolutional layer with kernel and stride sizes equal to the patch size, was evaluated and found to perform worse than the radar-adapted patch embedding. On top of that, classical patch embedding is inherently tied to a specific tensor by fixing the number of input channels determined

by the sensor configuration, whereas the linear projection over concatenated real and imaginary patches adapts naturally to any radar system, thereby promoting radar independence.

Finally, an alternative architecture, RadViT-RD, was explored in which the Range-Angle Decoder was replaced by a Range-Doppler Decoder, preserving the Range-Doppler representation throughout the network. This variant performed significantly worse than the Range-Angle Decoder. The Range-Angle Decoder leverages both the velocity information from the Doppler dimension and the phase differences across the 16 receiver antennas to reconstruct a Range-Azimuth latent representation, combining multi-antenna contributions into a spatially meaningful representation in terms of range and azimuth. In RadViT-RD, while the 16 antennas are still processed by the encoder, the output is reconstructed as a Range-Doppler map, which encodes only range and velocity information. The azimuth dimension is never explicitly reconstructed, depriving the detection head of the spatial localisation information it needs to accurately localise objects.

3.4 Extending the RADial dataset for classification and tracking

Another contribution of this thesis is the extension of the RADial dataset to support both object classification and multi-object tracking. Indeed, RADial does not provide class labels, which are essential for complete object detection.

To address this, we employ a semi-automated labelling strategy that leverages the cross-modal synchronisation available in the RADial dataset. Each sample is initially labelled in the camera domain by computing IoU between ground-truth bounding boxes and detections produced by YOLO26x, and assigning the class label of the best-matching detection. However, this automatic procedure introduced a risk of errors when YOLO26x produces low-confidence or incorrect detections, which could propagate noise into the training data. Given the dataset’s small size, we subsequently perform a manual verification step to ensure label quality and consistency across all samples. Each annotated object is assigned to one of five predefined categories. Table 3.1 reports the number of instances per class. As shown, the dataset exhibits a strong class imbalance, as expected, since it was not originally designed for classification tasks.

Class	Number of instances
Car	7952
Truck	1633
Bicycle	6
Bus	4
Person	1

Table 3.1: Distribution of ground-truth classes in the RADial dataset.

Concerning multi-object tracking, extending the dataset required assigning persistent track identifiers to each annotated object across successive frames. An automatic approach was first considered, but proved infeasible because annotations were not provided for every frame, making it impossible to reliably compute IoU between bounding boxes over time and to assign consistent track IDs to objects. A manual annotation approach was therefore adopted. Since tracking evaluation requires frame-level continuity, the annotation effort is substantially greater than for detection. Therefore, only the test set was annotated, ensuring consistent evaluation conditions across the object detection and tracking tasks.

3.5 Experimental results

Having described the RadViT architecture, this section presents the experimental evaluation of the proposed model. We first describe the dataset split and label-encoding strategy, followed by the training configuration and evaluation metrics. Detection results are then reported and compared against state-of-the-art baselines, with an analysis of the input representation that compares ADC signals and Range-Doppler maps. Finally, classification performance is evaluated and compared against YOLO26n and YOLO26x.

3.5.1 Dataset split and label encoding

RadViT has been trained on RD maps and ADC signals. The dataset is split into Training (6.231 samples), Validation (986 samples) and Test sets (1.035 samples). The test set is composed of sequences that appear exclusively in that split, ensuring that no frame from the test set is ever seen during training. The dataset is manually split between easy and hard cases. Hard cases primarily involve challenging situations in which radar signals are perturbed, e.g., by interference from other radars, significant sidelobe effects, or strong reflections from metallic surfaces. Detection performance is evaluated on the full test set, with a detailed breakdown into easy and hard cases provided in the Appendix B.

Since RadViT produces a dense Range-Azimuth prediction map, the ground-truth labels must be encoded into a compatible spatial representation. Each annotated object is mapped onto a target tensor of shape $(4, R/4, A/4)$, where the four channels correspond to the confidence map, the range residual, the azimuth residual and the class identifier.

For each annotation, the range and azimuth values are first quantised to their respective bin indices at the coarse prediction resolution, matching the detection head’s resolution. The residual offsets between the exact object position and the bin centre are then normalised using the dataset’s precomputed mean and standard deviation. The confidence channel is set to 1 at the target location, and the class identifier is set accordingly. Rather than marking only the target bin, a 3×3 spatial spread is applied around it, activating a neighbourhood of 9 bins per annotated object. Each neighbouring bin receives a residual offset adjusted by its physical distance to the

true object position, such that every active bin points back to the same ground-truth location. This facilitates learning and improves robustness to small localisation errors during inference.

At inference time, the decoding process is applied. We select bins whose confidence exceeds a threshold and reconstruct the predicted range and azimuth by combining their positions with the decoded residual offsets. The predicted class is determined by taking the argmax over the classification channels at each detected bin.

3.5.2 Training configuration

The confidence map is trained using a FocalLoss with $\gamma = 2$, which addresses the severe class imbalance inherent to dense prediction maps, where the vast majority of pixels correspond to background and only a sparse set of locations contain objects. The regression map is trained using Smooth L1 Loss with $\beta = 1$, which is robust to outliers by applying an L2 penalty for small errors and an L1 penalty for large errors. The loss is summed over the batch. Finally, the classification head is optimised using a standard cross-entropy loss over the five vehicle categories.

The patch embedding, transformer encoder and range-angle decoder are trained from scratch, while the Detection Header is fine-tuned from T-FFTRadNet weights. The FFT-Net module is also initialised from T-FFTRadNet weights and further fine-tuned during training. This partially explains the overfitting discussed in Section 3.3. Training the ViT backbone from scratch on a limited dataset is particularly challenging, as Transformer-based architectures typically require large amounts of data to learn effective representations.

RadViT is trained on 100 epochs with a batch size of 4, using the Adam optimiser with a learning rate of 1×10^{-4} and a weight decay of 0.2. A linear warmup is applied over the first 10 epochs, after which the learning rate follows a cosine annealing schedule for the remainder of the training. Training was performed in a data-parallel fashion across three NVIDIA GeForce GTX 1080 Ti GPUs (12 GB each).

3.5.3 Detection metrics used

The metrics used here are Average Precision (AP), Average Recall (AR), F1 score (F1), range error (R) and angle error (A). Average precision is computed by evaluating precision at confidence thresholds from 0.1 to 0.95 by step of 0.1, then averaging the results. For each threshold, precision is defined as:

$$Precision = \frac{TP}{TP + FP}$$

where a prediction is counted as True Positive (TP) if its IoU with a ground truth bounding box exceeds 0.5 (mAP@50) and as False Positive (FP) otherwise. Prior to IoU computation, predicted objects are converted from Range-Azimuth coordinates to Cartesian bounding boxes using a fixed size of 4m x 1.8m. This conversion is applied

consistently to both predictions and ground truth annotations across all evaluations. Average Recall is computed by evaluating recall as for precision, then averaging the results. For each threshold, recall is defined as

$$Recall = \frac{TP}{TP + FN}$$

Finally, the F1 score is computed by taking the mean average precision and the mean average recall defined as:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

For each TP and each threshold, the range error is computed as the absolute difference between the ground-truth range and the predicted range, accumulated over all correctly detected objects, then averaged. The same procedure is applied to obtain the angle error. The optimal model is selected based on the highest F1 score.

3.5.4 Detection performances

Before analysing RadViT results, several observations can be drawn from the baseline methods reported in Table 3.2. T-FFTRadNet with a shift RD achieves the highest F1 score, reflecting a strong balance between precision and recall. However, FFT-RadNet still outperforms it in terms of precision and range error. The Shift notations denote centring the Range-Doppler map at zero Doppler velocity, while Windowing refers to applying a Hamming window. Both preprocessing steps improve performance. Using the raw ADC signal degrades performance, and introducing a non-linear activation (NL) in the Fourier network yields no performance gains.

Regarding input representations, point cloud-based PIXOR achieves high precision but very low recall, resulting in a significantly degraded F1 score. This confirms the limitations of radar point clouds discussed in Section 2.7.3. Their inherent sparsity results in substantial information loss, making this representation less suited for object detection than dense tensor representations. PIXOR using Range-Azimuth input achieves the second-highest AP score, suggesting that the RA representation benefits localisation precision. Even though a direct architectural comparison between PIXOR and (T-)FFTRadNet is not straightforward, RD-based methods consistently achieve higher overall performance, which can be attributed to the complementary nature of the two representations. While the RA maps provide spatial localisation, the Range-Doppler maps add velocity information, allowing two objects at similar range but travelling at different speeds to be distinguished. The Range-Angle Decoder effectively combines both by reconstructing a Range-Azimuth representation from the Range-Doppler spectrum.

Regarding RadViT, results are consistent with its design objective of comparing input representations rather than maximising detection performance. The Range-Doppler input outperforms the ADC signal by 1.9% in F1 score, a smaller gap than observed in T-FFTRadNet (2.4%). This suggests that fine-tuning the FFT-Net and the Detection Header module from T-FFTRadNet weights may partially compensate for

the performance drop associated with raw ADC signals. The performance gap between RadViT and T-FFTRadNet is likely attributable to the simulated feature pyramid, which lacks the true multi-scale representation produced by hierarchical backbones. Nevertheless, RadViT achieves competitive results within its design constraints, with F1 scores of 76.9% and 75.0% for RD and ADC inputs, respectively, and angle errors of 0.12° and 0.13°, which are comparable to T-FFTRadNet.

Model	Radar Input	AP(%) $\uparrow$	AR(%) $\uparrow$	F1(%) $\uparrow$	R(m) $\downarrow$	A(°) $\downarrow$
FFT-RadNet	RD	96.8	82.2	88.9	<u>0.11</u>	0.17
T-FFTRadNet	RD	89.6	89.5	<u>89.5</u>	0.15	0.12
T-FFTRadNet	RD - Shift	91.3	<u>88.3</u>	89.8	0.15	0.12
T-FFTRadNet	RD (No Windowing)	89.4	87.6	88.5	0.16	0.12
T-FFTRadNet	ADC	88.2	86.7	87.4	0.16	<u>0.13</u>
T-FFTRadNet	ADC - NL	88.1	86.1	87.1	0.16	<u>0.13</u>
T-FFTRadNet	ADC - No Shift	88.1	85.5	86.7	0.16	<u>0.13</u>
Pixor	PC	96.5	32.2	48.3	0.17	0.25
Pixor	RA	<u>96.6</u>	81.7	88.5	0.10	0.20
RadViT	RD - Shift	83.3	71.3	76.9	0.17	0.12
RadViT	ADC	81.0	69.9	75.0	0.19	<u>0.13</u>

Table 3.2: Detection performance on the RADial test set (overall cases). Best results per column are shown in bold, and the second-best results are underlined.

Another aspect in which the ADC signal and the RD tensor can be compared is the end-to-end inference latency, reported in Table 3.3. Unlike the RD pipeline, which requires a 2D FFT, an FFT shift, and a normalisation step prior to network inference, the ADC pipeline requires no preprocessing. As a result, the total pipeline latency is substantially lower in the ADC mode, by 133.23 ms, which may be significant in real-time deployment scenarios. The two pipelines have comparable network inference times (116.23 ms vs 118.00 ms), consistent with their near-identical computational cost of approximately 187 GFLOPs. The ADC pipeline introduces approximately 650K additional parameters relative to the RD pipeline, corresponding to the learnable weights of the internal Fourier-Net.

3.5.5 Classification performances

To evaluate classification performance independently of detection performance, we first analyse the confusion matrix, which reports the number of samples for each combination of predicted and ground-truth labels. The confusion matrix is computed only for true-positive detections, defined as predictions with confidence exceeding 0.5 and IoU with a ground-truth bounding box exceeding 0.5. By restricting the analysis to correctly localised detections, the impact of missed detections and localisation errors is excluded from the comparison, allowing the evaluation to focus solely on classification performance.

