UNIVERSITY OF GENOA, ITALY

DITEN - Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture

Ph.D. in Science and Technology for Electronic and Telecommunication Engineering – Cycle XXXIV (2018-2021)

Development of a Cost-Efficient Multi-Target Classification System Based on FMCW Radar for Security Gate Monitoring

Ph.D. Thesis

This dissertation is submitted for the degree of

Doctor of Philosophy

Ph.D. Candidate: Ali Rizik

Tutors: Prof. Daniele Caviglia

Coordinator of the Ph.D. Course: Prof. Maurizio Valle

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. This dissertation contains less than 65,000 words including appendices, bibliography, footnotes, tables, and equations, and has less than 150 figures.

Ali Rizik

“If you cannot explain it simply, you don’t understand it well enough.”

Albert Einstein

Acknowledgments

This Dissertation is the result of my research work conducted as a Ph.D. student at the University of Genoa. The work was carried out in cooperation with the company of Darts engineering Srl.

First of all, I would like to thank my primary supervisor, Prof. Daniele Caviglia, for giving me the opportunity to work on this project which has been a life cherishing experience. He was always kind and patient to guide me step-by-step to finish this project. I am most grateful to him for making this Ph.D. both educational and enjoyable and for his positive support and guidance throughout.

Secondly, I greatly thank my second primary supervisor Dr. Hussein Chible, for opening this Ph.D. opportunity for me and giving me the chance to participate in the Ph.D. program. I am most grateful for his continued support and encouragement throughout the Ph.D. period.

Additionally, I want to thank my daily supervisor, Prof. Andrea Randazzo, for his assistance during this thesis project. He has provided me with great support and valuable guidance. He encouraged me to try some new ideas and helped me to solve the problems. I am very grateful for his encouragement, support, and his constant help in planning and reviewing my work.

I also thank Dr. Emanuele Tavanti for the inspiring discussions we had, the honest feedback we shared, and the mutual support we gave each other. Working with you really eased the whole process and made it fun to come to work.

Finally, special thanks to Alessandro Delucchi and Roberto Vio (Darts Engineering Srl, Genoa, Italy) for their kind support during the development phase and the experimental activities presented in this thesis.

Abstract

Radar systems have a long history. Like many other great inventions, the origin of radar systems lies in warfare. Only in the last decade, radar systems have found widespread civil use in industrial measurement scenarios and automotive safety applications. Due to their resilience against harsh environments, they are used instead of or in addition to optical or ultrasonic systems. Radar sensors hold excellent capabilities to estimate distance and motion accurately, penetrate non-metallic objects, and remain unaffected by weather conditions. These capabilities make these devices extremely flexible in their applications. Electromagnetic waves centered at frequencies around 24 GHz offer high precision target measurements, compact antenna, and circuitry design, and lower atmospheric absorption than higher frequency-based systems.

This thesis studies non-cooperative automatic radar multi-target detection and classification. A prototype of a radar system with a new microwave-radar-based technique for short-range detection and classification of multiple human and vehicle targets passing through a road gate is presented. It allows identifying different types of targets, i.e., pedestrians, motorcycles, cars, and trucks. The developed system is based on a low-cost 24 GHz off-the-shelf FMCW radar, combined with an embedded Raspberry Pi PC for data acquisition and transmission to a remote processing PC, which takes care of detection and classification. This approach, which can find applications in both security and infrastructure surveillance, relies upon the processing of the scattered-field data acquired by the radar.

The developed method is based on an ad-hoc processing chain to accomplish the automatic target recognition task, which consists of blocks performing clutter and leakage removal with a frame subtraction technique, clustering with a DBSCAN approach, tracking algorithm based on the *α-β* filter to follow the targets during traversal, features extraction, and finally classification of targets with a classification scheme based on support vector machines. The approach is validated in real experimental scenarios, showing its capabilities in correctly detecting multiple targets belonging to different classes (i.e., pedestrians, cars, motorcycles, and trucks). The approach has been validated with experimental data acquired in different scenarios, showing good identification capabilities.

Contents

List of Figures

List of Tables

Publications

I. Papers:

1. A. Rizik, A. Randazzo, R. Vio, A. Delucchi, H. Chible and D. D. Caviglia, "*Feature Extraction for Human-Vehicle Classification in FMCW Radar*," 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2019, pp. 131-132.

2. A. Rizik, A. Randazzo, R. Vio, A. Delucchi, H. Chible and D. D. Caviglia, "*Low-Cost FMCW Radar Human-Vehicle Classification Based on Transfer Learning*," 2020 32nd IEEE International Conference on Microelectronics (ICM), 2020, pp. 1-4.

3. A. Rizik, E. Tavanti, R. Vio, A. Delucchi, H. Chible, A. Randazzo, D. D. Caviglia, "*Single Target Recognition Using a Low-Cost FMCW Radar Based on Spectrum Analysis*" 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2020, pp. 1-4,

4. A. W. Daher, A. Rizik, M. Muselli, H. Chible, and D. D. Caviglia, "*Porting Rulex Machine Learning Software to the Raspberry Pi as an Edge Computing Device,*" in Applications in Electronics Pervading Industry, Environment and Society, Cham, 2021, pp. 273–279.

II. Journals:

1. A. Rizik, E. Tavanti, H. Chible, D. D. Caviglia and A. Randazzo, *"Cost-Efficient FMCW Radar for Multi-Target Classification in Security Gate Monitoring,"* in IEEE Sensors Journal, vol. 21, no. 18, pp. 20447-20461, 15 Sept.15, 2021.

2. Ali Walid Daher, Ali Rizik, Andrea Randazzo, Emanuele Tavanti, Hussein Chible, Marco Muselli, Daniele D. Caviglia, *"Pedestrian and Multi-Class Vehicle Classification in Radar Systems using Rulex Software on the Raspberry Pi"* Appl. Sci. 2020, 10(24), 9113.

3. A. W. Daher, A. Rizik, M. Muselli, H. Chible, and D. D. Caviglia, “Porting Rulex Software to the Raspberry Pi for Machine Learning Applications on the Edge,” *Sensors*, vol. 21, no. 19, p. 6526, Sep. 2021.

4. E. Tavanti, A. Rizik, A. Fedeli, D. D. Caviglia and A. Randazzo, "A Short-Range FMCW Radar-based Approach for Multi-Target Human-Vehicle Detection," in IEEE Transactions on Geoscience and Remote Sensing, doi: 10.1109/TGRS.2021.3138687.

5. Mohanna, Ammar, Christian Gianoglio, Ali Rizik, and Maurizio Valle. 2022. "A Convolutional Neural Network-Based Method for Discriminating Shadowed Targets in Frequency-Modulated Continuous-Wave Radar Systems" *Sensors* 22, no. 3: 1048. https://doi.org/10.3390/s22031048

Chapter 1: Introduction

Knowledge and understanding of our surrounding environment are very important to protect ourselves from external interference. For many years humanity has constructed defence systems to protect their goods. Surveillance of our surrounding environment is an important aspect of defence, as it helps protectors prepare helps for any unwelcome and sudden interference into their territory. Surveillance systems may also have many civil applications. For instance, fire preventing and extinguishing is more efficient if the number of persons and their positions is known inside buildings. Also, automatic (using sensors) door-opening and light switching devices could save resources. Observation of traffic conditions may help drivers avoid a collision. However, surveillance performed by humans may be inaccurate due to the "human factor", which may cause unpredictable consequences. Moreover, human sense of the surrounding environment is limited by personal abilities. The "human factor" may be reduced by auxiliary devices or assistant systems.

During the last 70 years, progress in electronics made it possible to build a computer, a device that utilizes logic and arithmetic-based operations. Further development helped to construct powerful computing units, which surpassed humans in logic and arithmetic calculations. One of the computer science branches, artificial intelligence, studies how to develop smart systems that could make a decision depending on input parameters. Success in artificial intelligence over the last 30 years has made it possible to build semi- and fully automatic decision-making systems. One of these systems is an automatic surveillance system that can detect, determine the class, and estimate the parameters of the objects around.

One of the advanced automatic surveillance technologies allows determination of the type of object, which is performed by automatic target classification. Such a system performs surveillance on large areas and for multiple target scenarios simultaneously. It can, for example, estimate the number and position of persons inside buildings, estimate traffic conditions, perform automatic door opening and light switching, warn of invasion to the area, etc. The "sensing", or environment understanding is done by devices called sensors. Some sensors can overcome human abilities in sensing and, therefore, such systems have high potential in automatic surveillance.

In recent years, there has been a growing interest in the development of reliable monitoring and surveillance devices, to be used in urban areas and near-critical zones [1]–[4]. For example, the emergence of autonomous driving cars along with the ongoing advancements in autonomous vehicles technology drove the research for developing a pedestrian recognition system to ensure pedestrians safety [5]–[8]. In addition, the continuously increasing threat of national terrorism and criminal acts is now considered the main driving force for researchers to develop more reliable surveillance systems to be installed in urban environments [1]. A significant push to the development of new systems is also given by the advancements in several applicative scenarios, such as subsurface prospection [9]–[11], non-destructive testing [12], [13], and transportation infrastructure monitoring [14]–[16]. Optical and radar sensors are nowadays considered the most promising technologies to be used in such systems [17]. Optical sensors measure the energy of optical light. Such sensors are low-cost and can reconstruct a picture seen by the human eye. It makes it easy to recognize objects in good light conditions and absolves obstacles. Systems based on camera sensors could be used for a wide view, short-range surveillance, or narrow view, far-field surveillance. However, good performance is achieved with good weather conditions only. Fog or rain may reduce their efficiency. Moreover, powerful hardware is required due to the processing of large data amounts. Radar sensors, on the other hand, are based on the reflection of electromagnetic waves. Radar devices are very well known for their high detection capabilities. They are also capable of providing a very precise measurements of the target's velocity, range, angle of arrival, and direction of motion. Other advantages of radar sensors include almost all weather and light condition operations, capability to penetrate through objects, in addition to short and long operation distance.

In this framework, short-range radars [18], [19] are particularly interesting because of their robustness against adverse weather conditions and non-sensitivity to lighting conditions [20], [21], problems that can severely affect video-based devices [22]. Different types of radars were used in target detection and classification applications. In particular, the Frequency Modulated Continuous Wave (FMCW) radar technology has been widely adopted in the production of cost-effective and compact systems for several applications [23]–[33], and in particular in automotive applications [34]–[39]. On the one hand, these radars do not suffer from severe blind range issues that normally affect monostatic pulsed radars [40]. On the other hand, they are generally cheaper and can cover bigger areas than Light Detection and Ranging (LIDAR) and Long-Wave Infrared (LWIR) devices [21]. they can simultaneously provide both range and velocity. Such information can be obtained through different processing schemes, e.g., by using a 2D FFT technique [41]. Moreover, unlike pulse and Ultra-Wide Band (UWB) radars, FMCW systems require lower sampling rates and lower peak-to-average power ratio to detect the distance and speed of multiple moving targets [42], [43].

1.1 State-of-art:

Many different approaches for FMCW radar target classification have been presented in the literature, where several solutions have been developed in [14], [20], [21], [44]–[60]. Often, the proposed approaches involve the use of Machine Learning methods (ML) [61], such as Support Vector Machines (SVM) or Deep Learning Neural Networks (DLNN). Some of the adopted classical machine learning techniques deal with features extracted from the FFT spectrums or the Range-Doppler maps. For example, authors in [62] and [52] presented a human-vehicle classification system based on a 24 GHz FMCW radars sensor, and a support vector machine classifier (SVM) [63]. The system showed a good classification performance using features extracted from the range and velocity profiles of the targets. Another human-vehicle classification system based on SVM and a 77 GHz FMCW radar sensor was introduced in [53], where features based on the Radar Cross Section (RCS) were used [64]. In that regard, a recursive Bayesian solution for the problem of joint tracking and classification of airborne targets using the RCS as the key feature was presented in [65]. Another Joint tracking and classification based on the aerodynamic model and radar cross-section was presented in [66]. The proposed methodologies, according to simulations, showed a good performance.

A multiclass-target classification method using an advanced mmwave FMCW radar operating in the frequency range of 77 - 81 GHz, based on custom range-angle heatmaps and machine learning tools has been proposed in [67]. The proposed technique achieves an accuracy of 97.6

% and 99.6 % for classifying the UAV and humans, respectively, and an accuracy of 98.1 % for classifying the car from the range-angle FOV (Field Of View) heatmap. In addition, a vehicle classification system based on radar measurements was presented in [56], where height and length profile-based features were used.

Deep learning classification techniques applied on radar measurements for radar target classification have been adopted in [68]–[70]. A moving target classification system based on automotive radar and deep neural networks was presented in [60]. Another radar target classification system based on FMCW radar and Recurrent Neural Networks (RNN) were presented in [71] and [72]. However, unlike classical machine learning techniques, for optimal performance, deep learning networks usually require a huge amount of data to perform the training step [55]. Indeed, this implies the need for a big memory capacity, especially with the largest networks. This puts some limits on the required hardware specifications to run deep learning algorithms with big training data, especially at the GPU (Graphical Processing Unit) level. Also, a bigger training set means that much more training time is required. Moreover, works based on Synthetic Aperture Radar (SAR) [73] and Inverse Synthetic Aperture Radar (ISAR) techniques [74] have been reported, too in [14], [57].

Alternative methods for the classification of ground targets are based on the micro-Doppler signatures [59], [60], [75]. The micro-Doppler content arises in radar returns due to the Doppler effect caused by micromotions [76]. All motions in the scene affect the frequency of the radar's transmitted wave (as well as all other reflected waves) by changing it, with the value of the frequency shift proportional to the radial velocity. The time-varying behaviour of micro-motions forms a micro-Doppler signature in a form of Frequency Modulated (FM) multi-component signals. The micro-Doppler phenomenon arises with the motion of a non-rigid body or a complex motion when the global motion consists of several local micro-motions. Micro-Doppler signature is a characteristic of motion preserved for a special type of objects. Human activities, generally, involve different body parts with different motion structures. Therefore, these activities could be analysed by micro-Doppler signatures. A Human-Robot classification system based on 25 GHz FMCW radar using micro-Doppler features was introduced in [77]. Another target classification system based on micro-Doppler signatures was introduced in [78]. An additional micro-Doppler-based target classification approach combined with an SVM classifier was presented in [79]. Although micro-Doppler features proved to be good in radar target classification, they are difficult to extract. Nevertheless, the extraction of a micro-Doppler signature usually requires a quite long illumination of the targets, along with a large number of consequent observations to be processed. This comes with consequent practical issues in the presence of relatively fast targets like cars and motorcycles [54]. This fact makes it impossible to extract such features with low-cost radar devices.

Despite the great advances in these fields, the radar-based ground surveillance with Automatic Target Recognition (ATR) capability for single or simultaneous multiple targets still represents a quite challenging problem. Indeed, many of the methods reported in literature allow the presence of only a single target at a time in the monitored area, and systems able to eventually manage multiple targets simultaneously are usually limited to the estimation of the positions and velocities, without ATR capabilities. Finally, compact, and low-cost systems are usually required, leading to further limitations in terms of computational resources and achievable radar resolutions. Consequently, there is the need for novel detection techniques, specifically tailored to low-end devices.