Step	ADC		RD	
	Mean (ms)	Std (ms)	Mean (ms)	Std (ms)
Load ADC	10.69	3.73	11.20	13.30
DC Removal	-	-	15.87	0.65
Range FFT	-	-	26.03	5.29
Doppler FFT	-	-	41.91	1.17
FFT Shift	-	-	23.79	0.80
Normalisation	-	-	23.48	0.53
CPU $\rightarrow$ GPU Transfer	14.77	1.36	14.55	1.39
Network Inference	116.23	1.70	118.00	2.78
Decode	0.31	0.09	0.40	0.12
Total	142.00 ms		275.23 ms	
FLOPs	187.2G		187.0G	
Params	7.86M		7.21M	

Table 3.3: Per-stage latency breakdown, FLOPs and parameters count for ADC and RD pipelines, averaged over the first 100 frames of the dataset.

This evaluation method is particularly important because the RADial dataset does not provide classification labels, so no radar-based baseline was available for this evaluation. Consequently, YOLO26 and, more specifically, its extra-large (YOLO26x) and nano (YOLO26n) variants, were selected as baselines, as they represent strong state-of-the-art object detectors against which classification performance can be compared.

However, comparing YOLO and RadViT remains challenging due to the domain mismatch between the two approaches. YOLO operates in the image domain, whereas RadViT predicts directly in the radar domain. To enable a fair comparison, YOLO bounding boxes are projected into Range-Azimuth coordinates using the camera calibration matrix provided by the RADial dataset. Specifically, the lower-midpoint of each bounding box is used as a reference point because it better approximates the object’s ground contact point. Using the geometric centre of the bounding box was found to overestimate the range because it lies above the ground plane.

The projected point is then converted to Cartesian coordinates and enclosed within a fixed bounding box of $4\text{ m} \times 1.8\text{ m}$ for IoU computation, thereby ensuring that both approaches are evaluated under the same geometric assumptions. Figure 3.4 presents the confusion matrices. We deliberately evaluate performance classification on car and truck classes as bicycles, buses and persons are underrepresented in the test set. Including these classes would yield statistically unreliable results and introduce significant bias into the evaluation. Table 3.4 reports the number of instances per class.

From Figure 3.4, we can observe several points from these results. All models struggle to correctly classify trucks. RadViT achieves a precision of 39.0%, while YOLO26n and YOLO26x reach 43.3% and 48.4%, respectively. For RadViT, this behaviour is

Class	Number of instances
Car	877
Truck	376
Bicycle	6
Bus	0
Person	1

Table 3.4: Distribution of ground-truth classes in the RADial test set.

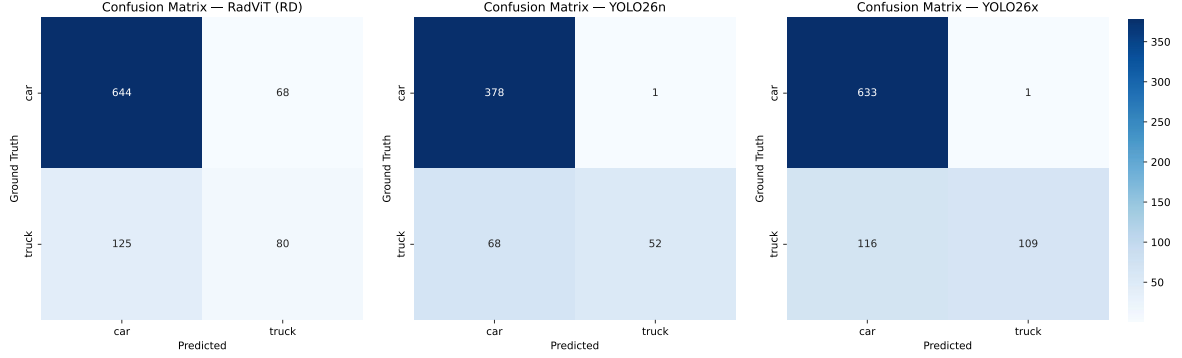


Figure 3.4: Confusion matrices for RadViT (RD), YOLO26n, and YOLO26x.

likely influenced by the class imbalance in the RADial dataset, where the car class is significantly more frequent than the truck class, which makes it harder to learn. For YOLO models, the difficulty seems to come from intra-class ambiguity. Most "truck" instances in RADial correspond to pickups, vans, and other more or less large vehicles, making it harder to clearly distinguish this class from cars in the camera domain.

Figure 3.5 presents the confusion matrices for RadViT operating in RD and ADC input modes. In ADC mode, RadViT achieves a higher car classification precision of 93.9% compared to 90.4% for the RD input. The improvement is even more pronounced for the truck class, where the ADC input reaches 55.3% compared to 39.0% for the RD input, surpassing even the truck classification precision of both YOLO variants. This result is consistent with the findings of Giroux et al. [36], who observed that linear layers mimicking Fourier transforms are able to better encapsulate radar sensor characteristics compared to fixed DSP preprocessing, particularly for classification tasks.

Although additional metrics such as mAP are also reported, these metrics inherently depend on detection performance, since a prediction is considered as a True Positive only if its IoU with a ground-truth bounding box exceeds the fixed IoU threshold and its predicted class label matches the ground-truth class label. In particular, YOLO is penalised by the projection from the image domain into the radar domain, which introduces additional localisation inaccuracies. On the other hand, RadViT is constrained by the fact that both training and evaluation are performed on the same RADial dataset, unlike YOLO models, which benefit from large-scale pretraining before evaluation on RADial. Table 3.5 reports detection performances across RadViT and

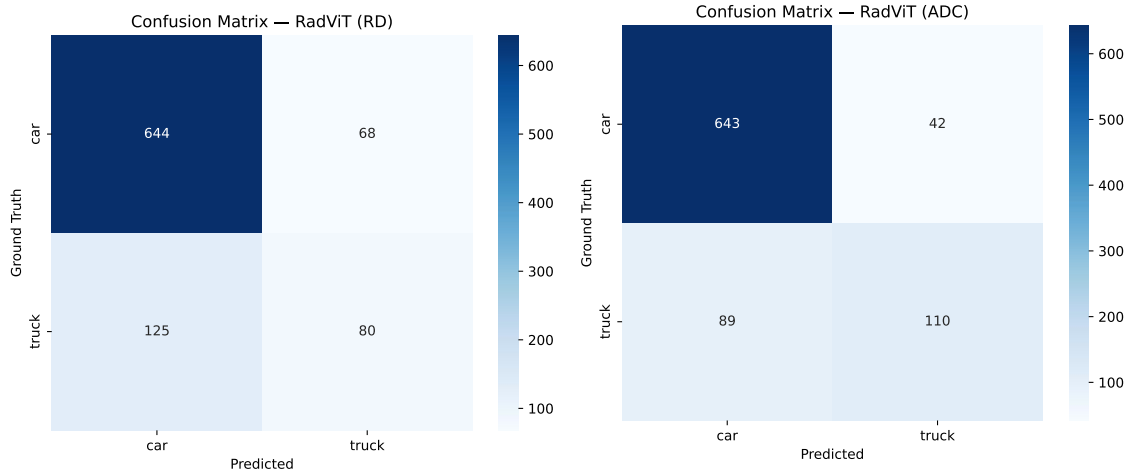


Figure 3.5: Confusion matrices for RadViT (RD) and RadViT(ADC)

YOLO models.

Model	Radar Input	mAP@50(%) $\uparrow$	mAR@50(%) $\uparrow$	F1(%) $\uparrow$
RadViT	RD - Shift	83.3	71.3	76.9
RadViT	ADC	81.0	69.9	75.0
YOLO26n	—	67.5	38.8	46.0
YOLO26x	—	67.5	60.1	63.6

Table 3.5: Detection performance comparison between RadViT and YOLO models evaluated in the radar domain.

Table 3.5 also explains why the total number of correctly localised instances differs across models. RadViT detects 712 cars and 205 trucks, compared to 379 and 120 for YOLO26n, and 634 and 225 for YOLO26x, reflecting the varying detection recall of each model in the radar domain and consistent with the penalties introduced by the camera-to-radar projection.

Table 3.6 reports mAP@50 and mAP@50:95 for all compared models. Detection performance has a significant impact on the results. RadViT using RD input suffers a drop of 13.8% compared to its detection-only AP (Table 3.2), while RadViT using ADC input experiences a small drop of 3.5%, partially compensated by its stronger classification performance. However, despite this classification advantage, the mAP@50:95 of RadViT using ADC input remains equal to that of RD input at 49.4%. This can be explained by the degradation of localisation precision in ADC mode at the stricter IoU thresholds.

Interestingly, the mAP@50 of YOLO models increases when classification is jointly evaluated, reaching 85.0% for YOLO26x and 70.2% for YOLO26n. This reflects that, among the detections YOLO successfully localises in the radar domain, the majority

are correctly classified, since YOLO’s classification head operates in the image domain. However, this advantage diminishes at stricter IoU thresholds, where YOLO26x achieves an mAP@50:95 of 48.9%, slightly below that of RadViT RD. This degradation reflects the localisation inaccuracies introduced by the camera-to-radar projection, which become increasingly penalising as the IoU threshold increases.

Model	mAP@50 (%)	mAP@50:95 (%)
RadViT (RD - Shift)	69.5	49.4
RadViT (ADC)	<u>77.5</u>	49.4
YOLOv26n	70.2	40.5
YOLOv26x	85.0	<u>48.9</u>

Table 3.6: mAP@50 and mAP@50:95 (%) comparison between RadViT and YOLO models for classification performance.

Overall, these results suggest that while RadViT on RD input achieves competitive detection performance, the ADC input offers a meaningful advantage in discriminating between vehicle categories. Combined with its lower end-to-end latency, the ADC pipeline therefore emerges as the preferred representation when both classification accuracy and real-time deployment are the primary objectives.

3.6 Multi-Object Tracking with Kalman Filtering

The preceding section focused on the object detection task. As introduced in Section 2.5, detections produced at each frame can be associated across time to maintain persistent object identities. This is called multi-object tracking. This section describes the Kalman filter-based tracking algorithm developed for this purpose and presents the corresponding evaluation results.

3.6.1 Kalman Filter-Based Algorithm

The Kalman Filter, also known as the linear-quadratic estimation filter, is a recursive algorithm that estimates a system’s current state over time by combining a mathematical process model with noisy input measurements. At each time step, the algorithm performs two steps: a prediction step, in which an estimate of the current state of the variables under uncertainty is produced, and an update step, which updates the prediction based on the incoming measurement [58]. The model is defined as follows:

$$x_k = F_{k-1}x_{k-1} + W_{k-1} \quad (3.8)$$

$$z_k = H_kx_k + V_k \quad (3.9)$$

where Equation (3.8) is the state transition equation with F_{k-1} the state transition matrix, x_{k-1} the state vector at time $k - 1$ and W_{k-1} a white noise matrix with covariance matrix Q . Equation (3.9) is the measurement equation, where z_k denotes the measurement observation at time k , H_k is the observation matrix, and V_k is the measurement noise with covariance matrix R .

The first step is the prediction step, where the filter will project the state ahead before receiving measurements.

$$x_k^- = F_{k-1}x_{k-1} \quad (3.10)$$

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_k \quad (3.11)$$

Equation (3.10) projects the state estimate forward in time, while Equation (3.11) projects the error covariance ahead. The matrix P represents the estimation error covariance. It quantifies the uncertainty associated with the estimate of the system state at time $k - 1$. The Notation "-" denotes a prior estimate, computed before incorporating the measurement z_k at time k , while the notation "+" denotes a posterior estimate, updated to account for the newly arrived measurement.

The second step is the measurement update. Once a new measurement z_k is obtained, the prior estimate is corrected by a weighted arithmetic mean of the predicted state and the incoming observation, yielding the posterior estimate $\hat{x}_k$. The Kalman gain is first computed as

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}$$

that weights up the significance of the new measurement against the prediction. The state estimate is then updated by correcting the prior prediction with the difference between the real measurements z_k and predicted measurement $H_k \hat{x}_k^-$ multiply by Kalman gain K z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-)$$

Finally, the error covariance is updated as

$$P_k = (I - K_k H_k) P_k^-$$

where I is the identity matrix. This step reduces uncertainty in the state estimate after integrating the new measurement.