1.2 Thesis Objectives:

The objectives of this thesis project are as follows:

1. Looking deeply into the possibility of detecting and classifying targets passing through a security gate using a system based on low-cost radar hardware.
2. Building the system's hardware with the best hardware tools suited for the required application
3. Design a novel processing chain for multi-target classification
4. Building a sufficient data set on which the system can be trained on
5. Training, testing, and optimizing the system performance in an outdoor environment.
6. Validating the system performance in a different environment with different environmental conditions.

The developed system's performance was tested and validated in two different environments, and it showed a good classification performance in classifying single and multiple targets passing in the FOV of the radar. These results will be further discussed in detail in the next chapters.

1.3 Overview of the Proposed System:

In this framework, a new short-range surveillance technique based on low-cost FMCW radar technology, aimed at overcoming the significant limitations in terms of resolution, acquisition speed, and ambiguity that afflict this hardware platform is proposed. In particular, a novel ad-hoc processing chain able to perform the detection and classification of multiple targets in a cluttered environment has been specifically developed. The proposed system consists of a Distance2Go radar board developed by Infineon technologies [80], a Raspberry Pi 3B+ mini-PC, and a portable PC running MATLAB software. The radar board continuously transmits the radar signal and receives echo signals reflected by the targets. The Raspberry Pi collects the

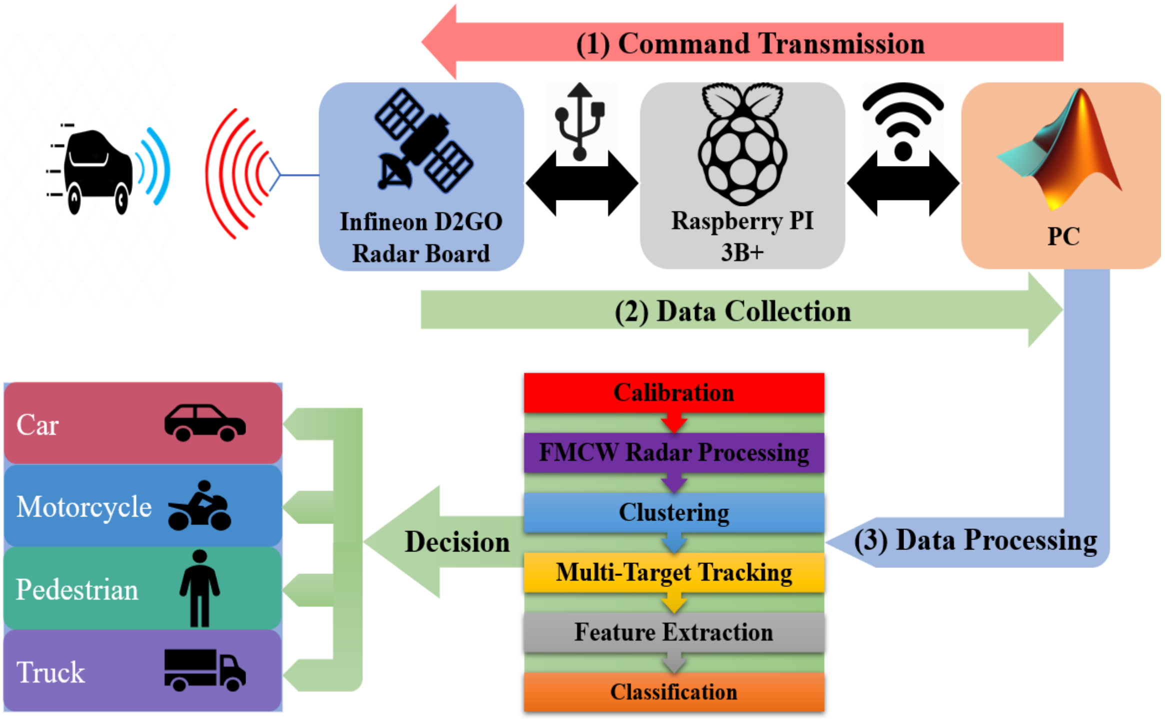


Figure 1.1 Overview of the system hardware and software architecture

target's echo signals and transmits them to the remote PC for processing. Firstly, clutter and leakage removal is provided by a modified Background Subtraction (BS) technique, which is based on the BS technique used in image processing [81]. Thereafter, a stretch processing approach is adopted for computing the range-Doppler maps for each collected data frame. To isolate the contributions of each target on these maps, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) method [82] is employed, which assigns a numerical tag to each detected cluster. After that, a specifically-designed multi-target tracking algorithm, based on the α-β filter [83], [84], and a proper tracking scheme [18], [40], [85], that assigns a unique target-ID to each target is used to track the targets among the different frames. The tracking filter provides the feature extraction block with the necessary information needed to later classify the target as either a truck, motorcycle, car, or pedestrian using a multi-class SVM. This whole procedure is summarized in Figure 1.1. The proposed system was tested in two different cluttered environments. The cost of the measurement setup presented in this paper is around just 300 € (radar board, ~200 €; Raspberry PI 3B+, ~50 €, optional camera, ~10 €; power bank along with the boxing and the other expenses, ~50 €), which is one of the key features of the system.

1.4 Contributions:

The proposed setup permits the use of low-cost radar hardware in a quite complex and advanced radar application, i.e., the detection and classification of multiple targets passing through a gate. It also offers the option to perform all the processing schemes in real-time, either on the Raspberry Pi embedded PC or using a portable PC. The possibility of performing some simple online processing tasks, like FMCW processing, directly on the radar board adds additional flexibility to the system. This is all offered in cost-efficient, portable, and power-friendly hardware.

1.5 Thesis Structure:

The thesis is structured as follows:

- Chapter 2 covers the parameter selection of the FMCW radar system, as well as hardware and software description.
- Chapter 3 covers the basic signal model and processing technique used in the FMCW radar system. It also covers the different pre-processing algorithms applied to the received target signal before tracking and classification.
- Chapter 4 covers in detail the different aspects of the developed multi-target tracking filter and shows its performance in some real-world scenarios.
- In chapter 5 a description of the chosen features is presented along with a detailed explanation of how they were extracted.
- The performance of the system in real-world scenarios is addressed in chapter 6.
- The last chapter lists the conclusion and future works.

Chapter 2: System Architecture

This chapter describes the hardware and software architectures of the developed gate monitoring system. First, the hardware components and the methods of their interconnection are described. Subsequently, the operating logic of the developed software and the details of its various processing blocks are illustrated.

2.1 Hardware Architecture:

The developed system mainly consists of three components (Figure 2.1 shows the prototype made, while Figure 2.2 schematizes the connections of the components):

- A radar board
- A Raspberry PI 3B+ embedded PC
- A control PC

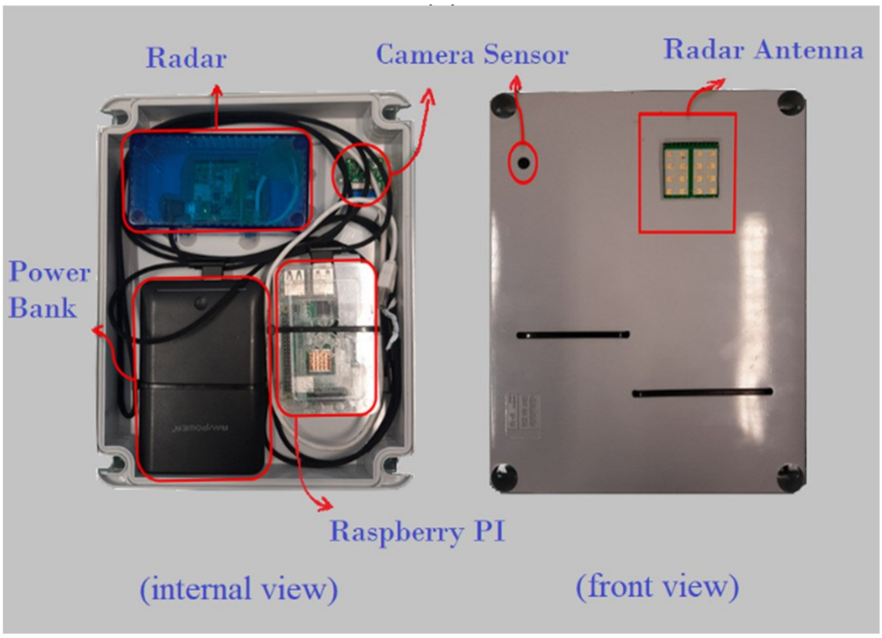


Figure 2.1 Hardware prototype of the gate monitoring system

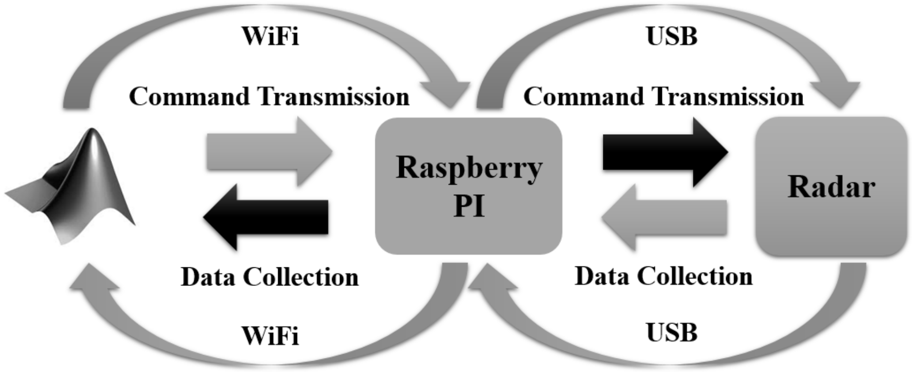


Figure 2.2 Diagram of the connections between the main blocks making up the system

We can notice that Figure 2.1 also shows a camera sensor included in the system. The camera was used only during the data acquisition phase to take a photo of the passing target. The radar device is based on the Distance2Go demo board developed by Infineon [80], which is a credit card sized board, and is composed of two parts:

- The radar main board
- The debugger board

The radar main board contains a BGT24MTR11 radar transceiver, equipped with a pair of arrays of microstrip patch antennas (a transmitting and a receiving one) characterized by a 12 dBi gain and 20° × 40° beamwidths (defining in this way the Field of View, FOV). The antennas used for the transmission and reception of the electromagnetic field (dedicated to the two channels) are integrated on the same board. In particular, such antennas are planar arrays composed of 4 × 2 elements (rectangular patch antennas). The board also includes a frequency control part for managing the timing and the number of transmitted chirps. In the receiving stage, there is also an analog amplification part dedicated to signal filtering and amplification, as well as a quadrature down-conversion mixer. Finally, it contains an XMC4200 micro-controller that controls the whole operation of the adopted radar board. In particular, it manages the data communication between board and PC and can be used for performing simple online signal processing tasks (provided that a proper firmware is adopted). In the developed system, the default firmware that enables the board to only capture the radar raw data without any other processing capabilities has been used. All the processing chains, including the FMCW radar processing, have been implemented on the control PC in a MATLAB environment. The communication between the PC and the radar board is performed via serial communication through the USB port. The debugger board can be used for programming the XMC microcontroller.

The second block of the system is a Raspberry PI 3B+ embedded PC, which was used as a link between the PC and the radar board. The Raspberry PI mini-PC is locally connected to the Distance2Go board via USB port while communicating with the control PC via WiFi. In this way, the Raspberry allows you to control the radar and collect its measurements remotely without the need for a wired network. The power supply of the Distance2Go board and the Raspberry Pi can be supplied via a power bank or by using a power supply with USB output. The Raspberry PI collects the data from the radar board and transmits it to the main PC via a Wi-Fi network.

The final part of the prototype consists of a PC running MATLAB, which controls the radar board via the Raspberry PI through the Raspberry PI toolbox. This toolbox helps create a virtual port for accessing the USB ports of the Raspberry PI. The PC collects the measured radar signals and performs the data processing and classification.

2.1.1 Radar Module Configuration Parameters:

The following radar module parameters have been set by the manufacturer and cannot be modified (unless there are modifications to the XMC firmware and/or circuit changes to the board):

$$T_{rd} = 200\ \mu\text{s} \tag{2.1}$$

$$T_{pll\_s} = 400\ \mu\text{s} \tag{2.2}$$

$$M = 6144\ \text{byte} \tag{2.3}$$

$$P_{out,max} = 10\ \text{dBm} \tag{2.4}$$

where T_{rd} is the down-chirp time following the transmitted up-chirp (an abrupt transition from the maximum frequency of the up-chirp to the minimum frequency of the subsequent up-chirp can cause various problems including the generation of spurious signals), T_{pll_s} is the recovery time (steady-state time) required by the PLL before generating the next up-chirp, M is the amount of memory available to the microcontroller ADCs to digitally store the radar echo, and $P_{out,max}$ is the maximum transmitted power. It is important to note that no samples are collected during the T_{rd} and T_{pll_s} periods.

Contrary to those just illustrated, the following parameters can potentially be modified with commands given to the board via the serial interface but, if set with values other than those reported and declared by the manufacturer as optimal, they can lead to a significant deviation of the digitized signal compared to what indicated by the theoretical models relating to the operation of FMCW radars:

$$T_c = 1500\ \mu\text{s} \tag{2.5}$$

$$B = 200\ \text{MHz} \tag{2.6}$$

where T_c is the duration of the up-chirp and B is the relative band. In particular, the frequency modulation covers the frequency range $[f_0, f_0 + B]$, with $f_0 = 24.025$ GHz. From the value of the band, we can calculate the theoretical resolution obtainable for the measurements of the radar-target distance (range):

$$\Delta R = \frac{c}{2B} = 75\ \text{cm} \tag{2.7}$$

where c is the speed of light in vacuum.

In addition, to have a maximum unambiguous detection distance of the targets equal to $R_{max} = 25$ m, an appropriate number of samples N_s equally spaced over time must be selected with which to digitize the radar echo based on the following relationship:

$$N_s \geq \frac{4BR_{max}}{c} = 67 \tag{2.8}$$

In order to optimize the execution of the FFTs that are necessary for the processing of the received signal, it was chosen to adopt a number of samples equal to a power of two, that is

$N_s = 64$, accepting a slight decrease in the maximum obtainable distance. In fact, it is not possible to adopt the next two power (128) as this would cause problems related to the limited amount of memory available on the XMC, which for example, includes the huge decrease in the number of chirps per frame (N_c) from 24 to 12, causing a degradation in the Doppler spectrum resolution. Furthermore, the adoption of $N_s = 128$ would imply a maximum limit on the size of the range (called the instrumental range) unnecessarily much greater than the energy range [86], i.e. greater than the maximum distance actually perceivable by the radar given the radiated power and that reflected by the target.

Note that the above values satisfy the following important condition:

$$T_c \geq \frac{2R_{max}}{c} \tag{2.9}$$

This imposes that the duration of the up-chirp is greater than the maximum round-trip time, i.e., the time it takes the electromagnetic wave to reach a target placed at the maximum distance and return to the antenna, in order to avoid ambiguity in the interpretation of the demodulated signal. It is also important to note the following points:

1. The maximum round-trip time is much less than the recovery time given by $T_{rd} + T_{pll\_s}$, that is

   $$\frac{2R_{max}}{c} = 166\text{ ns} \ll T_{rd} + T_{st} = 600\ \mu\text{s} \tag{2.10}$$

   This implies that the received chirp does not overlap with the next chirp to be transmitted and thus no spurious beat frequencies are generated in the receiving mixer.

2. The maximum round-trip time is much less than the sampling time, that is

   $$\frac{2R_{max}}{c} = 166\text{ ns} \ll T_s = \frac{T_c}{N_s} = 23.4\ \mu\text{s} \tag{2.11}$$

   Since the ADC timer starts counting when the transceiver starts generating the chirp to be transmitted and the first sample is collected after the time T_s, this implies that the first sample is collected when the demodulated signal is already at the ADC input. It follows that the effective echo observation window is equal to T_c and therefore the band used in the modulation is actually equal to B.

Due to the limited memory availability of the microcontroller, the number of chirps transmitted and collected in a single frame is equal to $N_c = 24$. In the first implementation of the system, due to some limitations in the interface of the Raspberry mini-PC with the control PC, this number was reduced to $N_{c,eff} = 21$. This value implies the following performances relating to the radial velocities directly observable by the Doppler component of the radar echo:

$$\left|v_{r,max}\right| = \frac{\lambda_0}{4T_{PRT}} \cong 5.3\ \frac{\mathrm{km}}{\mathrm{h}} \tag{2.12}$$

$$\Delta v_r = \frac{\lambda_0}{2T_{CPI,eff}} \cong 0.53\ \frac{\mathrm{km}}{\mathrm{h}} \tag{2.13}$$

where $\left|v_{r,max}\right|$ is the maximum absolute value of the detectable radial speed, $T_{PRT} = T_c + T_{rd} + T_{pll\_s}$ is the PRT (Pulse Repetition Time), λ_0 is the wavelength in vacuum relative to the frequency $f_c = f_0 + \frac{B}{2}$, Δv_r is the resolution of the radial speed measurement, $T_{CPI} = N_c T_{PRT}$ is the duration of the frame (also known as Coherent Processing Interval, CPI) and

Table 2.1 Radar Sensor Parameters

Symbol	Description	Value
B	Sweep Bandwidth	200 MHz
f_0	Center Frequency	24.025 GHz
f_s	Sampling Frequency	42667 Hz
R_{max}	Maximum Range	25 m
V_{max}	Maximum Velocity	5.4 km/hr
ΔR	Range Resolution	0.75 m
ΔV	Velocity Resolution	0.4 km/hr
N_s	Number of Samples/Chirp	64
N_c	Number of Chirps/Frame	21
T_c	Up-Chirp Time	1.5 ms
a	Chirp Slope	133 MHz/s
T_{PRT}	Pulse Repetition Time	2100 µs
T_{CPI}	Coherent Processing Interval	44.1 ms

$T_{CPI,eff} = N_{c,eff} T_{PRT}$ is the processed duration of the frame (due to the number of chirps available for processing). The bounded value of $|v_{r,max}|$ with respect to the radial speeds of the objects that can pass through the gate to be monitored, it is necessary to pay particular attention to the choice of the algorithms present in the processing chain for the classification of the targets. Finally, given that a frame lasts $T_{CPI} = N_c T_{PRT} = 0.05$ s and $T_f = 0.2$ s between the beginning of one frame and the next, the adopted card works with a duty cycle of about 25%. The complete set of the radar sensor parameters is shown in Table 2.1.

Figure 2.3 illustrates the time-frequency representation of the transmitted radar signals (chirps). It also further demonstrates the influence and significations of the different radar parameters on the time-frequency variations of the transmitted signal.

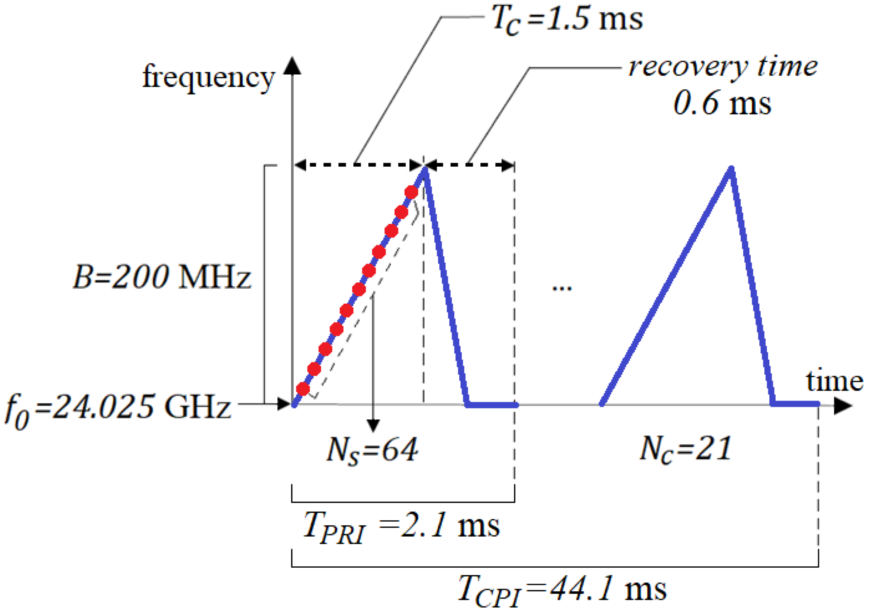


Figure 2.3 Time-frequency representation of the transmitted signal.

2.2 Software Architecture:

Figure 2.4 illustrates the flow chart describing the overall operating logic of the software developed for gate monitoring. The acquisition of frames from the radar card is achieved thanks to the ComLib libraries provided by Infineon. The beginning of an event to be recorded for any processing is given by the exceeding of a threshold th by the maximum peak of the IF signal belonging to the frame just acquired and calibrated (i.e., after the removal of clutter and leakage). Thus begins the formation of a collection of frames. Following the start of the event, should a frame be acquired whose maximum peak does not exceed the range-threshold (th), the recording is not immediately interrupted, as it could only be a large fluctuation in the reflectivity of the target again. in transit; recording is interrupted only if the threshold is not exceeded for a number of consecutive frames equal to N = 10 (V_c is the relative counter). Once the registration is finished, the number of frames that make up the dataset is checked (the final frames for which the V_c counter has been increased are not considered in the count); if this is less than 3, the collection is discarded, since it is probably a ghost target, otherwise, this is processed in order to obtain the identification of the transit targets.

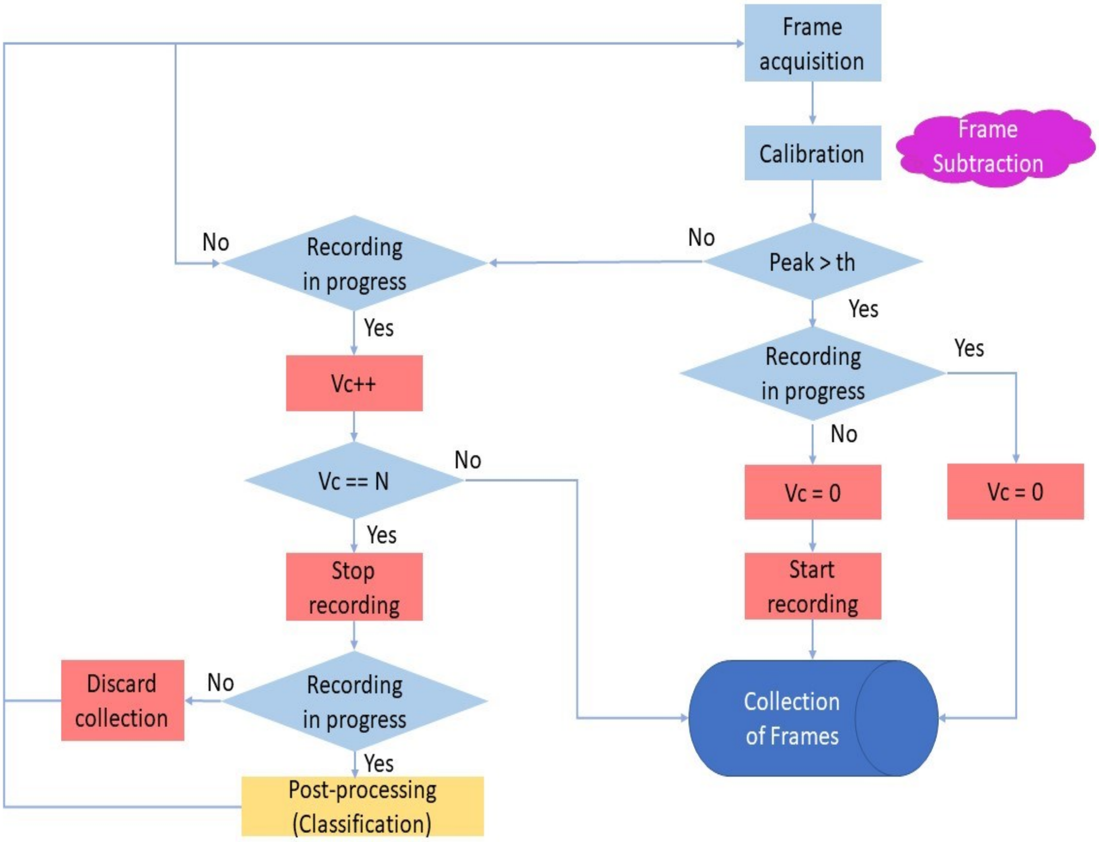


Figure 2.4 Gate monitoring flow chart.

Figure 2.5 illustrates the flow chart relating to the processing that the dataset relating to the recorded event undergoes. flowchart describing the main blocks of the developed data

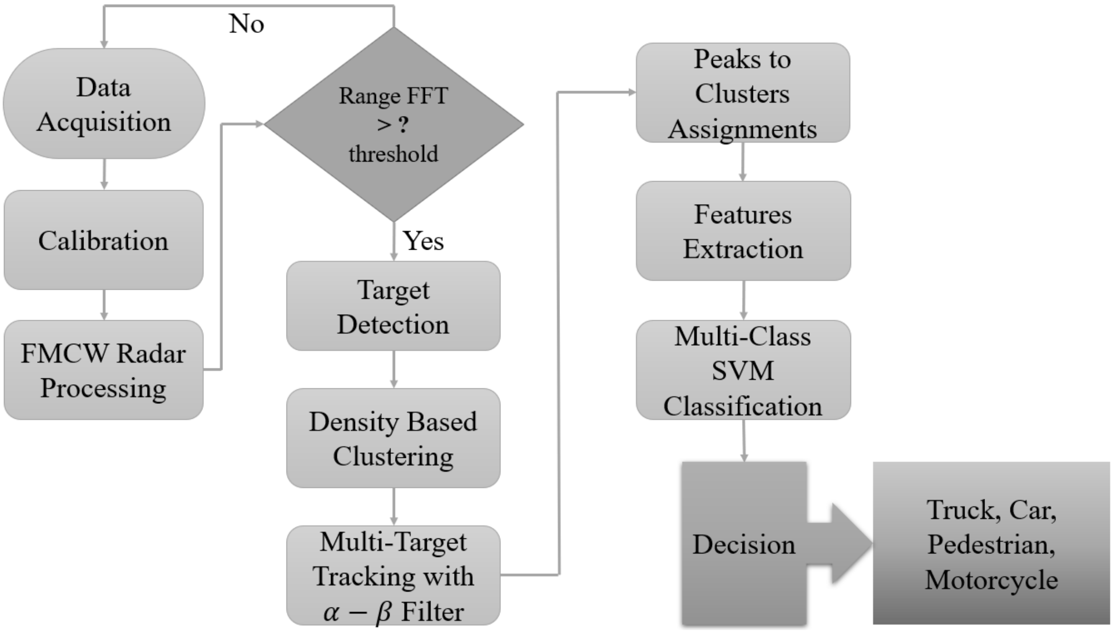

No
Data Acquisition
Range FFT > ? threshold
Peaks to Clusters Assignments
Calibration
Yes
Features Extraction
Target Detection
FMCW Radar Processing
Multi-Class SVM Classification
Density Based Clustering
Multi-Target Tracking with $\alpha - \beta$ Filter
Decision
Truck, Car, Pedestrian, Motorcycle

Figure 2.5 Flowchart of the developed data processing algorithm

processing algorithm. The developed approach has been specifically designed for extracting, from the measurements provided by a low-cost radar, a set of multi-target features to be used to classify different objects passing through the monitored gate. To this end, after the acquisition of the raw data, a background subtraction algorithm is used to calibrate the measurements. Then, the FMCW processing based on the 2D FFT analysis [41] starts whenever a signal with an amplitude higher than a proper range detection threshold is detected. The Range-Doppler matrix obtained from the 2D FFT Processing then undergoes clustering with the DBSCAN algorithm. After that, an ad-hoc designed peak to cluster assignment step is used to confirm the detections coming from real targets in the range spectrum, and later on to separate the contributions of each target in the spectrum in order to facilitate the subsequent extraction of the range-spectrum-based features. This step also decreases the false alarm rate as other detections coming from noise or clutters are discarded. The indices of the confirmed detections are then fed to a specifically designed multi-target tracking algorithm, based on the α-β filter, that assigns a unique target-ID to each target. The tracking filter provides the feature extraction block with the target IDs and indices, whereby each of the features is assigned to its respective target. The measured features are then fed to a multi-class SVM for classification. In the end, the detected target will be classified as either a truck, motorcycle, car, or pedestrian.

The output of the overall algorithm consists of a message indicating the type of targets passed through the gate during registration. The following chapters show the details of the individual blocks of the processing chain, including the calibration phase.

It is worth mentioning that all the thresholds mentioned in this chapter are only related to the range threshold. The other used thresholds will be discussed din the later chapters. It is also worth noting that it has been chosen to not adopt a Constant False Alarm Rate (CFAR) methods to automatically determine this threshold since these algorithms may degrade the range resolution [85], which is already quite low in the adopted low-cost board, and the adopted simple thresholding strategy already allows to obtain good classification performances.

Chapter 3: Pre-Processing of the Radar Data Frame

Prior to the tracking and the classification of the detected target as one of the aforementioned classes, the target's echo signal should undergo a necessary pre-processing phase. This phase starts with the calibration of the radar echo signal, which is necessary to remove any spurious collection of points, as well as cancelling the contributions of the environment from the spectrum while keeping only the contributions coming from the target. Next, the target's information related to distance and speed is obtained after the FMCW processing, which also produces the Range-Doppler map holding the signature of each target. Lastly, the DBSCAN clustering algorithm is applied to cluster the target's data points on the RD maps, which is a crucial step before the tracking. This chapter will illustrate and explain in detail the different pre-processing steps used in the processing chain.