3.6.2 Implementation of the tracking algorithm

The state vector is defined as $x_k = [x, y, vx, vy]^T$, where (x,y) denotes the Cartesian coordinates of the object centroid and (vx,vy) the corresponding velocity components. The measurement vector $z_k = [x, y]^T$ contains only the centroid position, as velocity is not directly measurable from RadViT output detections. The state transition matrix F models a constant velocity motion model with a time step of $\Delta t = 0.2s$, measured between two consecutive radar frames. The measurement noise covariance R , process noise covariance Q , and initial error covariance P are set to fixed diagonal matrices with values empirically tuned on the test set.

The tracking pipeline operates as follows. At each frame, all existing tracks first perform a Kalman filter prediction step to propagate their state estimates. Incoming detections are then associated with existing tracks by solving a linear assignment problem using the Hungarian algorithm [59], which minimises the total squared Mahalanobis cost across all possible track-detection pairs. Mahalanobis cost is defined as $d^2 = \delta^T S^{-1} \delta$,

where $\delta = z - \mu$ is the difference vector between the prediction and the measurement, and $S = HPH^T + R$ is the covariance matrix. A detection is associated with a track if the corresponding cost falls below a threshold of 9.21, corresponding to the 99th percentile of a χ^2 distribution with 2 degrees of freedom. Unmatched tracks have their miss counters incremented, and any track that exceeds 8 consecutive unmatched frames is deleted, allowing the tracker to handle temporary occlusion and missed detections. Unmatched detections are initialised as new tracks, with their position set to the detected centroid and their velocity initialised to zero. A track is confirmed only after it has been successfully matched for a minimum of 4 consecutive hits, ensuring that only persistent objects contribute to the evaluation.

3.6.3 Tracking metrics used

Metrics used to evaluate the tracking performance of RadViT include IDF1, MOTA, MOTP, and IDSW. IDF1 measures how consistently the tracker maintains correct object ID over time, rather than just counting detection errors:

$$IDF1 = \frac{2 \times IDTP}{2 \times IDTP + IDFP + IDFN}$$

where $IDTP$ denotes the number of correctly identified and consistently tracked detections, $IDFP$ is the number of predicted detections that do not match any ground truth, and $IDFN$ is the number of ground truth trajectories that are not tracked. MOTA provides a global measure of tracking accuracy that accounts for false positives, false negatives, and ID switches (IDSW):

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t}$$

MOTP measures the localisation precision of matched track-detection pairs, expressed in meters. It is defined as

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}$$

where $d_{t,i}$ is the distance between track i and its associate ground truth object at time step t and c_t is the total number of matches at time step t . Finally, IDSW counts how many times a tracked object is assigned a different ID than in the previous frame.

3.6.4 Tracking Results and Analysis

The tracker is evaluated on the RADial test set, using the manually annotated tracking labels described in Section 3.4, and metrics are computed with the py-motmetrics library [60]. To provide a performance baseline, the SORT algorithm [61] is used. SORT follows the same tracking-by-detection paradigm as the proposed approach, combining a Kalman filter for state estimation with the Hungarian algorithm for data association. The key difference lies in the association metric cost: SORT relies on the IoU between predicted and detected bounding boxes, whereas our approach uses the squared Mahalanobis distance. Additionally, SORT models each target with a state vector $x = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$, where (u, v) denotes the bounding box centre, s its area, and r its aspect ratio. In our case, the bounding box dimensions are fixed, eliminating the need to model scale and aspect ratio in the state vector.

Results are reported in Table 3.7. The results exhibit significant variability across sequences, reflecting the diversity of driving scenarios present in the test set. Comparing the two algorithms, the proposed tracker achieves a higher mean IDF1 (48.2% vs 46.0%), indicating that it maintains more consistent object identities over time. SORT, however, outperforms our approach in terms of MOTA (42.5% vs 41.3%) and MOTP, producing fewer false positives (19.75 vs 74.5) and achieving better localisation accuracy (3.968 m vs 4.124 m). This can be attributed to SORT’s IoU-based association, which directly optimises spatial overlap between predicted and detected bounding boxes. When the overlap is not met due to missed detections or large displacements, the track is not updated, leading to more false negatives. The Mahalanobis distance, by contrast, accounts for prediction uncertainty through the covariance matrix, allowing associations over large spatial distances and resulting in fewer missed objects, at the cost of higher false positives.

By analysing our approach, sequence 12.03.47 achieves the best results (MOTA 75.8%, IDF1 77.6%), which can be attributed to the simplicity of the scene where only one vehicle is present for most of the sequence, providing stable and consistent detections with few occlusions and trajectory changes. In contrast, sequence 12.45.05 is the most challenging, yielding a negative MOTA (-3.6%). The scenario involves a motorway overtaking manoeuvre in which the ego vehicle performs abrupt directional changes, generating sudden shifts in the tracked object’s relative position. This results in a high number of false positives (145) and missed detections.

Sequence 12.54.38 achieves a good MOTA (63.9%) and the best MOTP (0.937 m), suggesting accurate localisation when detections are correct. However, most of the vehicles are not tracked due to insufficient detections to confirm a track, resulting in a high number of mostly lost objects (12 out of 15) and false negatives. Finally, sequence 12.25.47 involves a country road scenario and achieves moderate performance. The sequence begins with a stationary vehicle on the roadside, followed by a phase where the ego vehicle follows a single truck. The tracker fails to confirm oncoming vehicles due to insufficient consecutive detections and struggles with abrupt velocity changes, such as braking at the end of the sequence. The high number of identity switches (5) reflects the difficulty of maintaining consistent associations under these conditions. For road observation applications such as the MUSE project, MOTA is arguably the more relevant metric, as it directly reflects the tracker’s ability to avoid missed detections and false alarms in each frame. From this perspective, SORT achieves a slightly higher mean MOTA. However, both algorithms are penalised by detection quality, as evidenced by the poor results on sequences 12.45.05 and 12.25.47. Improving detection performance would therefore be the first step before drawing definitive conclusions about the choice of association strategy. It should also be noted that ground-truth annotations are not available for every frame in the tracking test set, as discussed in 3.4. Since metrics are computed only on annotated frames, tracking behaviour between annotations is neither rewarded nor penalised, which may affect the reliability of the reported metrics. Further directions for improvement are discussed in Section 5.

Tracker	Sequence	IDF1	MT	PT	ML	FP	FN	IDSW	MOTA	MOTP (m)
Kalman	12.45.05	31.6%	0	2	2	145	226	2	-3.6%	5.108
	12.25.47	37.8%	0	2	3	82	94	5	29.0%	7.138
	12.03.47	77.6%	1	0	2	22	7	1	75.8%	3.312
	12.54.38	45.8%	1	2	12	49	78	1	63.9%	0.937
	Mean	48.2%	0.5	1.5	4.75	74.5	101.3	2.25	41.3%	4.124
SORT	12.45.05	17.5%	0	1	3	53	269	6	8.9%	4.598
	12.25.47	30.8%	0	1	4	24	183	5	16.9%	7.414
	12.03.47	50.0%	0	1	2	0	36	2	69.4%	3.100
	12.54.38	85.6%	1	0	14	2	88	0	74.6%	0.758
	Mean	46.0%	0.25	0.75	5.75	19.75	144.0	3.25	42.5%	3.968

Table 3.7: Per-sequence multi-object tracking performance of RadViT on the RADial test set, comparing the Kalman/Mahalanobis tracker against SORT. MT: Mostly Tracked, PT: Partially Tracked, ML: Mostly Lost, FP: False Positives, FN: False Negatives, IDSW: ID Switches.

3.7 Summary

This chapter presented the first main contribution of this thesis. RadViT, a Vision Transformer-based architecture for radar object detection. Built upon the ViT backbone, RadViT processes Range-Doppler maps or raw ADC signals through a radar-adapted patch embedding, a shared Transformer encoder, a multi-scale feature pyramid, a Range-Angle decoder and a Detection Header inherited from T-FFTRadNet. The RADial dataset was extended with manual classification and tracking annotations to support the full evaluation pipeline.

Experimental results showed that RadViT achieves competitive detection performance relative to its design constraints, with F1 scores of 76.9% and 75.0% for RD and ADC inputs, respectively, confirming that the Range-Doppler representation outperforms raw ADC signals for object detection. The ADC pipeline, however, offers a lower end-to-end latency (142.00 ms vs 275.23 ms), making it more suitable for real-time deployment scenarios. Furthermore, the ADC pipeline demonstrated stronger classification performance, consistent with findings from T-FFTRadNet, which show that learned Fourier-like transformations better encapsulate radar sensor characteristics than fixed DSP preprocessing. All models struggle to correctly classify trucks, a difficulty attributed to class imbalance in RADial and intra-class ambiguity between trucks and cars for YOLO. The joint detection-classification mAP metrics are penalised by detection performance, which explains the lower mAP@50:95 observed across all models. Finally, a Kalman filter-based tracker was evaluated against SORT on four test sequences. Tracking performance was shown to be heavily dependent on detection quality. SORT achieves higher MOTA and MOTP, whereas the proposed approach outperforms SORT on IDF1, indicating a better ability to maintain consistent object identities over time.

Chapter 4

Characterisation of K-MD2 radar

The previous chapter presented RadViT, a Transformer-based architecture for radar object detection, validated on the RADIAL dataset. While the model treats its input as a numerical tensor, the data it operates on is the direct product of a physical sensing process, as determined by the radar sensor’s hardware characteristics. Achieving true operational robustness requires a thorough understanding of the physical sensing process, including the sensor’s resolution limits, intrinsic noise floor, antenna leakage, and signal artefacts inherent to radar hardware.

A significant challenge in using public datasets like RADIAL is the lack of detailed documentation regarding the sensor’s internal hardware and low-level signal processing. While RadViT achieves high performance on this data, the underlying physical artefacts remain largely unknown.

Consequently, the experimental characterisation of the K-MD2 sensor presented in this chapter is a critical step. This study provides a necessary foundation for the eventual deployment of RadViT on the K-MD2 sensor by characterising the physical phenomena and signal artefacts that any radar perception pipeline operating on such data will inevitably encounter, a prerequisite for bridging the domain shift between the two sensors through transfer learning, as discussed in Section 5.2.

4.1 Devices characteristics

The K-MD2 is a traffic radar sensor operating at 24 GHz. It employs FMCW signal processing (as explained in 2.2). It uses one transmitter antenna (Tx) and three receiver antennas (Rx). The sensor is able to detect targets at distances of up to 200m and at speeds reaching ± 130 km/h. The radar is considered a 4D radar, as it estimates elevation and azimuth angles in addition to range and velocity. However, the elevation and azimuth fields of view are very limited, at $\pm 10^\circ$ (elevation) and $\pm 30^\circ$ (azimuth), with an angular resolution of 0.1° . It is therefore preferable not to use angle information with this radar, as this may result in accuracy errors. At its maximum configured range and velocity, the sensor offers a range resolution of 1 meter and a velocity resolution of 1 km/h. These resolutions improve as the maximum measurable range and speed

are reduced. The reader can find additional details in François Ledent’s master’s thesis [62].

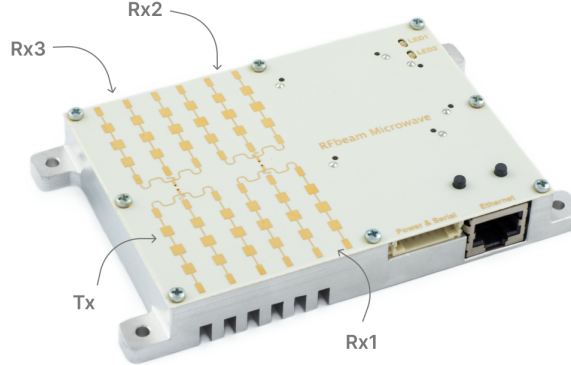


Figure 4.1: K-MD2 from RFBeam

As described in section 2.2, the K-MD2 operates using a standard FMCW signal processing. In practice, to obtain range, the K-MD2 samples 260 points per chirp but discards the first 4, resulting in only 256 samples for processing. This truncation eliminates the frequency discontinuity that occurs at the start of each ramp. Similarly, 257 chirps are emitted per frame, but the first is discarded, leaving 256 chirps for Doppler processing. Ignoring the first chirp is necessary to exclude disturbances in the first chirp, which are caused by jumping to the start frequency [63].