3.1 Background Removal with Frame Subtraction:

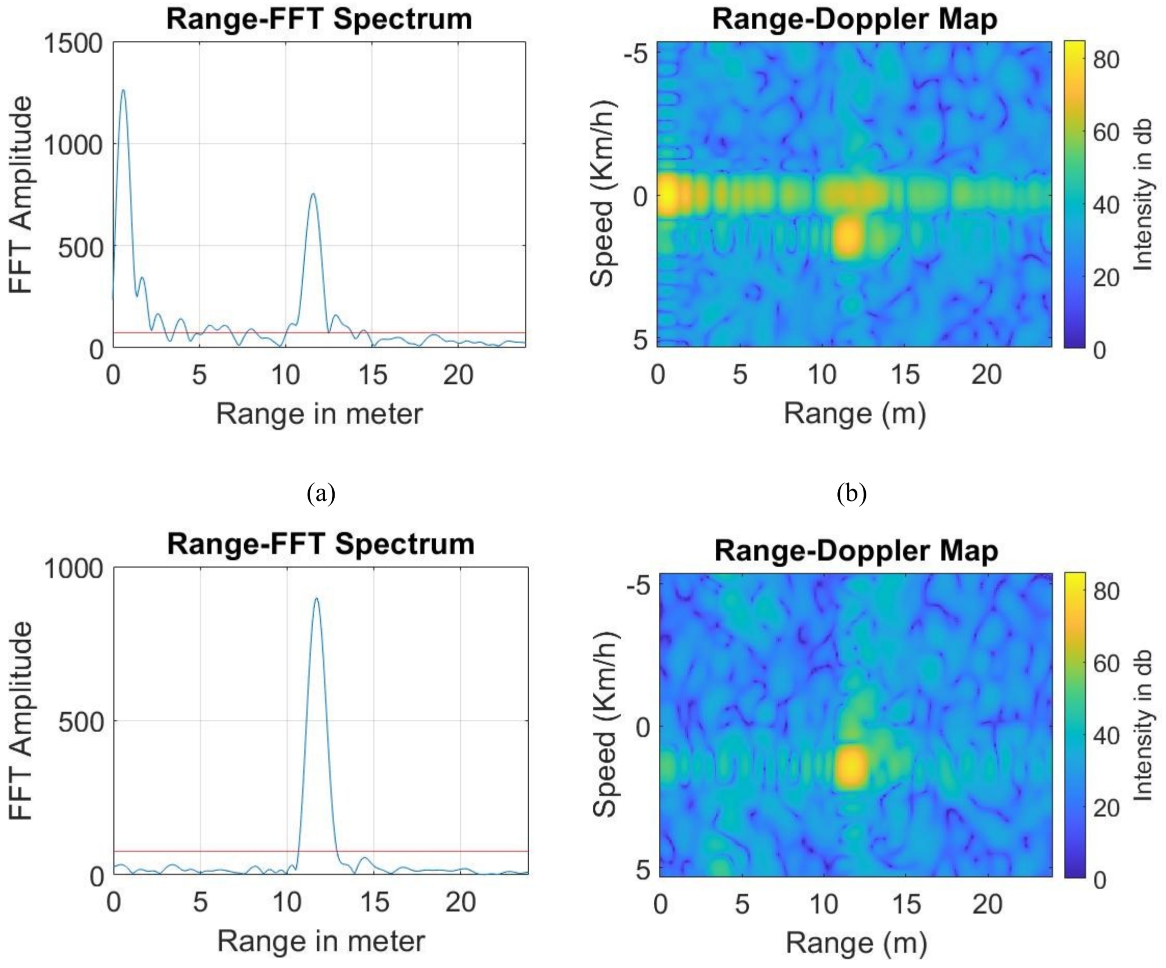
A Moving Target Indication (MTI) filtering technique [85] is firstly applied to cope with the RX-TX leakage [18] and the clutter caused by static objects [87]. In particular, the filter is a technique based on Background Subtraction (BS) which is a popular image processing scheme used for moving target recognition [81]. In the developed system, each radar data frame is represented as a $N_s \times N_c$ data matrix and can be then considered as a digital image. BS considers the foreground detection as the difference between a newly recorded image and an old image used as a reference (often called the "Background model" or "reference model") [81]. In particular, standard methods, which are often also provided by commercial radars, rely upon the use of a fixed reference background. For the application at hand, however, this results in a not optimal removal of the contribution of the background, leading to the appearance of clutters whenever a moving target appear in the FOV of the radar. Consequently, a modified BS algorithm has been developed for calibration and moving object detection.

The main difference with respect to standard BS is that the reference frame is not fixed. Indeed, since the electromagnetic scattering response of the scenario may vary during time (e.g., due to changes in the static objects present in the scene), a varying and adaptive background reference model is used. In particular, the previous frame is used as a reference model, i.e., the calibrated frame at time step k is given by:

$$Y(k) = F(k) - F(k-1) \tag{3.1}$$

where, $F(k)$ and $F(k-1)$ are the frames recorded at time steps k and $k-1$. The data frame considered here is the raw data matrix obtained before the 2D FFT processing, which is directly provided by the radar board and represents the input data to the processing chain. However, it is worth noting that BS could also be performed after the FFT computation, leading to comparable results. Besides, it is also worth highlighting that the subtraction is carried out on the complex-valued data provided by the I/Q receiver. In this way, the target response, which changes from a frame to another, can sum constructively or destructively, thus the resulting peak does not necessarily have a lower amplitude. However, the static response usually has a similar response, thus allowing the removal of the corresponding clutters. Consequently, the adopted frame subtraction technique allows to significantly decrease stationary clutters in the radar image while keeping the scattering response of moving targets. Indeed, although the target amplitude may be different, the overall information needed for the subsequent classification is retained. Nevertheless, if a target stops when passing through the monitored area, its response would be cancelled by BS. However, the tracking algorithm adopted in the processing chain still maintains a lock on the target, allowing it to recover its response when it restarts moving and to perform the classification. It is also worth noting that, with respect to other approaches relying on the removal of the zero-Doppler areas in the range-Doppler map, which are often used in radar applications, BS has the advantage of removing additional noise (e.g., due to random changes in the scenario or to other objects with non-strictly zero Doppler response that may be present).

Figure 3.1 exemplifies the application of the proposed clutter removal technique in the case of a single target located at 11.7 m from the radar and departing with a speed of 2 km/h. In particular, Figure 3.1(a) and Figure 3.1(c) show the magnitude of the range FFT (i.e., obtained through a Fourier transform of the chirp with the maximum FFT amplitude of the measured frame) before and after BS. It can be clearly seen that before applying the frame subtraction technique, the obtained range-FFT spectrum has, in addition to the main target peak at 11.7 m, several small other peaks related to clutters and one big peak close to zero caused by the leakage between the transmitting and receiving antennas. In the corresponding Range-Doppler maps in Figure 3.1(b) and Figure 3.1(d) several artifacts are present, and a very high peak clutter in the zero-speed axis is also present. The target is represented by a weaker peak at 11.7 m and 2 km/h. On the other hand, the range-FFT spectrum, after frame subtraction was applied, shows only one strong peak related to the target at 11.7 m. Moreover, a single sharp peak related only to the moving target, with no contribution from the clutters appearing on the zero-speed axis, is present in the Range-Doppler map.



(a) (b)

(c) (d)

Figure 3.1 Example of results of the range FFT and Range-Doppler map (a)-(b) before and (c)-(d) after calibration with the frame subtraction algorithm.

3.2 FMCW Radar Processing:

In the developed system, the board operates as a FMCW radar with a saw-tooth-like chirp signal and a linear frequency modulation. Specifically, a multi-chirp sequence FMCW approach is adopted, in which a frame composed by $N_c = 21$ subsequent chirps is transmitted and processed to extract the range and Doppler information of the targets. Figure 3.2 illustrates the architecture of the FMCW transceiver present on the Infineon Distance2Go board. Through a VCO (Voltage Controlled Oscillator), externally controlled by a PLL (Phase Locked Loop), the following signal is generated, called up-chirp [88], [89].

Frequency modulation is done to estimate the range of the target. The general form of the FM signal of linear frequency modulation (chirp) signal is:

$$S_{tx}(t) = A_{tx} \cos\left[2\pi\left(f_0 t + \frac{B}{2T_c} t^2\right) + \emptyset_0\right] \quad 0 \le t \le T_c \tag{3.2}$$

where A_{tx} is linked to the amplification provided by the power amplifier that feeds the transmitting antenna, f_0, B and T_c are respectively the initial frequency, the band, and the duration of the up-chirp, while $\emptyset_0$ is the initial phase. The blue trace in the time-frequency

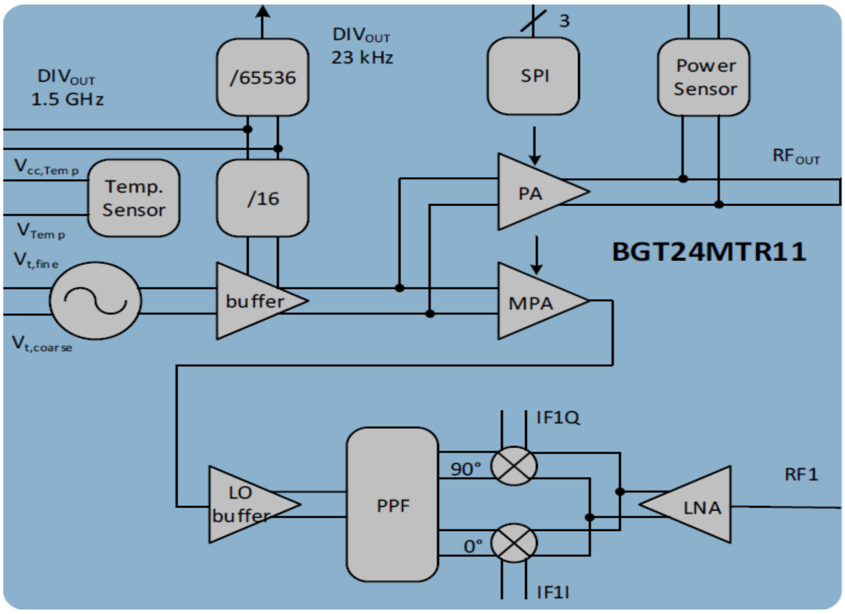


Figure 3.2 FMCW transceiver present on the Infineon Distance2Go board.

graph in Figure 3.3 exemplifies the signal in (3.2). The relative radiated electromagnetic wave travels to the target positioned at a distance $R(t) = R_0 + vt$ (with v radial velocity), is reflected by the latter and is finally picked up by a special receiving antenna after a total time (called round-trip) equal to

$$\tau = \frac{2R(t)}{c} \tag{3.3}$$

where c is the speed of light in vacuum. The signal received in this way (which is a copy of the transmitted one delayed by a time τ), amplified by an LNA (Low Noise Amplifier), is presented in the following form:

$$S_{rx}(t) = A_{rx}\cos\left\{2\pi\left[f_0(t-\tau) + \frac{B}{2T_c}(t-\tau)^2\right] + \emptyset_1\right\} \quad \tau \leq t \leq T_c + \tau \tag{3.4}$$

This signal is input to an I / Q demodulator (or I / Q mixer), which, using a copy of the transmitted signal, returns the following analytical signal at an intermediate frequency (Intermediate Frequency, IF):

$$IF(t) = S_{rx}(t)S_{tx}^*(t) = I(t) + jQ(t) = A_b e^{-j2\pi\left\{\emptyset_2 + \left[\frac{2v}{c_0}f_0 + \alpha\tau_0\left(1-\frac{2v}{c_0}\right)\right]t + \frac{2v}{c_0}\alpha t^2\left(1-\frac{v}{c_0}\right)\right\}} \tag{3.5}$$

$$\tau \leq t \leq T_c$$

where S_{tx}^* indicates the conjugate complex of the transmitted signal, $A_b \propto A_{tx}A_{rx}$ and $\emptyset_2$ is a constant phase term. If multiple targets are present, the demodulation result will be a sum of IF signals each corresponding to a single reflector. The coefficient that multiplies t in the second term of the exponential in (3.5) is the beat frequency, while the third term can be interpreted as a chirp on the beat frequency caused by the variation of the distance and can usually be neglected. As regards the beat frequency, this is composed of two addends: $\alpha\tau_0\left(1-\frac{2v}{c_0}\right)$, which can be approximated with $\alpha\tau_0$, represents the component of the beat frequency directly linked to the distance of the target, while the term $\frac{2v}{c_0}f_0$ is the well-known Doppler shift. The fact that both the distance to the target and its radial velocity contribute to the beat frequency is a phenomenon known as range-Doppler coupling. However, due to the parameters illustrated in section 2.1.1, the Doppler term within the beat frequency is negligible, i.e., when using a small chirp time T_c, the Doppler component is usually very small and is negligible with respect to the size of the range bins.

The instantaneous frequency is the derivative of the phase of the beat signal:

$$f_b = \frac{1}{2\pi}\frac{d\varphi(t)}{dt} = \frac{B}{T_c}\tau \tag{3.6}$$

Based on (3.3) and (3.6) we can therefore conclude that by transmitting a single chirp and measuring the beat frequency f_b of the relative demodulated echo, it is possible to obtain the distance from the target as follows:

$$R = \frac{cT_c f_b}{2B} \tag{3.7}$$

This is known as the FMCW range equation.