4.2 K-MD2 radar calibration in Anechoic Chamber

An anechoic chamber is a room designed to completely absorb sound or electromagnetic waves. The walls, ceiling and floor are covered with wedge-shaped foam that prevents waves from reflecting back into the room. As a result, the space provides a controlled acoustic environment. In our case, this chamber serves two purposes in this study. First, it provides a reflection-free environment to validate the accuracy and precision of our radar. Second, it enables characterisation of the system’s intrinsic noise floor and TX-RX antenna leakage. Antenna leakage is particularly present in FMCW radar. It consists of direct and undesired transmission of signal power from the transmitter antenna (Tx) to the receiver antenna (Rx) without passing through a target. This coupling creates a strong false "zero-range" echo that can saturate the receiver’s low noise amplifier and increase the noise floor. In addition, the leakage signal has a frequency similar to that of short-range target echoes. As a result, its higher power can obscure these signals, making it more difficult for the radar to distinguish target frequencies and accurately detect objects. The reduced distance between the Tx antenna and the Rx antenna increases the magnitude of the leakage signals [64].

4.2.1 Computation of the system’s intrinsic noise floor

To perform this experiment, we place the radar inside the anechoic chamber with nothing inside, and record several frames. Figure 4.2 illustrates the range Doppler map obtained under these conditions.

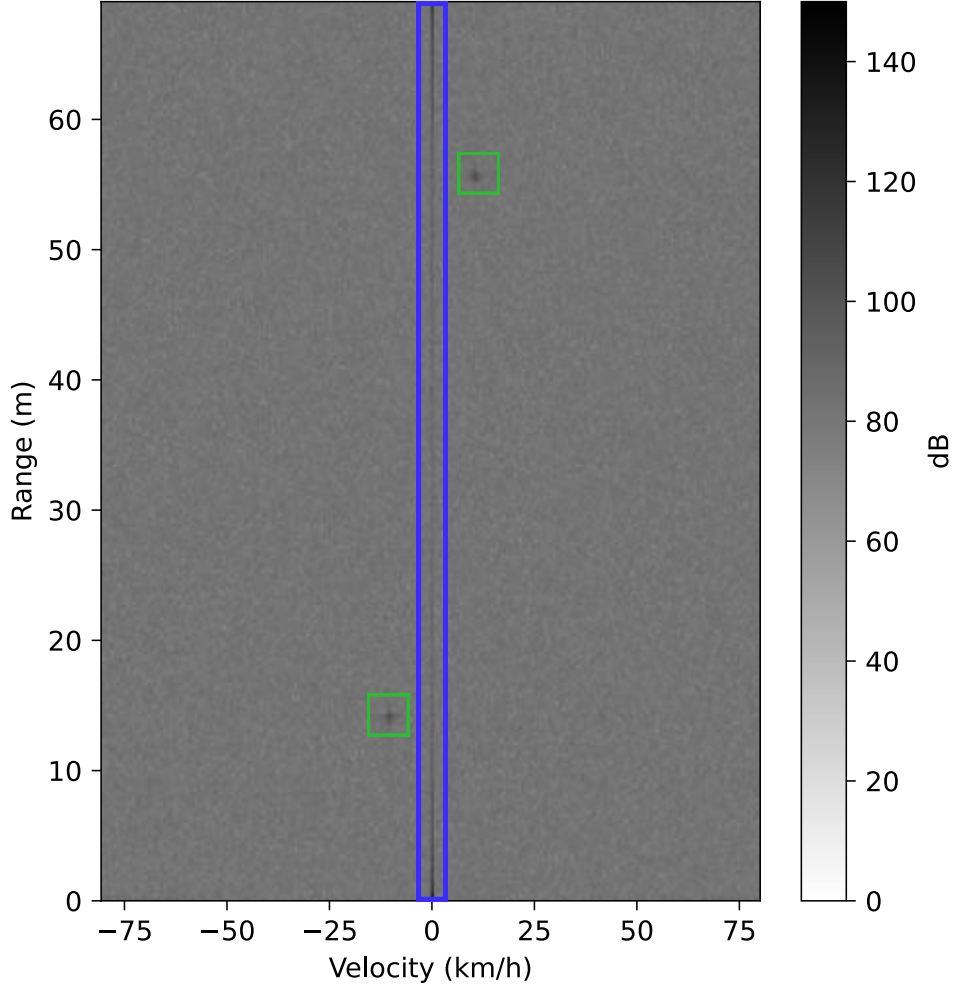


Figure 4.2: Range Doppler map result

Several observations can be drawn from this figure. First, the direct TX-RX antenna leakage is surrounded in blue. That corresponds exactly to the signal component we wanted to calculate and observe. Second, static artefacts are also observed and surrounded in green. These artefacts appear at fixed positions across all frames and do not correspond to real physical targets. These artefacts are not problematic because, thanks to this background measurement, they can be subtracted from the subsequent range Doppler spectrum. Finally, anything visible on the Range-Doppler map that is not surrounded corresponds to the radar's initial noise. This noise is particularly useful when we want to calculate the signal's effective power from the measured power, which is estimated as 82.03 dB.

The K-MD2 supports multiple presets that define the maximum achievable range and velocity, controlled through parameters such as chirp bandwidth and initial delay. The range resolution is given by:

$$\Delta R = \frac{c}{2} \times \left(\frac{N + M}{N \times B_w} \right)$$

where c is the speed of light, N is the number of used samples, M is the number of ignored samples, and B_w is the bandwidth in MHz. The corrective factor $\frac{N+M}{N \times B_w}$ is necessary to discard the first 4 samples. The maximum range then follows as

$$R_{max} = (N - 1) \times \Delta R$$

The velocity resolution is given by

$$\Delta V = \frac{\lambda \times clk \times 3.6}{2 \times N \times (S_{clk} \times (N + M) + delay)}$$

where λ is equal to wavelength, clk is the core clock frequency, S_{clk} that is the number of core clocks per sample and $delay$ is the initial chirp delay in clock cycle. The factor 3.6 converts the result from m/s to km/h. The maximum velocity is therefore

$$V_{max} = \Delta V \times \left(\frac{N}{2} - 1\right)$$

Increasing both the maximum detection range and the maximum measurable velocity results in a degradation of resolution. Note that person detection is not possible above 100 meters, and vehicle detection is not possible beyond 200 meters.

Since these parameters directly influence the system noise floor and the spatial distribution of static artefacts, we have performed a dedicated background measurement for each preset configuration. These presets are detailed in the datasheet of K-MD2 [63]. For our measurements, a preset of 70 m and 80 km/h was selected, unless explicitly stated otherwise.

4.3 Radar Cross-Section

Radar cross-section (RCS) measures the object's ability to reflect radar energy back towards its source. It characterises the target's signature and indicates how easily it can be detected. It is often expressed in m^2 , although RCS does not correspond directly to the physical surface of the object, but an equivalent area that quantifies how much power a target reflects toward the receiver, compared to an idealised reference reflector. Characterising the RCS of radar targets is essential for understanding their radar signatures, since objects such as vehicles and pedestrians exhibit significantly different RCS values.

There are two main approaches to measuring the RCS of a target [65]. The first is calibration without a calibration target, which derives the RCS directly from the radar range equation using all known system parameters. The second is calibration with a calibration target, which does not require knowledge of individual system parameters but instead uses a reference target with a known RCS to calibrate the measurement. In this work, we adopt the last approach using a metallic sphere as the reference target.

4.3.1 Radar Cross-Section computation

The radar range equation relates the received power to the system parameters and the target RCS:

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4} = K \times \frac{\sigma}{R^4} \quad (4.1)$$

where P_t is transmitted power, G_t is transmitter antenna gain, G_r is receiver antenna gain, λ is the wavelength, R is the target range, and σ is the RCS. Since all system parameters are fixed in a given configuration, they can be encapsulated in a single constant K . The calibration with a calibration target does not require knowledge of the individual system parameters. A metallic sphere is commonly chosen as this reference because it reflects the signal equally in all directions.

$$\sigma_{sphere} = \pi r^2 \quad (4.2)$$

In our case, the calibration sphere has a radius of $r = 0.3m$, giving $\pi \times 0.3^2 = 0.09\pi$. Applying the radar range equation to sphere measurement allows us to isolate K

$$P_{sphere} = K \times \frac{\sigma_{sphere}}{R_{sphere}^4} = K \times \frac{\pi r^2}{R_{sphere}^4} \quad (4.3)$$

$$K = \frac{P_{sphere} R_{sphere}^4}{\pi r^2} \quad (4.4)$$

This substitution method relies on several hypotheses:

1. **Identical radar configuration:** The radar preset must remain identical between the sphere calibration and the object measurement, so that K remains unchanged.
2. **Uniform plane wave illumination:** The incident wave at the target must satisfactorily approximate a uniform plane wave. Following Kouyoumjian & Peters [66], this requires that the maximum phase variation over the target does not exceed $\pi/8$ radians and the maximum amplitude variation does not exceed 1.0dB. A sufficient condition to require this measurement is to be in the far field of the target, defined as $R \gg \frac{2L^2}{\lambda}$, where L is the largest transverse dimension of the target.
3. **Equivalent propagation conditions:** Equation 4.1 assumes free-space propagation, which holds in anechoic chambers used for sphere calibration but not in the outdoor measurement environment, where multipath reflections and varying environmental conditions alter the effective propagation. In what follows, we assume that the propagation conditions are equivalent between the two measurements, so that K remains unchanged. This is a strong hypothesis and constitutes a source of uncertainty in the absolute RCS values reported here.
4. **Rectilinear target trajectory:** The target is assumed to follow a straight trajectory during the measurement, ensuring a constant aspect angle. A non-straight trajectory would introduce variations in the aspect angle, leading to errors in the RCS estimate.

4.3.2 Experimental Results

The sphere calibration measurement was performed in an anechoic chamber, where the sphere was placed 2 meters from the radar. The sphere is mounted on a support designed to minimise reflections. The radar, positioned on this same support and on another box to be more or less centred in relation to the centre of the sphere, points toward the sphere. Figure 4.3 describes the experimental setup put in place.

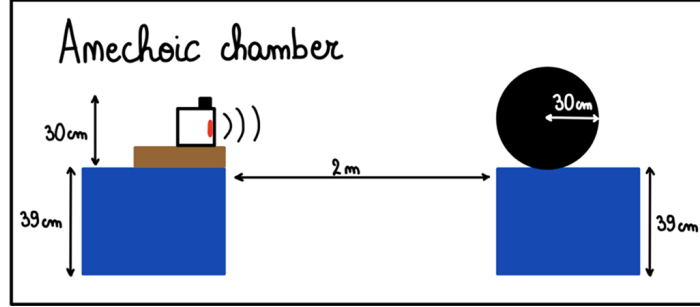


Figure 4.3: Anechoic chamber setup

Figure 4.4 shows the resulting Range Doppler map. The map is obtained after subtracting the background signal. Otherwise, antenna leakage would mask the static target. A clear peak is observed at approximately $2.15\text{m} \pm 0.2749\text{m}$. This is consistent with the given resolution, as a physical distance of 2m corresponds to bin $\frac{2}{\Delta R} = \frac{2}{0.2749} = 7.275$. This places the peak between bin 7 and bin 8, which, in meters, corresponds to $[7 \times 0.2749, 8 \times 0.2749] = [1.924\text{m}, 2.199\text{m}]$. The peak amplitude reaches 23.4 dB after background subtraction. In the right panel of Figure 4.4, a vertical spread along the range axis is clearly visible, arising from the sidelobe structure of the range FFT response.