In the radar field, it is often important to have information about the speeds of the targets as well as their distance from the radar. In this case, the common approach is to measure the rate of change of the phase of the beat signal between a pair of adjacent chirps; this rate of variation is directly linked to the Doppler shift and therefore to the radial velocity of the target. We, therefore, consider the transmission of N_c chirp, whose overall duration is called CPI (Coherent Processing Interval), and whose echo constitutes the block of measurements, called frame, necessary to obtain the information of distance and radial speed of the targets present in the scene. illuminated. To illustrate the process that leads to the determination of the desired information, we report below, except for multiplicative coefficients, the sampled IF signal, and in which we assume the presence of a single target (in the case of multiple targets the underlying procedure is extended by adding the contributions of individual targets) [90]:

$$IF(n_s, n_c) = e^{j2\pi(f_b T_S n_S + f_D n_c T_{PRI})} \quad n_s = 0, \dots, N_s - 1 \quad n_c = 0, \dots, N_c - 1 \tag{3.8}$$

where n_s is the index of the samples acquired in the single chirp, n_c is the index of the chirp within the frame, T_s is the sampling interval, $T_{PRI} = T_c + T_{rd} + T_{st}$ is the PRI (Pulse Repetition Interval) and $f_D = \frac{2v}{c_0} f_0$ is the Doppler frequency. The second addendum in the exponential of the previous signal arises from the variation of $\emptyset_2$ due to the radial displacement of the target that occurs between a pair of adjacent chirps. In fact, this displacement can be quantified in $\Delta X = vT_{PRI}$ which corresponds to a phase variation equal to $\frac{4\pi\Delta X}{\lambda_0} = 2\pi f_D T_{PRI}$. We apply the FFT to (3.8) along the n_s dimension (this operation is called range-FFT). From (3.9) we know that in the resulting spectrum, having support $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right]$ with $f_s = \frac{1}{T_s}$, a peak occurs at the frequency f_b; we denote by $\hat{r}$ the bin in which this frequency falls. Except for multiplicative coefficients, the transform just carried out evaluated in $\hat{r}$ has the following form:

We now perform an FFT along the n_c dimension (Doppler-FFT). The result of this operation

$$FFT[IF](\hat{r}, n_c) = e^{j2\pi f_D n_c T_{PRI}} \tag{3.9}$$

is a spectrum with support $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right] \times \left[-\frac{f_{Ds}}{2}, \frac{f_{Ds}}{2}\right]$, with $f_{Ds} = \frac{1}{T_{PRI}}$ sampling frequency along the Doppler axis, having a peak spectral in the bin $(\hat{r}, \hat{d})$, with $\hat{d}$ bin along the Doppler dimension in which the frequency f_D falls. This spectrum, indicated by the term range-Doppler map, therefore collects all the information regarding the distances and radial speeds of the illuminated targets. To sum up, The IF signal is sampled by considering, for each chirp, $N_s = 64$ samples. Consequently, a data matrix of dimensions $N_c \times N_s$ is formed from the output of the I/Q receiver. The board provides such a data matrix as a data frame, which can be processed and analysed to obtain the target's information. In particular, a 2D FFT processing [41] is applied to the collected data frames to simultaneously obtain the range and speed of the target. Such an operation is performed on the baseband signals available at the output of the I/Q receiver. The first FFT is applied along each chirp to provide the range bins and is called range-FFT. The second FFT is applied along the chirps for a single range bin to provide the Doppler information, and it is called Doppler-FFT. The matrix obtained at the end of the 2D FFT is

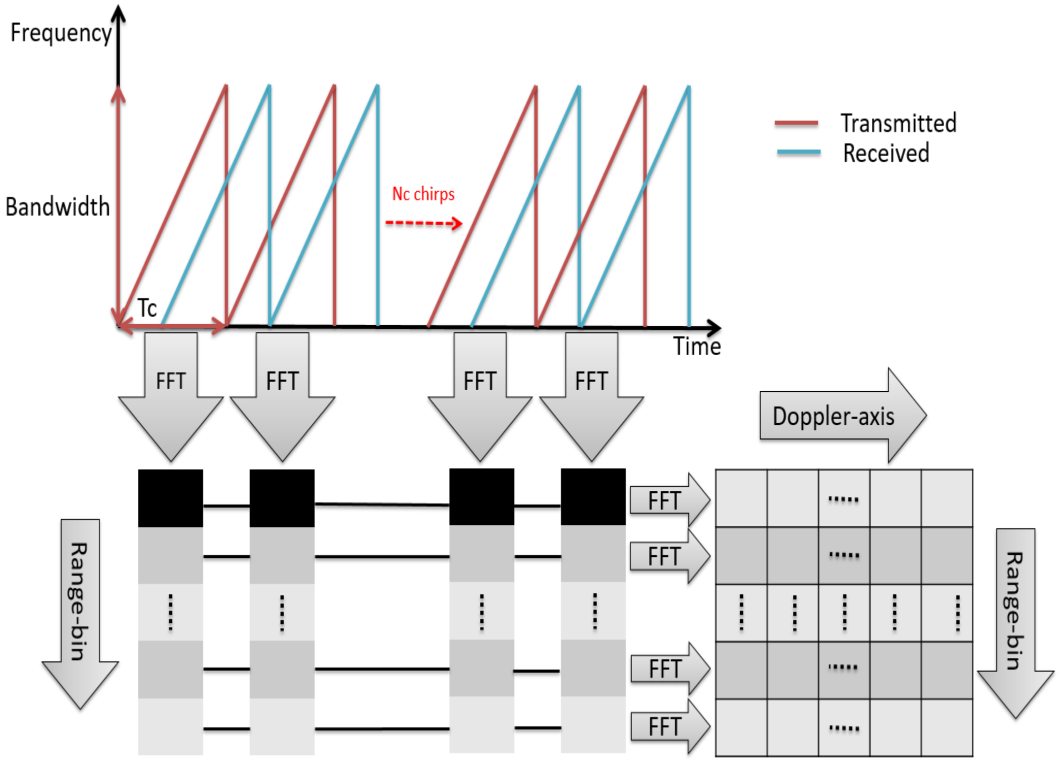


Figure 3.3 Basic concept of the 2D FFT processing in multi-chirp sequence FMCW.

called the Range-Doppler map and holds the signature derived from the echo signal of each target. The approach for the 2D FFT processing is illustrated in Figure 3.3.

In the present work the range-Doppler maps are calculated after applying zero-padding along the range and Doppler dimensions in order to obtain for these axes a number of points respectively equal to $N_{RFFT} = 512$ and $N_{DFFT} = 256$, so as to improve visibility spectral peaks isolated on the maps. Furthermore, downstream of the calculation of the two-dimensional FFT, we consider the application of the Kaiser window with the form factor $K_{sf,R}$ and $K_{sf,D}$ respectively along the dimensions of the range and Doppler. This allows reducing the side-lobes of the peaks relative to the targets on the maps [74]. The Kaiser window was chosen since it provides a better trade off when compared to other known windows, with the ability to tune it's parameters.

We conclude with a note regarding the sign of the radial velocity indicated in the range-Doppler map: due to the particular I / Q mixer present in the Distance2Go card, the spectral peaks occur in correspondence with the Doppler shift of opposite sign with respect to the approach/departure motion of the target with respect to the radar and it is, therefore, necessary to correct this behaviour in the construction of the maps.

It is worth noting that the 2D FFT processing is a well-known technique in the FMCW community and consequently several commercial radars provide integrated support to the 2D FFT processing, which may be transparent to the user. However, in order to have full control of the processing chain, in the present implementation of the system the whole processing chain is fully performed on the control PC starting from the radar raw ADC data frame provided by the radar board.

3.3 Range-Doppler Map Clustering with DBSCAN:

Considering the resolution of the adopted radar board along the range and Doppler axes (given in Table 2.1), as well as the dimensions of the targets and the dynamics of their movements, the targets will appear as extended clouds of points on the Range-Doppler map. In addition to that, targets with extended range profiles like cars will produce multiple peaks at different indices in the range-FFT spectrum. Consequently, an appropriate clustering algorithm is needed, in order to assign the indices of the points on the cluster maps to their respective peaks, which are related to each target on the spectrum, and thus consistently calculate the features for classification. To this end, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [82] is adopted. It is worth emphasizing that only the points at which the intensity of the map is greater than or equal to a proper threshold value th_{DBSCAN} selected by the user are considered in the DBSCAN method to remove spurious components in the Range-Doppler map that may be not fully removed by the clutter and leakage removal method. Moreover, the DBSCAN algorithm requires the definition of two more parameters: the minimum number M of neighbour required to identify a core point and the neighbourhood search radius ε.

There are two particular characteristics that made DBSCAN the best choice for the envisioned application: First, there is no need for any prior knowledge about the number of clusters as DBSCAN can automatically determine the number of obtained clusters. Second, DBSCAN clustering has the potential to recognize any spurious aggregation of points that are isolated and consequently should not be included in the clusters; they are assigned a class value of -1.

This feature, together with the adoption of a tracking scheme, is also useful to remove ghost targets generated by multipath reflections and thus reducing the false alarm rate. Indeed, the appearance of multipath reflections on the range-Doppler maps is hardly exploitable for classification, because of their strong dependence on a-priori unknown and barely predictable environmental factors, and they can be expected to act more as a flickering disturbance to be removed.

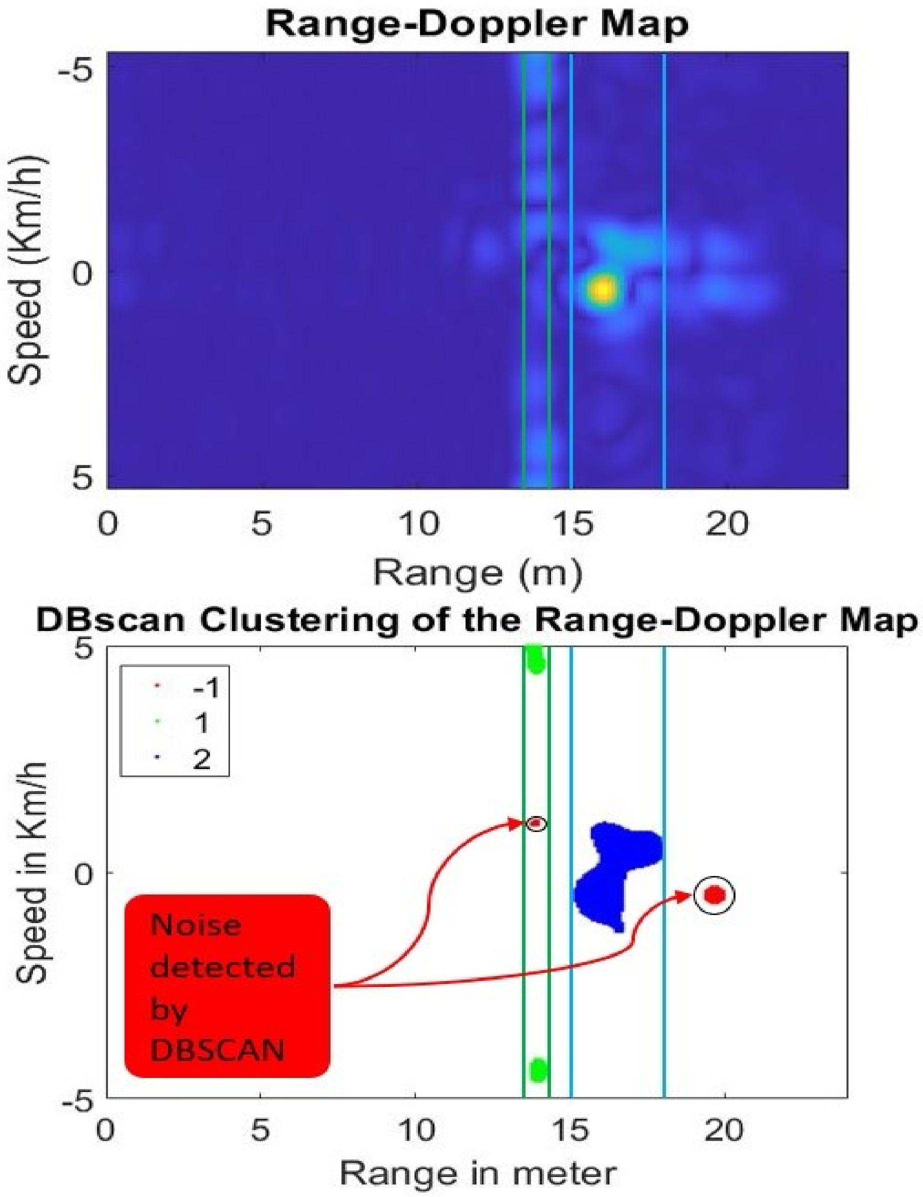


Figure 3.4 Result of the DBSCAN clustering on the Range-Doppler map, showing two clusters for two moving targets and a cluster of some spurious aggregation of points that is assigned a value of -1.

The steps that make up this algorithm are shown below:

1. Select the first untagged point in the dataset as the current search point. Initialize the C = 1 cluster counter.
2. Find the points located within the distance ϵ from the current point.
    1. If the number of points in that neighbourhood set is less than M, then mark the current point as noise. Go to step 4.
    2. Otherwise, mark the current point as belonging to cluster C.
3. Iteratively select each point of the proximity set as a new current point and repeat step 2 until further points are detected in the related proximity sets that can be labelled as belonging to cluster C.
4. Select the next unlabelled point in the dataset as the new current point and increment the cluster counter C = C + 1.
5. Repeat steps 2-4 until you have labels for all points in the dataset.

DBSCAN provides the true range points values along with their indices for each target, which is very helpful especially in the multi-target situation. These values are very crucial for the next steps related to multi-tracking and features extraction. Figure 3.4 presents the result obtained after applying DBSCAN clustering on the Range-Doppler map. Three different clusters were detected. The two clusters labelled (1&2) indicate the presence of two different targets as they have different colors. The red cluster which is labelled (-1) indicates the presence of certain noise in the measurements and it is not assigned to any target.

3.4 Peaks to Clusters Assignments:

This is a crucial and important step before the tracking and features extraction blocks, and it is at this stage where the DBSCAN algorithm shows its benefits and contribution to the processing chain. This step is used to confirm the detections coming from real targets in the range spectrum, and later on to separate the contributions of each target in the spectrum in order to facilitate the subsequent extraction of the range-spectrum-based features (especially in the multi-targe scenario). In other words, this step assigns the reflections of the targets on the range spectrum to their respective clusters on the Range-Doppler map.

This step can be formulated as follows. Let C_i be the i^{th} cluster obtained by DBSCAN. A target that presents multiple reflection points generates multiple peaks in the spectrum located at different ranges along the range axis. All the peaks p_j^i with ranges R_j^i belonging to the i^{th} cluster C_i are associated to a set of peaks $P_i = \{p_1^i, p_2^i, \dots, p_j^i\}$. The peaks that are not associated with any cluster are neglected, which highly decreases the probability of false alarm. The tracking algorithm benefits hugely from the "peaks to clusters assignments" step by introducing

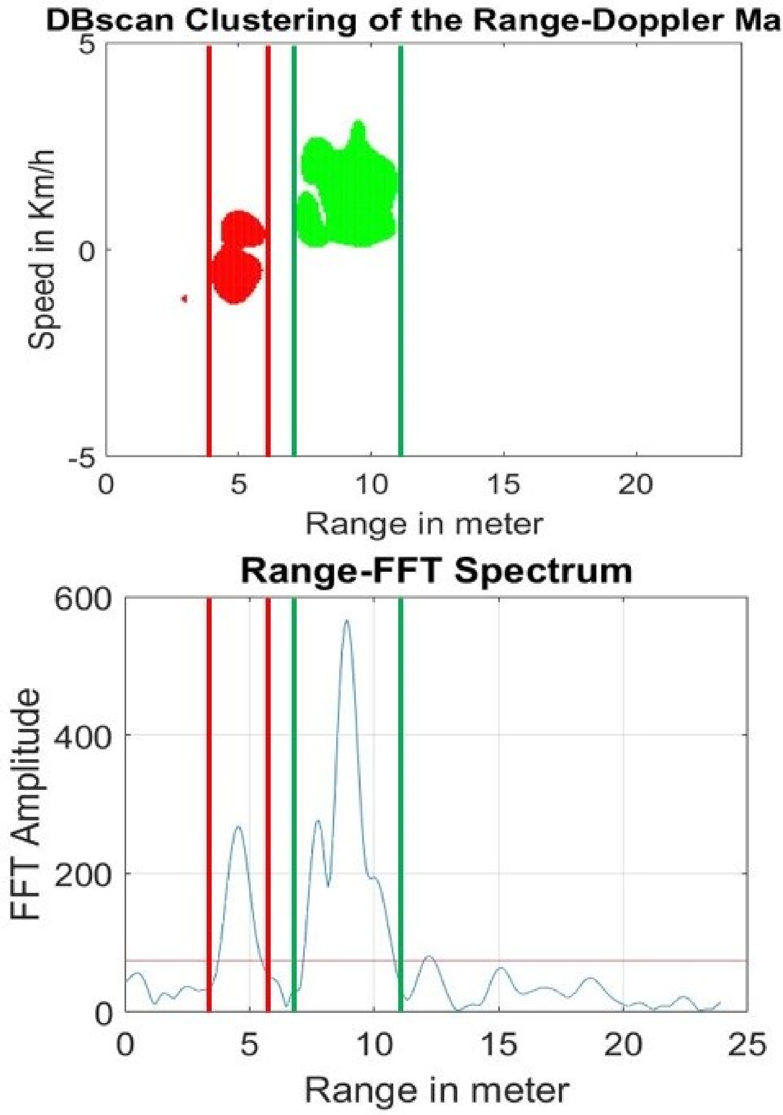


Figure 3.5 Car (green) and pedestrian (red) clustering with DBSCAN along with their respective contributions in the range-FFT spectrum.

only one tracking file for targets having multiple peaks. The tracking of such targets is based on tracking the range of the maximum peak in the spectrum. In the multi-target scenario, the tracking algorithm uses the information collected from the “peaks to clusters assignments” step to recognize the different targets on the spectrum and thus creating different tracking files for them.