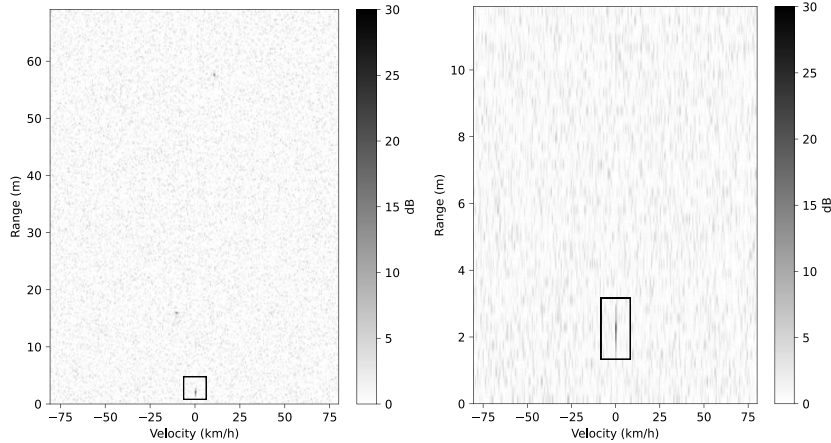


Figure 4.4: Resulting Range-Doppler map: the left image provides the full, unzoomed map, while the right image provides a zoomed-in view.

Object measurements were performed in three distinct outdoor environments, covering two different vehicles and one pedestrian, allowing a qualitative comparison of radar reflectivity under varying conditions, as illustrated in Figure 4.5. Before analysing the RCS results, we introduce the Signal-to-Noise Ratio (SNR), defined as:

$$SNR = 10 \times \log_{10}\left(\frac{P_{signal}}{P_{noise}}\right)$$

where P_{signal} corresponds to the effective signal power, while P_{noise} corresponds to initial noise measured in anechoic chamber. The SNR serves as a reliability indicator for the RCS measurement: a high SNR indicates a clean, well-defined signal, while a low SNR indicates a weak signal relative to noise, leading to unreliable measurements.



Figure 4.5: Camera images of the three measurement environments: (a) Citroën DS4, (b) Skoda Fabia, (c) pedestrian.

Validation of the Hypothesis

We will now verify whether the hypotheses in Section 4.3.1 are satisfied under our measurement conditions:

- **Hypothesis 1 - Identical radar configuration** is satisfied by construction: the same radar preset was used for both the sphere calibration and the outdoor measurements.
- **Hypothesis 2 - Uniform plane wave illumination:** For the sphere measurement, the far-field condition is not respected $R \gg 2L^2/\lambda \approx 57.9m$. However, the necessary condition is not to be in the far field of the target, but rather that the incident field approximates a uniform plane wave at the target location, which is the case thanks to an anechoic chamber environment that eliminates multipath reflections [67]. For the object measurement, the far-field condition $R \gg 2L^2/\lambda$ cannot be satisfied for a vehicle of transverse dimension $L \approx 1.8$ m at $\lambda = 0.01243$ m, which would require $R \gg 521$ m. This range is beyond the radar's detection range. However, following [66], for large targets where only median RCS values are of interest, the phase condition can be relaxed considerably, while the amplitude variation must remain below 3 dB. Given that the incident fields decay as R^{-1} in the far field of the antenna, the amplitude variation over the longitudinal extent D of the target at range R is given by:

$$A_D^{dB} = 20 \log_{10}(1 + D/R)$$

where D is assumed to be the vehicle length. But usually, the minimum range is restricted by the variation of the incident field in the transverse directions. So D is the same as L . RCS measurements are therefore considered valid for $5 \leq R \leq 70$

- **Hypothesis 3 - Equivalent propagation conditions** is the strongest assumption. In practice, outdoor multipath reflections introduce deviations from free-space propagation that are not present in the anechoic chamber.
- **Hypothesis 4 - Rectilinear trajectory** was ensured by experimental design, where the target moved in a straight line directly away from the radar.

Results

Figures 4.6 and 4.7 show the SNR and RCS as a function of distance for the Skoda Fabia and the Citroën DS4, respectively. The SNR curve exhibits the expected $1/R^4$ decay predicted by the radar range equation 4.1. Since received power scales as $P_r \propto 1/R^4$, when we double the distance $1/(2R)^4 = 1/16R^4$, we divide the SNR by 16. In dB, this is equal to $10 \times \log_{10}(\frac{1}{16}) \approx -12\text{dB}$. This is consistent with the observed drop in Figure 4.6 at 5m, the SNR is equal to $\pm 35\text{dB}$ and at 10m, the SNR is equal to $\pm 25\text{dB}$. In practice, the decay is not perfectly monotonic due to multipath interference and other environmental effects.

We define an arbitrary threshold of 17dB under which we consider measurements too noise-corrupted to be trustworthy. Several RCS peaks are observed at large distances (50–65 m for the Skoda Fabia, beyond 40 m for the Citroën DS4), where both SNR and RCS increase simultaneously, inconsistent with the expected $1/R^4$ decay. These anomalies are attributed to multipath reflections, which introduce spurious power increases that are strongly amplified by the R^4 scaling in the RCS formula. Measurements beyond these distances are therefore excluded from the mean RCS computation. Over the reliable range, the mean estimated RCS reaches 44.5 m^2 for the Skoda Fabia and 35.6 m^2 for the Citroën DS4, both consistent with values reported in the literature for vehicles, typically ranging between 10 and 50 m^2 [68].

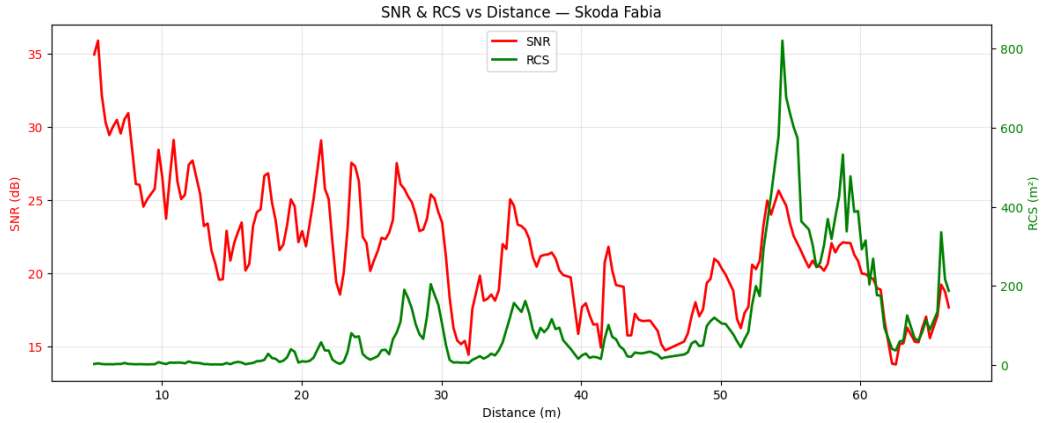


Figure 4.6: SNR and RCS evolution across distances for Skoda Fabia.

Figure 4.8 shows the corresponding results for the pedestrian. The SNR exhibits a similar $1/R^4$ decay trend, but the RCS shows significantly greater variability than the vehicle measurements. This is attributed to the micro-Doppler effect inherent to human movements. Arms, legs and torso move at different radial velocities and present

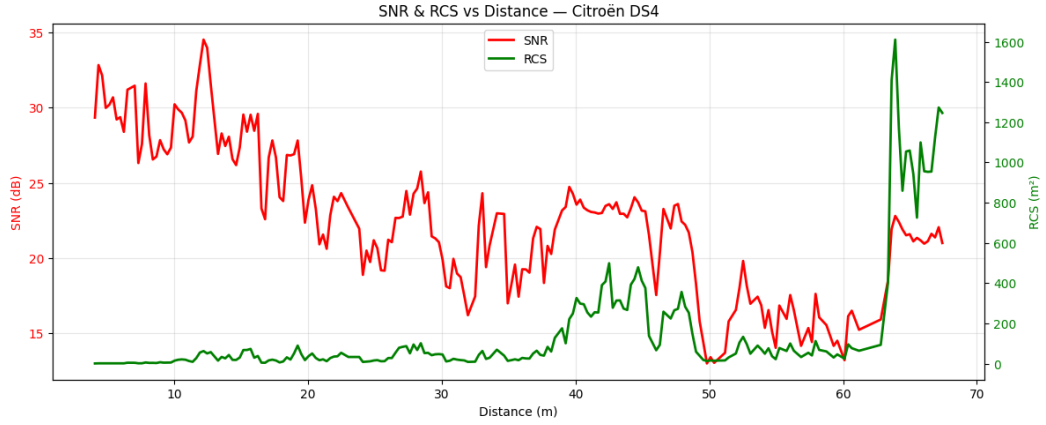


Figure 4.7: SNR and RCS evolution across distances for Citroën DS4.

varying reflective surfaces at each radar frame, causing the received power to fluctuate substantially between consecutive frames. An arbitrary lower SNR threshold of 15 dB is applied to account for the weaker radar returns from pedestrians. Over the reliable range, the mean estimated RCS reaches $0.565m^2$, which is also consistent with values reported in the literature [68] for a pedestrian ranging between 0.1 and $1 m^2$.

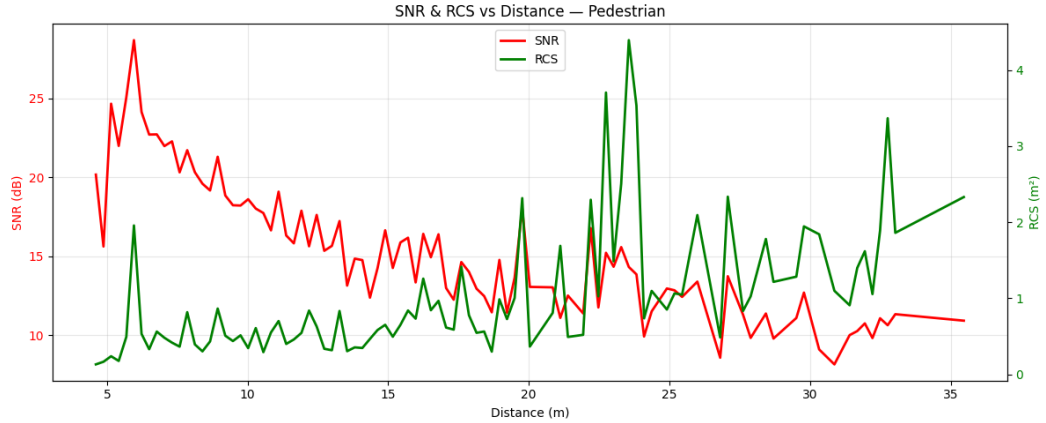


Figure 4.8: SNR and RCS evolution across distances for pedestrian data.

4.4 Radar artefacts

In this section, we will describe two other artefacts/ambiguities that appear in the Range Doppler spectrum, but that do not correspond to a real physical target. We added to those observed in the anechoic chamber.

4.4.1 I/Q Imbalance

The first one is called I/Q imbalance, which occurs when there is a mismatch between the In-phase (I) and Quadrature (Q) channels, primarily due to hardware imperfections

in the direct-conversion receiver. In an FMCW radar, the first step is generating a chirp signal whose frequency increases linearly over time. This signal is amplified by a power amplifier and then transmitted by the transmitting antenna. This signal will serve as a reference for demodulation. The quadrature receiver demodulates the received echo by multiplying it with the transmitting signal, producing sum and difference frequency components. The high-frequency term is eliminated by a low-pass filter (LPF), retaining only the low-frequency term. This operation is performed twice in parallel, once in the in-phase channel and once in the quadrature channel, where the reference signal is phased-shifted by $\frac{\pi}{2}$, yielding the cosine and sine components of the beat signal, respectively. Combining them forms a complex baseband signal. Ideally, the I and Q channels produce:

$$I(t) = \alpha \cos(\omega t), \quad Q(t) = \beta \sin(\omega t + \theta)$$

where $\omega = 2\pi f_0$, that is, the frequency, and t is time. However, in practice, hardware imperfection introduces three sources of mismatch [69]: an amplitude mismatch ($\alpha \neq \beta$), a phase imbalance ($\theta \neq 0$) and DC offsets on each channel. The I/Q imbalance can considerably degrade the target detection performance of radar.

In Figure 4.9, the peak surrounded in green corresponds to the van identified in the camera image, while its complex conjugate is shown in blue. The peak surrounded in red corresponds to the parked car.