Figure 3.5 exemplifies the results of using DBSCAN to cluster targets (pedestrian and car) on the Range-Doppler map, and it also shows the scattering from each target on the range spectrum. Comparing the images, the green cluster can be associated to the three peaks located in the same positions of its points, while the red cluster can be associated with the first peak. The importance of the “peaks to clusters assignments” step to the tracking and feature extraction steps will be further illustrated in the next chapters.

3.5 Summary:

To sum up, the collected radar data are arranged in frames that need to undergo a pre-processing step before being used by the tracking, and classification blocks. The first pre-processing step includes the calibration of the collected radar data that is used to cope with the RX-TX leakage, and remove the clutter caused by static objects and environmental reflections. A modified frame subtraction technique (based on the BS algorithm) was adopted in this case. The second pre-processing step includes the FMCW radar processing technique to obtain the target’s range and speed information. Specifically, a multi-chirp sequence FMCW approach is adopted, in which a frame composed by $N_c = 21$ subsequent chirps is transmitted and processed to extract the range and Doppler information of the targets. After, the DBSCAN algorithm was used to cluster the targets points on the resulting RD maps, which is used to separate the contribution of the targets on the spectrum following the “peaks to clusters assignments” step. The separation of the target’s contribution on the spectrum is of huge importance for the tracking and feature extraction blocks which will be further illustrated in the next chapters.

Chapter 4: Target Tracking with alpha-beta Filter

4.1 Introduction:

In order to perform multi-target classification, it is necessary to define a proper set of features for each target at every time frame. In particular, such features are extracted from the range and Doppler spectra and are related to the range and Doppler profiles, range and Doppler spreads, and the reflectivity of the targets, as detailed in Chapter 6. Then, the features of each target across several frames are used to form the final feature vector that is used for classification. Consequently, there is the need of performing multi-target tracking to correctly follow the targets throughout the various frames.

An ad-hoc tracking algorithm is proposed in this chapter, which is based on the solution of a one-dimensional tracking problem for each different target present in the FOV of the radar, using only the radar range measurements as inputs. At first, it creates separate track files for each of the radar detections, which are initialized with the first radar range measurements and zero speed. Subsequently, detection and prediction are performed, followed by an association phase, in order to use the radar measurements to predict the position of the targets on the next frame. Such predictions are compared with the new radar measurements to determine the track files the new measurements should be assigned to. After that, it uses a credit system to determine and control the lifetime of the created track files of each target. It also provides an estimate of the target’s speed, which is not directly possible with the radar hardware since its ability to unambiguously detect speed is limited to 5.4 km/h.

In particular, the α-β filter has been adopted in this study [84]. This chapter describes the implementation of the α-β filter for multiple target tracking. The parameters and the testing of the constant speed α-β filter are discussed. The data association and the credits system implemented along with the predictor are also discussed.

4.2 Tracking Model:

The tracking algorithm predicts the target's range and speed supposing a constant speed model. Then the predicted values are saved in a target tracking file. The tracking information is then provided to the feature extraction block for extracting the features of each target alone and later on classifying the targets to one of the aforementioned classes.

The α-β tracking algorithm uses the predicted range values from previous detects along with the current measured range to predict the range and speed of each target on the next frame. The α-β proposed here adopts Newton's motion model with a constant velocity which has the following equation of motion for range prediction:

$$R(k) = vk + R_0 \tag{4.1}$$

Where $R(k)$ is the instantaneous range predicted at time k, v is the constant velocity, k is the time, and R_0 is the reference range at the previous time frame $k - 1$.

4.3 Alpha-Beta Filter Implementation:

Figure 4.1 illustrates the considered α-β filter structure. In the figure, $y(k)$ is the current range measurement, while α and β are filter parameters, and T_f is the frame sampling time.

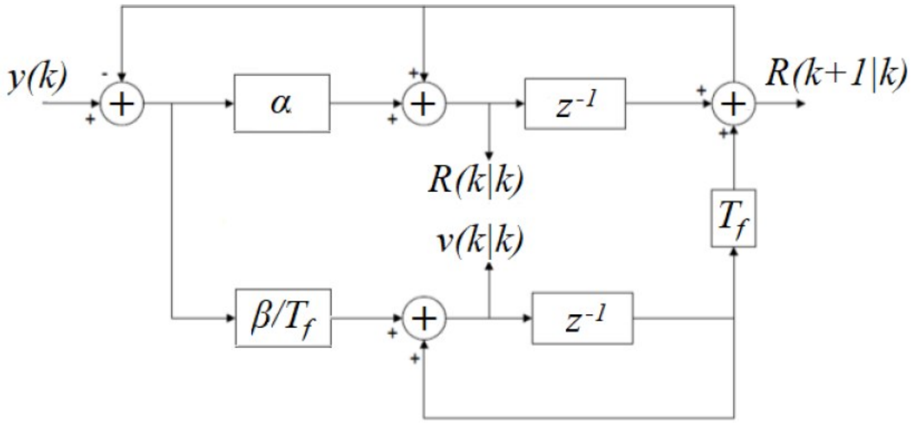


Figure 4.1 α-β tracking filter implementation.

The α-β tracking system can be modelled by the following state model [91]:

$$A = \begin{bmatrix} 1-\alpha_n & (1-\alpha_n)T_f \\ \frac{-\beta_n}{T} & 1-\beta_n \end{bmatrix} \tag{4.2}$$

$$C = \begin{bmatrix} 1-\alpha_n \\ \frac{\beta_n}{T_f} \end{bmatrix} \tag{4.3}$$

$$G = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.4}$$

$$\Phi = \begin{bmatrix} 1 & T_f \\ 0 & 1 \end{bmatrix} \tag{4.5}$$

The state equation is:

$$x(k|k) = A \times x(k|k-1) + C \times y_n \tag{4.6}$$

Where,

$$A = (1 - CG)\Phi \tag{4.7}$$

The z transform of (4.6) is

$$x(z) = Az^{-1}x(z) + Cy_n x(z) \tag{4.8}$$

Combining the above equations yields the transfer function of the system [91]:

$$\begin{bmatrix} h_x(z) \\ h_{x'}(z) \end{bmatrix} = \frac{1}{(z^2 - z(2-\alpha_n-\beta_n) + (1-\alpha_n))\begin{bmatrix} \frac{\alpha_n \times z(z-(\alpha_n-\beta_n))}{\alpha_n} \\ \frac{\alpha_n \times z(z-1)}{T_f} \end{bmatrix}} \tag{4.9}$$

Real-time multi-target tracking starts with initiation followed by measurement association and predication. Detections at the first frame initiate one or more tracks. On the next frames, the new detections will be either associated with the existing tracks or used to create new ones. At first, detections at the current frame are associated with the targets. However, for the un-associated detections, new tracks will be created. After that, predictions will be made for all the existing tracks.

4.4 Multi-Target Tracking:

In our system, in order to make multi-target classification feasible, we need to calculate the proposed features for each target at each range bin at every frame. After that, the measured features for each target among several frames need to be assigned together and averaged to form the feature vector to be used later for classification. This means that multi-target classification requires a multi-target tracking system.

In our radar sensor, the sampling time (T_f) is measured in frames that are collected every $200\ ms$. The target classification phase is done in conjunction with the multiple target tracking process.

The overall algorithm for multi-target tracking is illustrated in Figure 4.2, and it is summarized in detail in the following sections:

4.4.1 Track Initialization:

In the first frame, a track is initialized for each target identified by DBSCAN, whose initial state is given by the distance measured by the radar and zero speed. Furthermore, for each of these sets, a common credit value of value 6 is assigned (illustrated below).

4.4.2 Detection:

When a new frame is acquired, it is analysed to detect the targets that are present in the scene.

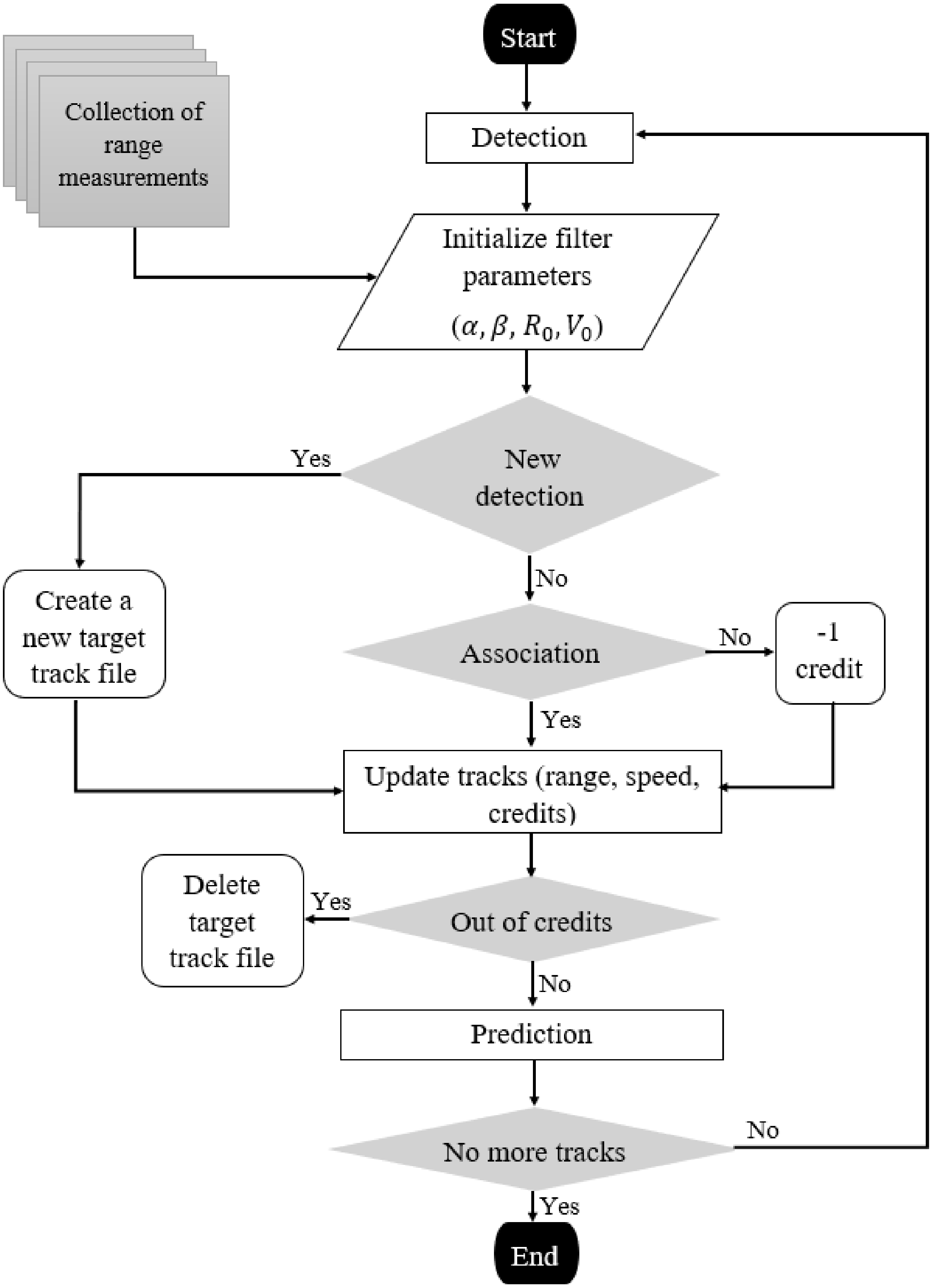

Start
Collection of range measurements
Detection
Initialize filter parameters
$(\alpha, \beta, R_0, V_0)$
Yes
New detection
No
Create a new target track file
Association
No
-1 credit
Yes
Update tracks (range, speed, credits)
Delete target track file
Yes
Out of credits
No
Prediction
No more tracks
No
Yes
End

Figure 4.2 Tracking scheme

4.4.3 Data Association:

The new detections are associated with existing tracks or are used to initialize new tracks. In particular, the association between an existing track and a new survey takes place by solving an assignment problem using the Hungarian algorithm [92], where the cost function is given by the difference between the distance corresponding to the survey itself and the distance predicted by the track in step 4.4.5. The detections not associated with any existing tracks are used to initialize new tracks, whose initial status is given by the measured distance and zero speed. A common credit value is also set for the new tracks. The cost is given as follows:

$$Cost_k^i = R_{k-1}^i - y_k^i \tag{4.10}$$

where, R_{k-1}^i is the i^{th} range prediction of the i^{th} target at the previous time frame, and y_k^i is the i^{th} range detection measured of the i^{th} target on the current time frame. The cost is calculated for each new track, and tracks associations are based on lowering the value of the corresponding cost matrix. In other words, each new detection is associated with the track where the value of the cost is minimal.

4.4.4 Credit System:

The credit system is used to check the activity of a certain target track file. The active tracks get new detections at every new frame, and so they get new predictions and associations. This permits the targets to have a lifetime measure based on their history. Every new target starts with an initially given credit that is constant for all targets. Whenever a target track misses an association with the new detections it loses a credit. On the other hand, whenever a target track gets an association, it gains credit. When the target track runs out of credits it is then considered a dead and inactive track and so it is deleted.