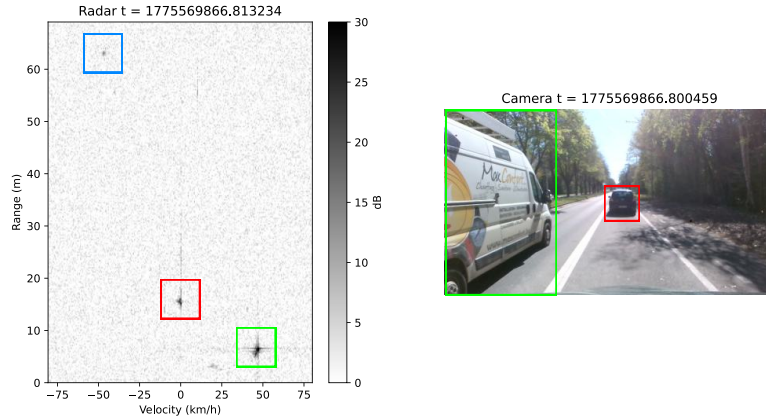


Figure 4.9: I/Q Imbalance illustration

Several methods exist in the literature to correct I/Q imbalance. Here we describe the method proposed by Kurniawan et al. [69]. The first step is DC offset removal. The DC offset on each channel is estimated as the mean of the signal over N samples and subtracted from the I and Q signals. The second step is phase correction. The true phase angle ϕ between I and Q is estimated via the normalised cross correlation :

$$\phi = \cos^{-1} \left(\frac{\sum_{n=0}^{N-1} I_n Q_n}{\sqrt{\sum_{n=0}^{N-1} I_n^2} \cdot \sqrt{\sum_{n=0}^{N-1} Q_n^2}} \right)$$

The phase imbalance angle is then $\theta = \pi/2 - \phi$, representing the deviation from the ideal $\pi/2$ phase shift. The corrected Q signal is obtained by removing the leakage of I into Q :

$$Q_{corr} = \frac{Q - I \sin(\theta)}{\cos(\theta)}$$

The last step is the amplitude correction. The amplitude of each channel is estimated via Parseval's theorem ($A = \sqrt{2P_0}$) and Q_{corr} is scaled by the ratio of α/β to equalise the two channel amplitudes. The correct complex signal will therefore be $x_{corr}(t) = I(t) + jQ_{corr}(t)$.

4.4.2 Range and Doppler aliasing

The second is called range aliasing, which occurs when a target detected by the radar exceeds the radar's configured maximum unambiguous range R_{max} . As established in section 2.2.1, an FMCW radar system emits chirps whose frequency increases linearly over a bandwidth B during a specific duration T_c , yielding a slope $S = \frac{B}{T_c}$. When the transmitted signal reflects off a target at a distance R , it returns with a time delay equal to $\tau = \frac{2R}{c}$, where c is the speed of light. When we mix the transmitted and received signals, we obtain the beat frequency:

$$f_b = S \cdot \tau = \frac{2RS}{c}$$

which encodes the range linearly. The FFT can resolve frequencies only within $[0, F_s/2]$ by the Nyquist-Shannon theorem, which defines the maximum unambiguous range

$$f_b \leq \frac{F_s}{2} \tag{4.5}$$

$$\frac{2R_{max}S}{c} \leq \frac{F_s}{2} \tag{4.6}$$

$$R_{max} = \frac{cF_s}{4S} \tag{4.7}$$

For any target with $R > R_{max}$, the beat frequency f_b exceeds $F_s/2$ and is folded back by the ADC.

$$f_{visible} = f_{real} - F_s \implies R_{visible} = R_{real} - R_{max} \tag{4.8}$$

Figure 4.10 illustrates this phenomenon: the van (surrounded in green) is physically located at approximately 80m ahead of the radar, with $R_{max} = 70m$ (Section 4.2.1), it appears at $80 - 70 = 10m$ on the Range-Doppler. The red car behind the van is surrounded in pink.

An analogous ambiguity arises in the velocity dimension. Between two consecutive chirps separated by T_c , a target moving at velocity V induces a phase shift [13].

$$V = \frac{\lambda \Delta\phi}{4\pi T_c} \tag{4.9}$$

$$\Delta\phi = \frac{4\pi V T_c}{\lambda} \tag{4.10}$$

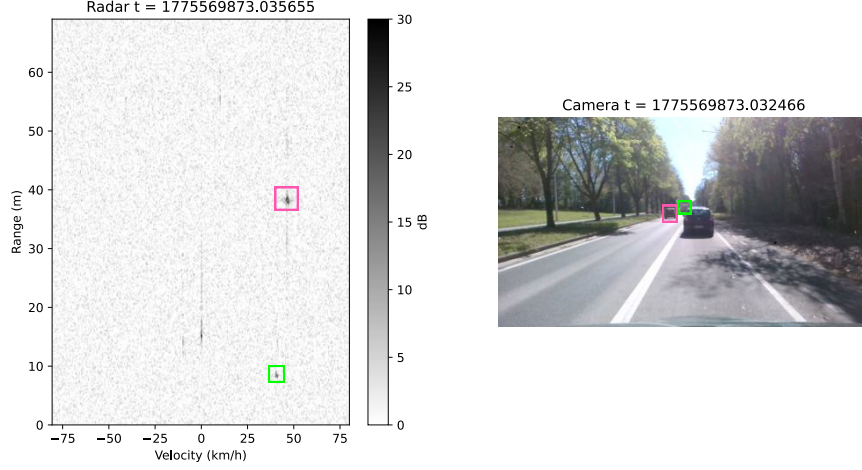


Figure 4.10: Range aliasing illustration

The phase shift can only be unambiguously recovered within $[-\pi, +\pi]$ [70]. The unambiguity condition, therefore, requires:

$$\begin{aligned}
 |\Delta\phi| &\leq \pi \\
 \frac{4\pi V_{max} T_c}{\lambda} &\leq \pi \\
 V_{max} &= \frac{\lambda}{4T_c}
 \end{aligned} \tag{4.11}$$

Any target whose speed exceeds V_{max} will appear at a ghost velocity $V_{visible} = V_{real} - 2V_{max}$, potentially with an inverted sign. For the preset configured at ± 80 km/h (Section 4.2.1), a target at 90 km/h would appear at $90 - 160 = -70$ km/h, as an approaching target.

4.5 Summary

This chapter presented the characterisation of the K-MD2 radar sensor, providing the physical requirements necessary for any perception pipeline operating on its data. Three main contributions are made :

First, the system's intrinsic noise floor was measured in an anechoic chamber, enabling background subtraction that removes TX-RX antenna leakage and static artefacts, thereby improving the visibility of target signatures on the Range-Doppler map.

Second, the Radar Cross-Section of two vehicles and one pedestrian was estimated using the substitution method with a metal sphere. The results are consistent with the literature: $44.5m^2$ and $35.6m^2$ for the Skoda Fabia and Citroën DS4, respectively, and $0.565m^2$ for the pedestrian. However, the RCS values are subject to uncertainty due to multipath reflections in the outdoor measurement environment, which deviate from the

free-space propagation assumed in the radar range equation. These RCS measurements can be useful to improve classification accuracy.

Third, three signal artefacts inherent to the K-MD2 hardware were characterised: I/Q imbalance, range and Doppler aliasing. Each of these can produce ghost targets or distorted signatures on the Range-Doppler map, and must be accounted for in any detection pipeline.

Having established the physical characteristics of the target domain, the next step is to bridge the domain shift between the RADIAL dataset and the K-MD2 sensor through transfer learning, as discussed in Chapter 5.

Chapter 5

Next steps

This section outlines several key directions for future work, building on the contributions presented in this master’s thesis. We first discuss strategies to mitigate overfitting, followed by transfer learning approaches to adapt RadViT to the K-MD2 radar sensor and by fusion of radar and camera modalities. Finally, we propose enhancements to the multi-object tracking pipeline.

5.1 Methods to mitigate overfitting

A primary limitation of RadViT is its tendency to overfit, which limits generalisation performance. This arises from the limited size of the RADIAL dataset, whereas training a Vision Transformer typically requires large amounts of data. A first mitigation strategy is data augmentation, which artificially increases the effective size of the training set by applying transformations such as rotation, cropping, and flipping. The second, and potentially more impactful, is the use of synthetic radar data. RadSimReal [71], for example, offers a physical radar simulation method that requires no prior knowledge of radar hardware specification or signal processing. A third direction, explored but not yet fully converged in this work, is self-supervised pre-training of the transformer encoder via the Masked Autoencoders (MAE). By randomly masking patches of the Range-Doppler spectrum and training the encoder to reconstruct the missing regions. The model can learn meaningful spectral representations from unlabelled data prior to fine-tuning on the detection task. Further investigation into masking ratio scheduling, decoder depth, and the choice of reconstruction target represents a direction to improve initialisation quality and reduce overfitting.

5.2 Transfer learning strategies

Closely related to the lack-of-data problem discussed above, training a Vision Transformer for radar object detection not only requires large amounts of annotated data but also raises the question of how well a trained model can transfer across different radar sensors. Given the diversity of available radar sensors, with different antenna configurations, operating frequencies and parameter settings, it is impossible to collect a dedicated dataset for every sensor type. Deploying a model trained on one radar

to another will introduce a distribution mismatch in the input data, also known as a domain shift in machine learning.

In the radar object detection literature, domain shift has been studied primarily in two settings: varying configurations of the same sensor [72], and varying environmental conditions [73]. To the best of our knowledge, cross-sensor transfer learning between two physically distinct radar systems remains a largely open research problem.

A natural first step toward deploying RadViT on the K-MD2 would be to collect and annotate a dataset with the K-MD2 sensor and apply standard fine-tuning, freezing the deeper components of the network while adapting the task-specific layers to the new sensor’s characteristics. If the domain gap between two sensors is too large, dedicated cross-sensor adaptation may be required. Establishing such techniques for physically distinct radar systems represents an open direction for future research.

5.3 Multi-modal fusion with camera

Having analysed radar data and designed an architecture for robust object detection, the natural next step is to integrate the radar sensor into a multimodal pipeline alongside a camera. Since the two modalities offer complementary characteristics, their fusion enables a richer perception of the surrounding environment, capturing outlines, colours, textures, ranges and velocities simultaneously. Furthermore, such a system can operate continuously regardless of lighting or weather conditions. This integration is the subject of Léonel Ngague’s master’s thesis: "Design and Validation of a Radar-Camera Fusion Framework For Object Detection".

5.4 Improvement of Tracking pipeline

Tracking performance is inherently tied to detection quality. As shown in the experimental results 3.6.4, missed detections and false positives directly degrade the tracker’s ability to maintain consistent object identities. Improving RadViT’s detection performance would therefore translate directly into better tracking results.

Beyond detection improvements, the tracking algorithm itself could be extended. A first limitation is its inability to reliably track oncoming vehicles, which appear briefly in the radar field of view and rarely accumulate enough consecutive detections to satisfy the confirmation threshold. Lowering this threshold for fast-moving objects approaching the sensor, or adopting a two-stage association strategy such as ByteTrack [74], could improve recall for this challenging scenario.

Finally, another direction could be to move away from the Tracking-by-Detection paradigm and instead train the full pipeline end-to-end. We can cite MOTR [75] or TrackFormer [76], which integrate object tracking directly into the Transformer decoder, allowing the model to learn temporally consistent representations rather than relying on detection-based association.

Chapter 6

Conclusion

This master’s thesis addressed the challenge of radar-based object detection and tracking for autonomous driving perception, motivated by the need for robust, cost-effective sensing solutions that operate reliably under adverse conditions. As a primary contribution, we developed RadViT, a Vision Transformer-based architecture designed around three guiding principles: enabling fair comparison between Range-Doppler maps and raw ADC signals, exploiting the multi-antenna structure of radar data through a dedicated patch-embedding module, and remaining independent of specific radar hardware to facilitate generalisation across sensor configurations.