4.4.5 Alpha-Beta Filter Update and Prediction:

The traces associated with detections are updated through the correction operation indicated in (4.11)-(4.12).

$$R(k|k) = R(k|k-1) + \alpha[y(k) - R(k|k-1)] \tag{4.11}$$

$$v(k|k) = v(k|k-1) + \frac{\beta}{T_f}[y(k) - R(k|k-1)] \tag{4.12}$$

where $R(k|k)$ and $v(k|k)$ are the corrections, based on the current distance measurement, of the estimated distance and speed, respectively.

the operations in (4.13)-(4.14) are applied to predict the status of all the active tracks (associated or not) in the next frame. When no detection for the target track at the current time frame is recorded, only a prediction is performed using (4.13) and (4.14). In that case, the predicted velocity stays constant, and the predicted range is based on that predicted velocity. If no detections are associated with a target track for some number of frames (e.g., equals to the initially assigned credit value), it results in a linear prediction or 'coasting' target. Coasting is allowed for a short period of time after which the target track is deleted.

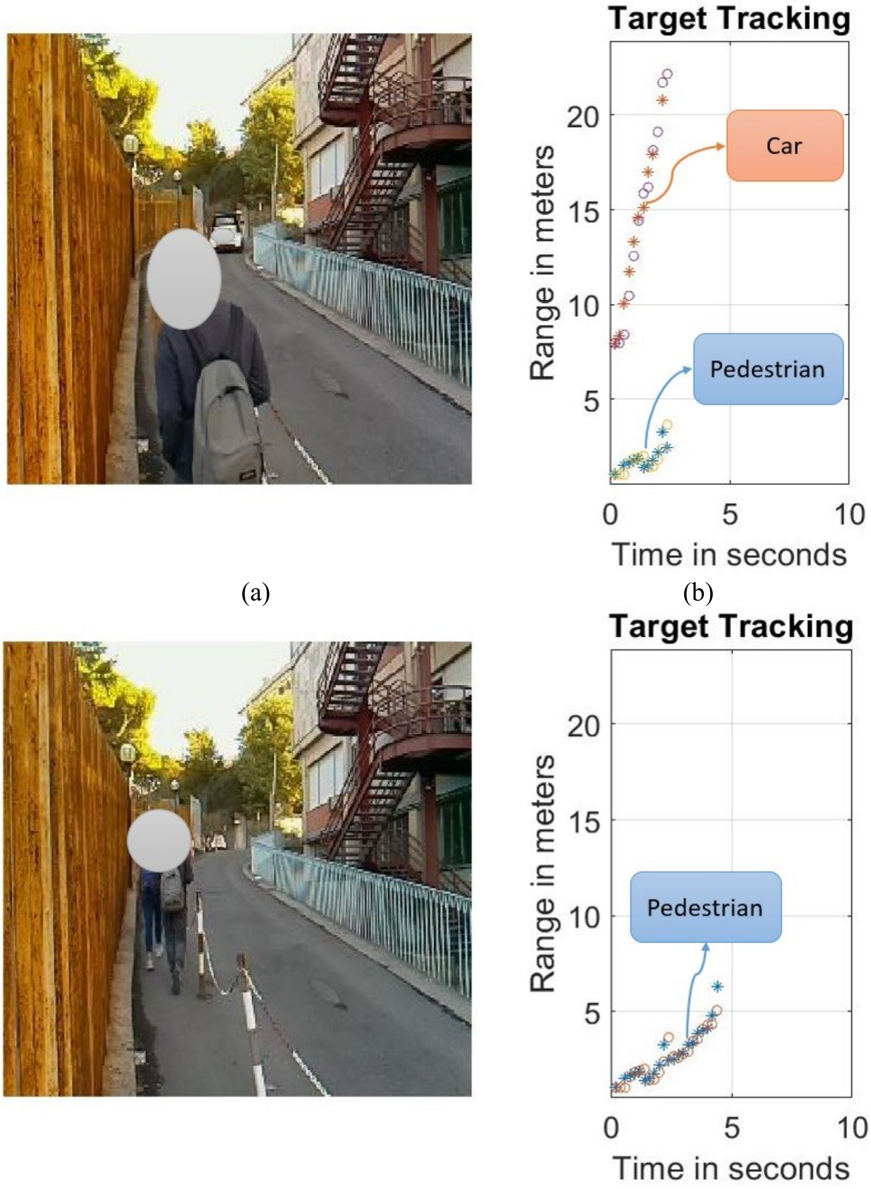
$$R(k+1|k) = R(k|k) + T_f v(k+1|k) \tag{4.13}$$

$$v(k+1|k) = v(k|k) \tag{4.14}$$

where $R(k+1|k)$ and $v(k+1|k)$ are the predictions for the next frame, based on the current state of the filter $R(k|k)$, and $v(k|k)$, respectively.

4.4.6 Alpha-Beta Filter Weights:

The parameters α and β are constant values selected by the user. Such parameters, whose values must reside in the range (0,1) for stability reasons, significantly affect the dynamic characteristics of the filter. Some methods for defining their optimal values are available for specific problems, e.g., [84]. However, it is also possible to heuristically find suitable values by performing some empirical analysis of their meaning. Indeed, the choice of α and β should be based on the accuracy of the measurements returned by the radar module of the distance and velocity of the targets occupying the illuminated scene: if this accuracy is high, then high values can be adopted for the two parameters, while otherwise low values allow obtaining a greater smoothing effect on the state estimate. Choosing high values for α means that more priority is put on range measurements rather than previous range predictions, whereas high values for β result in higher priority on the velocity measurements other than the previously predicted velocity.



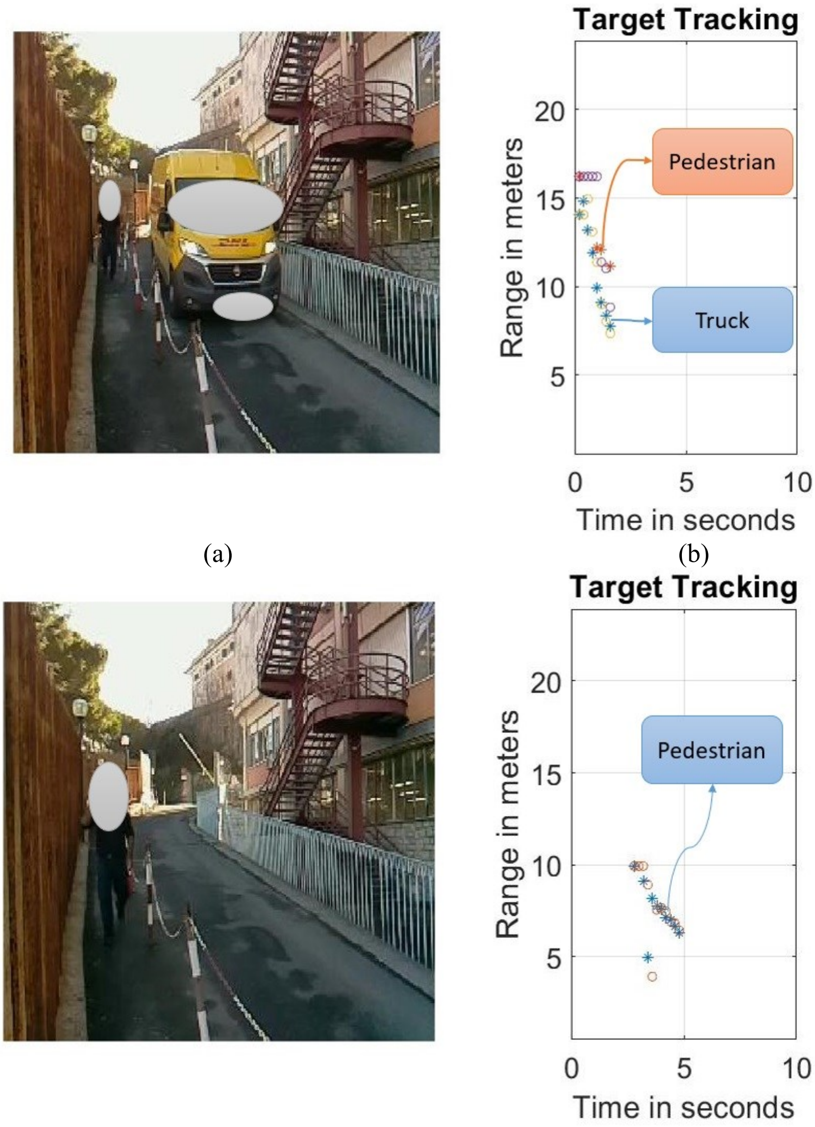
(a) (b)

(c) (d)

Figure 4.3 Camera images and tracking of test case 1. The circle symbol indicates the predicted path by the alpha-beta filter, while the star symbol indicates the measured path by the radar.

4.4.7 Tracking Results:

In the first test case of Figure 4.3, there are a pair of pedestrians moving away from the radar and a car also moving away (Figure 4.3(a)). Following the disappearance of the car from the scene, the recording ends with the presence of pedestrians only (Figure 4.3(c)). Since the two pedestrians are very close to each other (the separation distance is lower than the range resolution) and move with almost coincident trajectories, they appear on the Range-Doppler



(a) (b)

(c) (d)

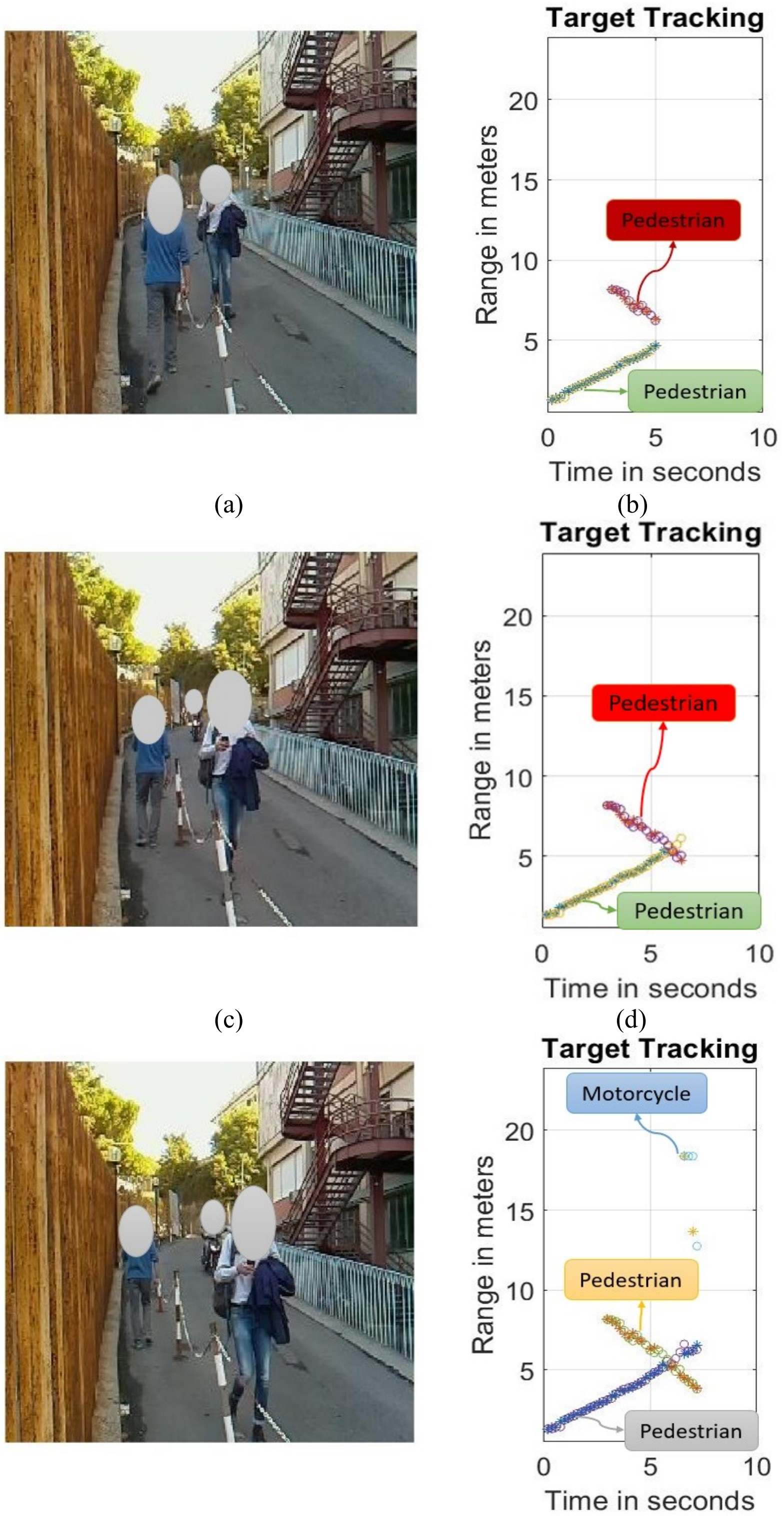
Figure 4.4 Camera images and tracking of test case 2. The circle symbol indicates the predicted path by the alpha-beta filter, while the star symbol indicates the measured path by the radar.

map as a single cluster and therefore the tracking is not able to distinguish the two underlying real targets, as we can see both in Figure 4.3(b) and in Figure 4.3(d). However, the method can follow their overall motion at a constant speed. In Figure 4.3(b) it is also clear the presence of the car, whose track is characterized by a greater slope than pedestrians because of its higher speed. We can also note in Figure 4.3(d) how the method correctly deactivates the track relating to the car when it leaves the radar range.

In the second considered test case, a pedestrian and a truck are present in the radar FOV as seen in Figure 4.4. In Figure 4.4(a)-(b), the tracking method has been able to follow both targets throughout their stay in the illuminated scene. It was also able to differentiate and keep track of the targets even after the overlapping of their trajectories. In addition, the track related to the truck is correctly deactivated when it leaves the radar range for several frames (depending on the number of credits it has), and only the pedestrian is tracked until the end of the event recording, as can be seen in Figure 4.4(c)-(d). It can also be seen from Figure 4.4(b), that the coasting target was detected for a short time after which it was deleted as coasting is allowed only for a short time, which proves the robustness of the tracking.

The third test case consists of two pedestrians, one of whom is approaching and one moving away, and subsequently an approaching motorcycle. Figure 4.5 illustrates some moments of the recorded event and the related tracking results (the different colors are used to denote different classes of targets in the image). In Figure 4.5(a) and Figure 4.5(c) there are the two pedestrians before and after the instant in which they found themselves at the same distance from the radar. From Figure 4.5(b) and Figure 4.5(d) it can be seen how the method is able to follow the trend at the constant speed of their trajectories. What is interesting is that the filter was able to keep the correct tracks of the pedestrians even after the occlusion as shown in Figure 4.5(d). In Figure 4.5(e) the motorcycle appears. However, due to its high speed, it is present in recording for a few frames, as can be seen from the scarcity of points linked to it in Figure 4.5(f).

In the last example case of Figure 4.6 (test case 4), an interesting and complex tracking scenario is presented. The scene started in Figure 4.6(a) with the presence of two pedestrians in the field of view of the radar. Similarly, to test case 1, since the two pedestrians are very

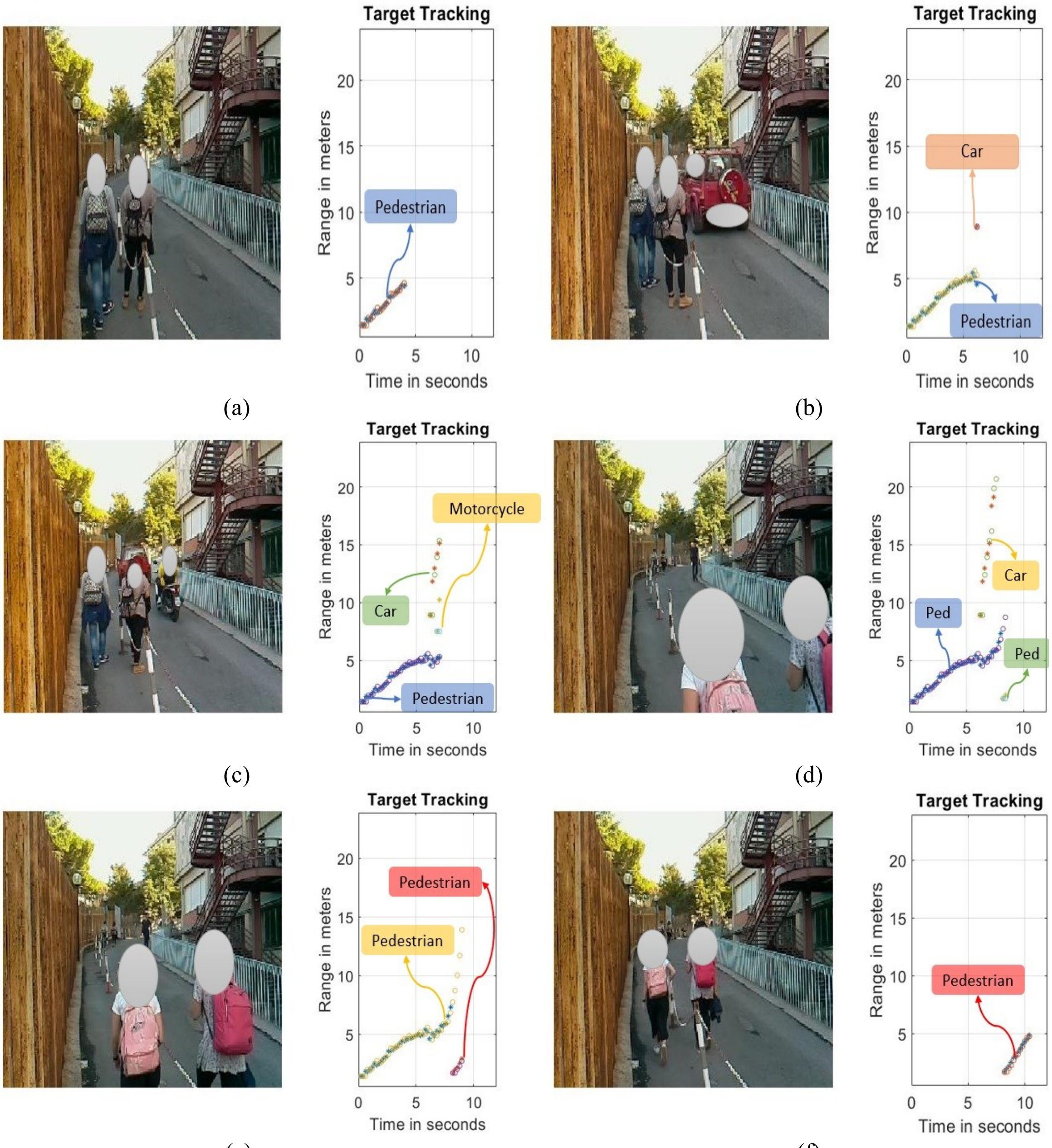


(a) (b)

(c) (d)

(e) (f)

Figure 4.5 Camera images and tracking of test case 3. The circle symbol indicates the predicted path by the alpha-beta filter, while the star symbol indicates the measured path by the radar.



(a) (b)

(c) (d)

(e) (f)

Figure 4.6 Camera images and tracking of test case 1. The circle symbol indicates the predicted path by the alpha-beta filter, while the star symbol indicates the measured path by the radar.

close to each other, they were recognized and tracked as a single target. In Figure 4.6(b) the movement of the pedestrians was disrupted by a passing car followed by a passing motorcycle. At this stage, the three targets present in the FOV of the radar were detected and tracked successfully, the tracking algorithm was able to successfully detect and track the three targets presented in the illuminated scene with three separate tracks as shown in Figure 4.6(c). This indeed validates the robustness of the tracking algorithm in detecting and tracking the multiple targets passing in the FOV of the radar. Following the disappearance of the motorcycle for several frames from the FOV of the radar in Figure 4.6(d), its track was deleted, and a new track was created for the newly detected pedestrians, who were detected and tracked as a single target. In Figure 4.6(e)-(f), the tracking algorithm again successfully deactivates the tracks referred to the car and pedestrian as they have left the scene for several frames while keeping an active track for the new pedestrians who are still moving in the FOV of the radar.

4.5 Summary:

In summary, the alpha-beta filter can track the maneuvering targets in both single and multi-target scenarios. The results of the tracking obtained in the multi-target cases are very good and showed that the tracking filter is capable of tracking different targets at the same time. The target track files generated by the filter allow for multi-target classification which is described in the coming chapters.

Chapter 5: Data Analysis and Features Extraction

This chapter presents the data analysis of the radar echo signals acquired from different types of targets. The signal analysis provides an idea about how the different targets have different spectrum behaviour and the reasons behind it. Understanding the difference in the spectrum behaviour of each target provides us with an idea about the different features that can be extracted from the spectrum obtained from each target. In addition, the type of the features along with the methods used to extract them will be further illustrated in detail.

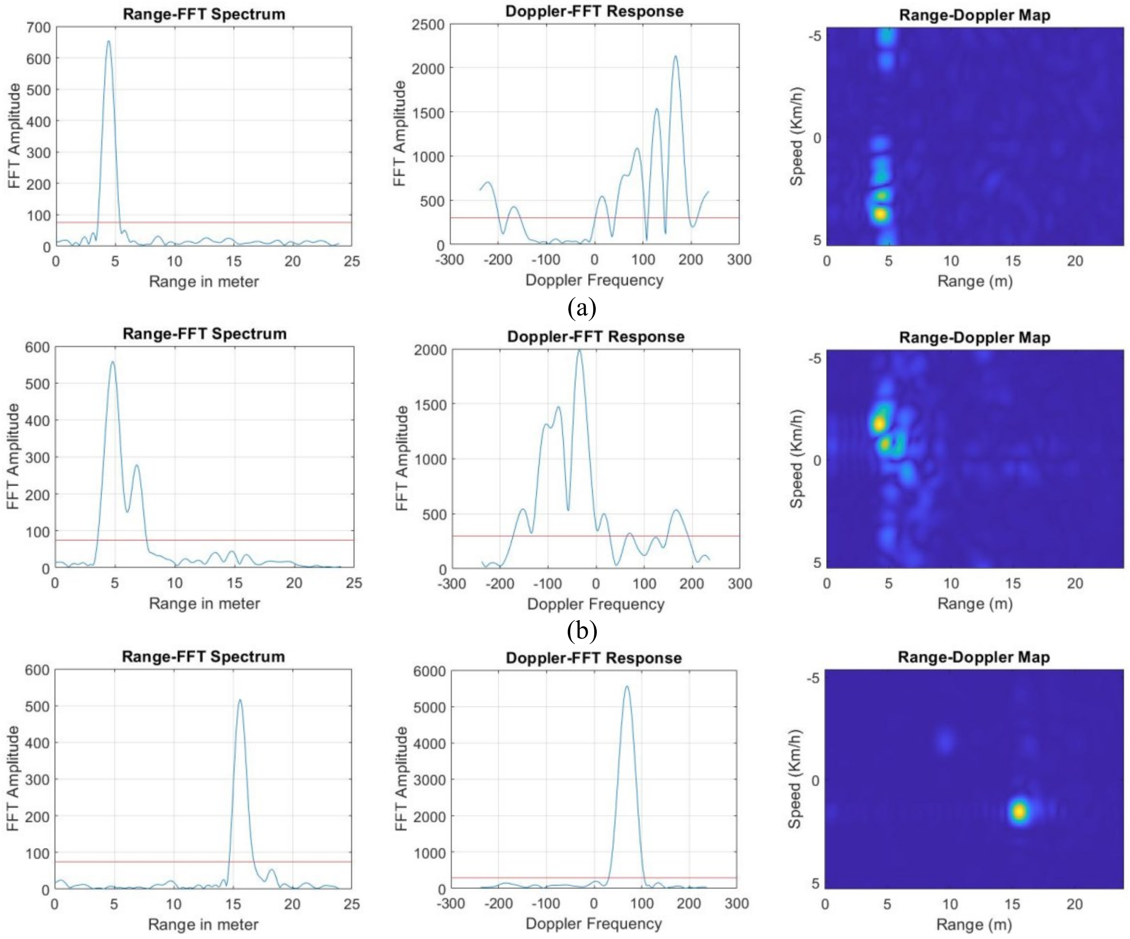
5.1 Data Analysis:

This section will provide some examples of the results obtained from the different targets. A spectrum analysis that explains the different spectrum behaviour of each target will also be provided, along with its effects on the shape of the Range-Doppler map. The examples will contain samples for the range-FFT spectrum, Doppler-FFT spectrum, and the Range-Doppler map.

5.1.1 Some Examples of the Recorded Data:

This part will show examples showing the difference between the echo signals reflected from pedestrians and vehicles. The examples will show the behaviour of the signal in three different domains, i.e., range-spectrum, Doppler-spectrum, and the Range-Doppler map.

Figure 5.1 shows the range, and Doppler spectrums along with the Range-Doppler maps of a pedestrian, a cross-range moving vehicle, and a along-range moving vehicle. In the case of pedestrians, Figure 5.1(a) shows a single peak in the range spectrum and a spread in the Doppler spectrum (resulting from the presence of multiple peaks), which is translated to a spread in the Range-Doppler maps along the Doppler axis. However, in the case of vehicles, two different behaviours of the recorded echo signals can be noticed based on the position of



(a)

(b)

(c)

Figure 5.1 Range and Doppler spectrums along with the Rang-Doppler maps of the reflected signals from: (a) pedestrian, (b) cross-range moving vehicle, (c) along-range moving vehicle

the vehicle with respect to the radar (cross-range or along-range), as shown in Figure 5.1(b), and Figure 5.1(c).

From the comparison between Figure 5.1(b) and Figure 5.1(c), it can be noticed that the cross-range moving vehicle shows some spread in range and Doppler spectrum which is translated to a spread on the axis of the corresponding heat map. On the other hand, an along-range moving vehicle shows no spread on the range and Doppler spectrum resulting in a single point-shaped target on the corresponding heat map.

Note that the range spread produced by the vehicle depends much on the size of the vehicle. For example, large vehicles like trucks are expected to have a big spread on the range spectrum, where small vehicles like small cars or motorcycles are expected to show a small spread on the range spectrum.

5.1.2 Spectrum and Heat Map Analysis:

- **Pedestrian echo signal**

Any non-rigid body motion is modelled as several jointly connected body segments (as in walking humans) [76]. The human body is considered a non-rigid and deformable-body; hence, the motion of the human body is a mixed motion of several moving parts in addition to the main body, i.e., legs and arms. This indicates that pedestrians are expected to have a spread in the Doppler spectrum since each part of the human body will have a different Doppler frequency. This explains the presence of several peaks in the Doppler spectrum of Figure 5.1(b) which is translated to a spread along the Doppler axis of the Range-Doppler map of Figure 5.1(c). However, the single peak in the range spectrum of Figure 5.1(a), is explained by the fact that the size of the human body along the range axis is less than the radar range resolution (Table 2.1), which indicates that pedestrians are always expected to be recognized as a single target and, in the general case, should always have one frequency in the range spectrum.

- **Vehicle echo signal**

Two different cases need to be considered here, i.e., cross-range, and along-range.

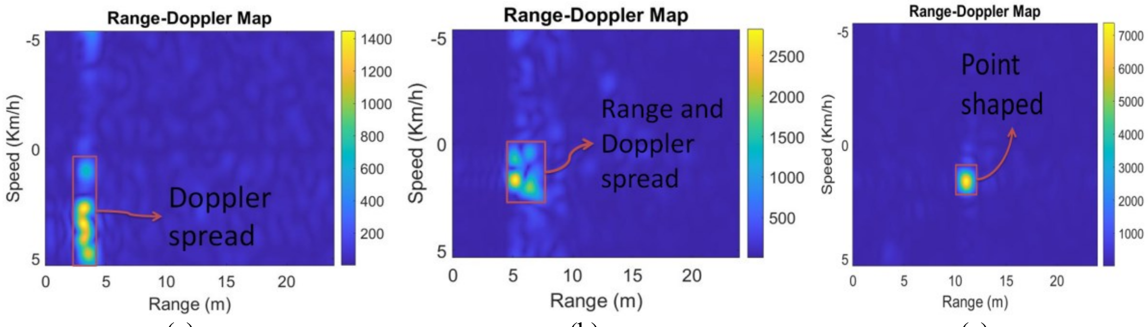
When the vehicle is very close to the radar, i.e., all the body of the vehicle is in the FOV of the radar (cross-range position) multiple peaks are present in the range-FFT spectrum and Doppler-FFT spectrum (Figure 5.1(a)-(b)), indicating a spread in the range and the Doppler spectrum simultaneously. This explains the result obtained in the Range-Doppler map of Figure 5.1(c), where the spread is clearly visible on the range and speed axis of the Range-Doppler map.

This is explained by the fact that the length of the vehicle is bigger than the range resolution of the radar and so the radar can detect more than one peak on the range-FFT spectrum depending on the length of the vehicle. Each of the detected peaks could have different radial velocities that can be bigger than the velocity resolution of the radar (Table 2.1), which results in multiple peaks in the Doppler-FFT. This resulted in a Range-Doppler map having a spread on both axes.

On the other hand, a moving vehicle in the far-field of the radar, where only the back or the front of the car appears (along-range position), shows only a single peak on the range-FFT spectrum. Consequently, one peak on the Doppler-FFT as well and a clean Range-Doppler map showing only a point shape target is obtained as in Figure 5.1(c). To explain this one should clearly understand the meaning of the range resolution of the radar as well as how the radar transmits the electromagnetic wave. So, as the radar range resolution requirement implies that the two targets should be separated by a minimum distance that is equal to the range resolution itself, one should know that this distance is only considered at different range bins or in other words along the range axis of the radar. Consequently, as the radar transmits the electromagnetic wave radially, then all the reflection points coming from the back of the vehicle are considered at the same range. Therefore, the result is a single big scattering point as seen in Figure 5.1(a).

5.2 Features Extraction:

To achieve a real-time classification system, it is needed to look for very simple features that can be extracted in an easy fashion without any complex computation that can take time and resources. In that regard, the feature vector is built using information that can be directly extracted from the range and Doppler profiles of the target. In particular, the range profile is



(a) (b) (c)

Figure 5.2 Range profile and velocity profile of a single measurement for a, (a) pedestrian, (b) along-range vehicle, and (c) cross-range vehicle

the measure of the spread caused by the