RadViT was trained and evaluated on the RADial dataset, which provides synchronised multi-modal data and multiple radar representations, making it well-suited for studying the impact of input representation on detection performance. The experimental results reveal a clear trade-off between the two representations. Range-Doppler maps achieve superior performance, with an F1 score of 76.9% compared to 75.0% for ADC input. The ADC pipeline, by contrast, offers substantially lower end-to-end latency (142.00 ms vs. 275.23 ms) and stronger classification performance. To answer the research question, the choice of input representation should be guided by the deployment context: Range-Doppler input should be prioritised when detection performance is the primary objective, whereas raw ADC signals are preferable when end-to-end latency is a critical constraint or when classification performance is an additional objective. The detections produced by RadViT were subsequently used as input to a Kalman filter-based multi-object tracking pipeline, whose results were benchmarked against the SORT algorithm. Our approach achieves a superior mean IDF1 of 48.2% compared to 46.0% for SORT, reflecting stronger identity consistency, while SORT retains an advantage on MOTA (42.5% vs 41.3%) and MOTP (3.968 m vs 4.124 m), suggesting a trade-off between identity preservation and localisation accuracy.

Nevertheless, RadViT and the tracking pipeline share several limitations. On the detection side, the primary bottleneck is the simulated feature pyramid, which lacks the true multi-scale representations produced by a hierarchical backbone such as the Swin Transformer. Training the ViT backbone from scratch on a dataset of only 6.231 samples led to overfitting, limiting the model’s generalisation capacity. On the tracking side, the evaluation was performed on the test set, revealing that overall performance

was heavily influenced by the quality of upstream detections.

The second part of this thesis focused on characterising the K-MD2 FMCW radar in an anechoic chamber. This step is crucial for enabling true operational robustness and quantifying the domain shift between the RADial radar sensor and K-MD2. This characterisation yielded mean RCS estimates of 44.5 m^2 and 35.6 m^2 for two different cars, and 0.565 m^2 for a pedestrian, all consistent with values reported in the literature. While the far-field condition for the target could not be strictly met, it was shown that an appropriate relaxation yields valid median RCS estimates for large targets. This characterisation has helped us to understand physical phenomena and signal artefacts that any radar perception pipeline operating on such data will inevitably encounter. However, the cross-sensor gap introduced by differences in operating frequency, antenna configuration, and spatial resolution between the two sensors remains an open research question, and addressing it represents a key direction for future work.

Finally, while radar offers unique advantages in robustness to adverse conditions and direct measurement of both velocity and range, combining it with complementary modalities, such as a camera, which provides richer semantic and visual information, represents a natural extension of this work. To this end, the present master’s thesis and the parallel work of Léonel Ngague, which develops a radar-camera fusion model for object detection, are deeply complementary: together, they address the full perception pipeline from single-sensor radar detection to multi-modal fusion, and collectively constitute a promising foundation for the development of more robust and complete autonomous driving perception systems.

Appendix A

RadViT Architecture Diagrams

This appendix provides architectural diagrams of RadViT’s core components, namely the Transformer encoder block (Figure A.1), the Range-Angle decoder block (Figure A.2), and the detection head block (Figure A.3).

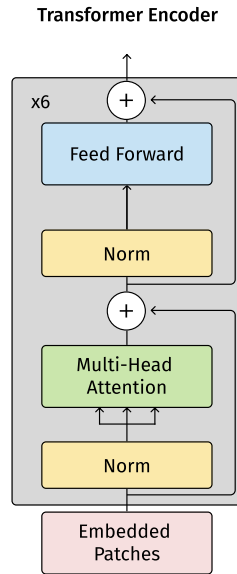


Figure A.1: Transformer Encoder block

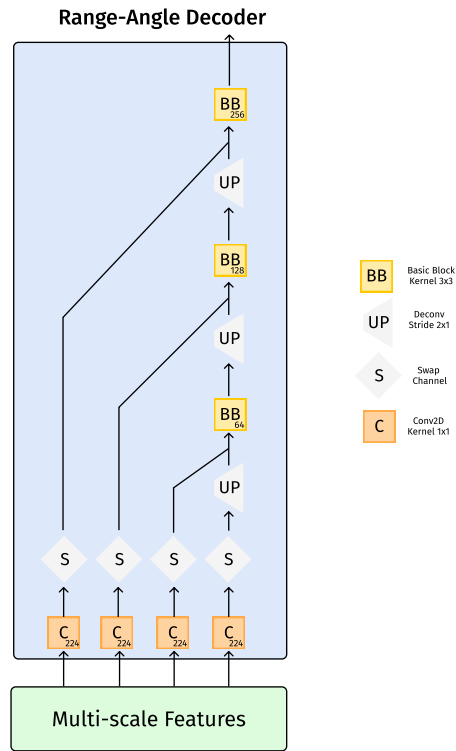


Figure A.2: Range-Angle decoder block

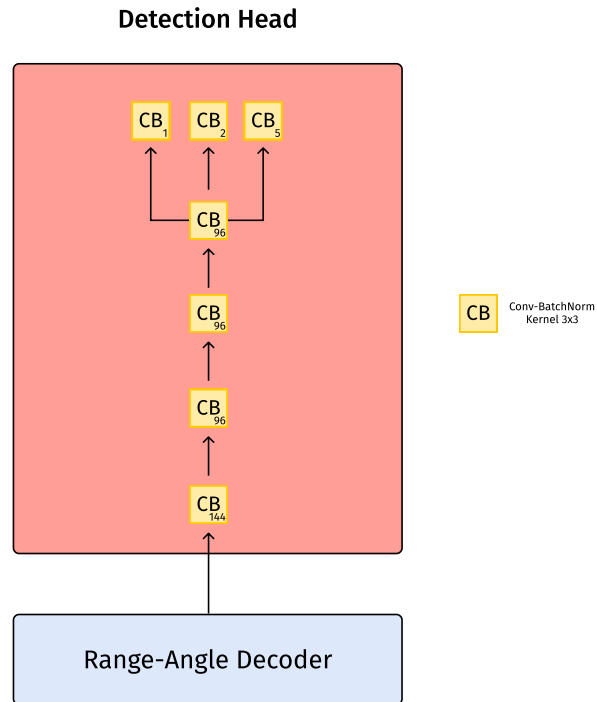


Figure A.3: Detection Head block

Appendix B

Complementary results

Model	Radar Input	AP(%) $\uparrow$	AR(%) $\uparrow$	F1(%) $\uparrow$	R(m) $\downarrow$	A($^{\circ}$) $\downarrow$
FFT-RadNet	RD	98.5	91.7	95.0	0.10	0.13
T-FFTRadNet	RD - Shift	93.5	95.5	94.4	0.15	0.11
Pixor	PC	99.0	28.8	44.7	0.15	0.19
Pixor	RA	96.9	88.0	92.2	0.09	0.16
RadViT	RD	84.5	80.8	82.6	0.16	0.11
RadViT	ADC	83.4	80.7	82.0	0.18	0.12

Table B.1: easy cases

Model	Radar Input	AP(%) $\uparrow$	AR(%) $\uparrow$	F1(%) $\uparrow$	R(m) $\downarrow$	A($^{\circ}$) $\downarrow$
FFT-RadNet	RD	92.9	64.8	76.3	0.13	0.26
T-FFTRadNet	RD - Shift	86.8	75.1	80.6	0.17	0.13
Pixor	PC	93.3	38.7	54.7	0.19	0.33
Pixor	RA	95.9	70.1	81.0	0.12	0.27
RadViT	RD	80.8	54.2	64.9	0.20	0.14
RadViT	ADC	74.9	50.0	60.0	0.22	0.15

Table B.2: hard cases

Appendix C

Detection output visualisation

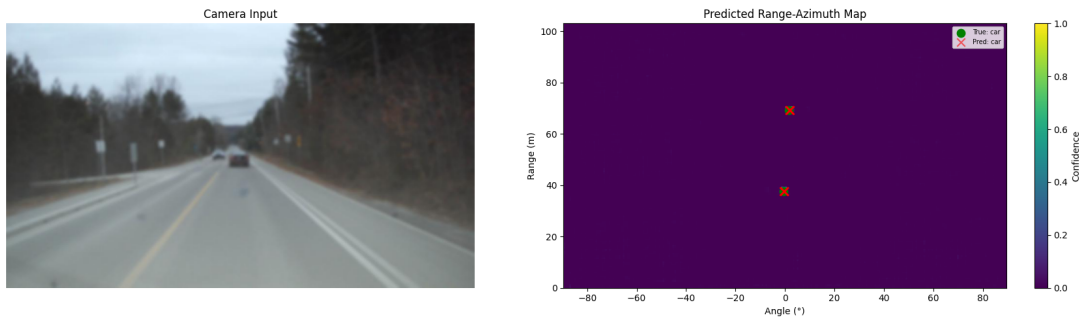


Figure C.1: True Positive detection example. The predicted target (red cross) correctly matches the ground-truth object (green circle) on the Range-Azimuth confidence map, alongside the corresponding camera view.

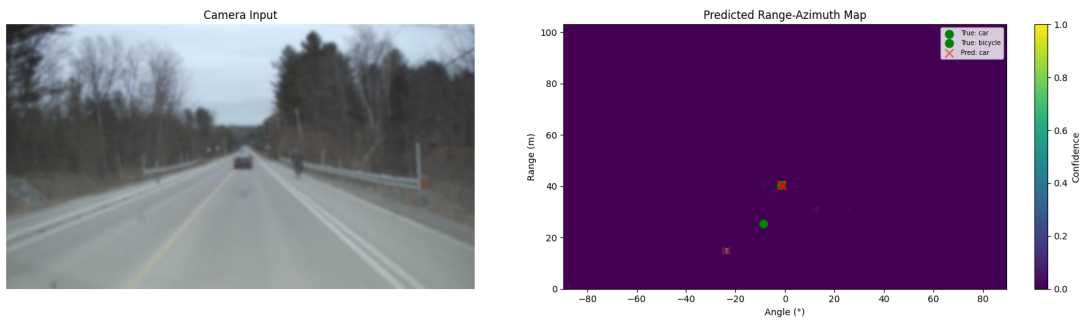


Figure C.2: False Positive detection example. RadViT produces a detection (red cross) with no corresponding ground-truth object (green circle) in the Range-Azimuth confidence map, indicating a spurious detection.

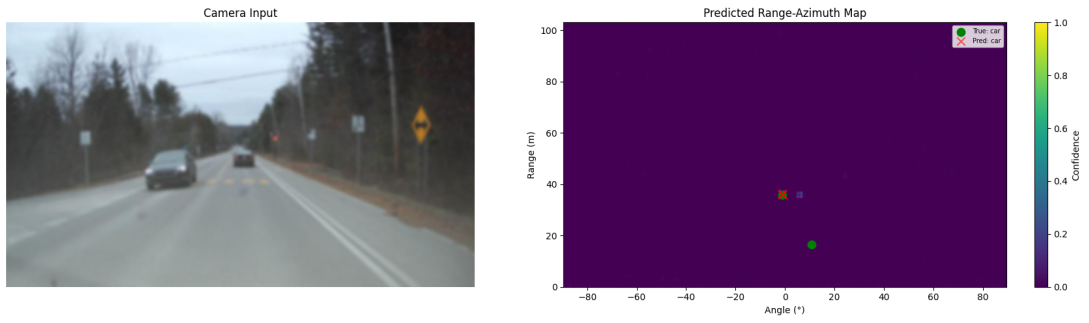


Figure C.3: False Negative detection example. A ground-truth object (green circle) is present in the scene but is not detected by RadViT, resulting in a missed detection on the Range-Azimuth confidence map.

Appendix D

Declaration on the use of AI

In compliance with the guidelines governing the use of generative AI tools, we declare that Claude.ai, NotebookLM, and ChatGPT were used to assist in drafting and reformulating parts of this thesis. Generative AI tools were additionally used to support the development of selected portions of the project's source code.

Bibliography

- [1] D. Wu, F. Yang, B. Xu, P. Liao, and B. Liu, *A survey of deep learning based radar and vision fusion for 3d object detection in autonomous driving*, 2024. arXiv: 2406.00714 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2406.00714>.
- [2] World Health Organization, *Road traffic injuries*, <https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>, Accessed: May 10, 2026, Dec. 2023.
- [3] Statbel – Statistics Belgium, *Accidents de la circulation*, <https://statbel.fgov.be/fr/themes/mobilite/circulation/accidents-de-la-circulation>, Service Public Fédéral Économie. Accessed: May 10, 2026, Sep. 2025.
- [4] Service Public Fédéral Mobilité et Transports, *Les systèmes d’aide à la conduite (ADAS) – introduction générale aux ADAS*, <https://mobilit.belgium.be/fr/route/vehicules/systemes-daide-la-conduite-adas/introduction-generale-aux-adas>, Accessed: May 10, 2026, Apr. 2026.
- [5] YellowScan, *Surprising LiDAR penetration capabilities through different surfaces*, <https://www.yellowscan.com/knowledge/surprising-lidar-penetration-capabilities-through-different-surfaces/>, Accessed: 2026, Jul. 2025.
- [6] E. Desjardin, “Camera, radar and lidar sensor synchronization and dataset creation for traffic monitoring,” M.S. thesis, Université Catholique de Louvain, 2023-2024.
- [7] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, “K-radar: 4d radar object detection for autonomous driving in various weather conditions,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 3819–3829, 2022.
- [8] K. Shi *et al.*, “Radar and camera fusion for object detection and tracking: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 28, pp. 3478–3520, 2026. DOI: 10.1109/COMST.2025.3599596.
- [9] G. Jocher and J. Qiu, *Ultralytics YOLO*, 2024. DOI: 10.5281/zenodo.264818686. [Online]. Available: <https://zenodo.org/badge/latestdoi/264818686>.
- [10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, *End-to-end object detection with transformers*, 2020. arXiv: 2005.12872 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2005.12872>.

- [12] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: Fast encoders for object detection from point clouds,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.
- [13] S. Yao *et al.*, “Exploring radar data representations in autonomous driving: A comprehensive review,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 6, pp. 7401–7425, Jun. 2025, ISSN: 1558-0016. DOI: 10.1109/tits.2025.3554781. [Online]. Available: <http://dx.doi.org/10.1109/TITS.2025.3554781>.
- [14] C. Waldschmidt, C. Bonfert, and T. Grebner, “Radar types and modulation schemes,” in *Millimeter Wave Radar: Hardware and Signal Processing*. Cham: Springer Nature Switzerland, 2025, pp. 51–82, ISBN: 978-3-031-89118-2. DOI: 10.1007/978-3-031-89118-2_4. [Online]. Available: https://doi.org/10.1007/978-3-031-89118-2_4.
- [15] J. Rebut, A. Ouaknine, W. Malik, and P. Pérez, “Raw high-definition radar for multi-task learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 021–17 030.
- [16] J. Mao, S. Shi, X. Wang, and H. Li, “3d object detection for autonomous driving: A comprehensive survey,” *International Journal of Computer Vision*, vol. 131, pp. 1–55, Apr. 2023. DOI: 10.1007/s11263-023-01790-1.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [18] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.
- [19] A. Dosovitskiy *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [20] Sharkyun, *Computer vision object detection: One-stage vs two-stage*, <https://sharkyun.medium.com/computer-vision-object-detection-one-stage-vs-two-stage-b05dbff88195>, Accessed: May 13, 2026.
- [21] A. Sarkar, *Anchor-based vs anchor-free object detection*, <https://www.abhik.ai/concepts/computer-vision/anchor-based-vs-anchor-free>, Accessed: May 13, 2026, Dec. 2024.
- [22] R. E. Kalman, “A new approach to linear filtering and prediction problems,” 1960.
- [23] M. Liu *et al.*, “A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook,” *IEEE Transactions on Intelligent Vehicles*, 2024.
- [24] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 3354–3361.
- [25] L. Zheng *et al.*, “Tj4dradset: A 4d radar dataset for autonomous driving,” in *2022 IEEE 25th international conference on intelligent transportation systems (ITSC)*, IEEE, 2022, pp. 493–498.

- [26] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, “Radiate: A radar dataset for automotive perception in bad weather,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 1–7.
- [27] A. Ouaknine, A. Newson, J. Rebut, F. Tupin, and P. Pérez, “Carrada dataset: Camera and automotive radar with range-angle-doppler annotations,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 5068–5075.
- [28] A. Zhang, F. E. Nowruzi, and R. Laganier, “Raddet: Range-azimuth-doppler based radar object detection for dynamic road users,” in *2021 18th Conference on Robots and Vision (CRV)*, IEEE, 2021, pp. 95–102.
- [29] H. Caesar *et al.*, “Nuscenes: A multimodal dataset for autonomous driving,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.
- [30] O. Schumann *et al.*, “Radarscenes: A real-world radar point cloud data set for automotive applications,” in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, IEEE, 2021, pp. 1–8.
- [31] S. Yao *et al.*, “Waterscenes: A multi-task 4d radar-camera fusion dataset and benchmarks for autonomous driving on water surfaces,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 11, pp. 16 584–16 598, 2024.
- [32] T.-Y. Lim, S. A. Markowitz, and M. N. Do, “Radical: A synchronized fmcw radar, depth, imu and rgb camera data dataset with low-level fmcw radar signals,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 941–953, 2021.
- [33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [34] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.
- [35] B. Yang, I. Khatri, M. Happold, and C. Chen, *Adcnet: Learning from raw radar data via distillation*, 2023. arXiv: 2303.11420 [eess.SP]. [Online]. Available: <https://arxiv.org/abs/2303.11420>.
- [36] J. Giroux, M. Bouchard, and R. Laganier, *T-fftradnet: Object detection with swin vision transformers from raw adc radar signals*, 2023. arXiv: 2303.16940 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2303.16940>.
- [37] P. Zhao, C. X. Lu, B. Wang, N. Trigoni, and A. Markham, *Cubelearn: End-to-end learning for human motion recognition from raw mmwave radar signals*, 2021. arXiv: 2111.03976 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2111.03976>.
- [38] Y.-J. Li, S. Hunt, J. Park, M. O’Toole, and K. Kitani, “Azimuth super-resolution for fmcw radar in autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 504–17 513.

- [39] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.
- [40] C. Decourt, R. VanRullen, D. Salle, and T. Oberlin, “Darod: A deep automotive radar object detector on range-doppler maps,” in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 112–118. DOI: 10.1109/IV51971.2022.9827281.
- [41] B. Major *et al.*, “Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct. 2019.
- [42] D.-H. Paek, S.-H. Kong, and K. T. Wijaya, “K-radar: 4d radar object detection for autonomous driving in various weather conditions,” in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. [Online]. Available: https://openreview.net/forum?id=W_bsDmzwaZ7.
- [43] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.
- [44] Y. Zhou and O. Tuzel, *Voxelnet: End-to-end learning for point cloud based 3d object detection*, 2017. arXiv: 1711.06396 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1711.06396>.
- [45] Y. Yan, Y. Mao, and B. Li, “Second: Sparsely embedded convolutional detection,” *Sensors*, vol. 18, no. 10, p. 3337, 2018.
- [46] B. Yang, W. Luo, and R. Urtasun, *Pixor: Real-time 3d object detection from point clouds*, 2019. arXiv: 1902.06326 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1902.06326>.
- [47] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015. arXiv: 1512.03385 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1512.03385>.
- [48] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/1810.04805>.
- [49] Q. Wang *et al.*, “Learning deep transformer models for machine translation,” in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1810–1822.
- [50] A. Vaswani *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [51] A. Baevski and M. Auli, *Adaptive input representations for neural language modeling*, 2019. arXiv: 1809.10853 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/1809.10853>.
- [52] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, *Feature pyramid networks for object detection*, 2017. arXiv: 1612.03144 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/1612.03144>.

- [53] Z. Liu *et al.*, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.
- [54] R. Ranftl, A. Bochkovskiy, and V. Koltun, *Vision transformers for dense prediction*, 2021. arXiv: 2103.13413 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2103.13413>.
- [55] T.-Y. Lin *et al.*, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, Springer, 2014, pp. 740–755.
- [56] R. Hussein, S. Lee, and R. Ward, “Multi-channel vision transformer for epileptic seizure prediction,” *Biomedicines*, vol. 10, no. 7, p. 1551, 2022.
- [57] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.
- [58] M. Khodarahmi and V. Maihami, “A review on kalman filter models,” *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 727–747, 2023.
- [59] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [60] C. Heindl, *Py-motmetrics: Benchmark multiple object trackers (mot) in python*, 2019. [Online]. Available: <https://github.com/cheind/py-motmetrics>.
- [61] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *2016 IEEE international conference on image processing (ICIP)*, Ieee, 2016, pp. 3464–3468.
- [62] F. Ledent, “Synchronization of multimodal data flows for real-time ai analysis,” M.S. thesis, Université Catholique de Louvain, 2021-2022.
- [63] RFbeam Microwave GmbH, *K-MD2 Engineering Sample Radar Transceiver Datasheet*, https://rfbeam.ch/wp-content/uploads/dlm_uploads/2022/10/K-MD2_Datasheet.pdf, Revision A, Jul. 2018. [Online]. Available: https://rfbeam.ch/wp-content/uploads/dlm_uploads/2022/10/K-MD2_Datasheet.pdf.
- [64] H.-S. Kim *et al.*, “Miniaturized 24 ghz fmcw radar leakage mitigation with synchronized subtraction,” *IEEE Sensors Journal*, vol. 24, no. 16, pp. 25 701–25 714, Aug. 2024, ISSN: 1558-1748. DOI: 10.1109/JSEN.2024.3422397.
- [65] C. Larsson, M. Gustafsson, and G. Kristensson, “Calibration methods for wideband forward rcs measurements,” in *Proceedings of the Fourth European Conference on Antennas and Propagation*, 2010, pp. 1–5.
- [66] R. Kouyoumjian and L. Peters, “Range requirements in radar cross-section measurements,” *Proceedings of the IEEE*, vol. 53, no. 8, pp. 920–928, 1965. DOI: 10.1109/PROC.1965.4070.
- [67] M. Ezuma, M. Funderburk, and I. Guvenc, “Compact-range rcs measurements and modeling of small drones at 15 ghz and 25 ghz,” in *2020 IEEE Radio and Wireless Symposium (RWS)*, 2020, pp. 313–316. DOI: 10.1109/RWS45077.2020.9050049.

- [68] RFbeam Microwave GmbH, *Understanding detection range of radar sensors*, <https://rfbeam.ch/understanding-detection-range-of-radar-sensors/>, Accessed: 2026-04-24, 2025.
- [69] D. Kurniawan, C. Bin Ali Wael, T. Miftahushudur, and O. Heriana, “Implementation of automatic i/q imbalance correction for fmcw radar system,” in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2017, pp. 100–105. DOI: 10.1109/ICITISEE.2017.8285474.
- [70] Q. Chaudhari, *FMCW radar part 3 – design guidelines*, <https://wirelesspi.com/fmcw-radar-part-3-design-guidelines/>, Wireless Pi. Accessed: 2026, Nov. 2023.
- [71] O. Bialer and Y. Haitman, “Radsimreal: Bridging the gap between synthetic and real data in radar object detection with simulation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 407–15 416.
- [72] H. Khodabakhshandeh, T. Visentin, R. Hernangómez, and M. Pütz, “Domain adaptation across configurations of fmcw radar for deep learning based human activity classification,” in *2021 21st International Radar Symposium (IRS)*, 2021, pp. 1–10. DOI: 10.23919/IRS51887.2021.9466179.
- [73] M. Zhang, S. Abdulatif, B. Loesch, M. Altmann, M. Schwarz, and B. Yang, “Exploring domain shift on radar-based 3d object detection amidst diverse environmental conditions,” in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2024, pp. 3513–3518.
- [74] Y. Zhang *et al.*, “Bytetrack: Multi-object tracking by associating every detection box,” in *European conference on computer vision*, Springer, 2022, pp. 1–21.
- [75] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, “Motr: End-to-end multiple-object tracking with transformer,” in *European conference on computer vision*, Springer, 2022, pp. 659–675.
- [76] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, “Trackformer: Multi-object tracking with transformers,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8844–8854.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl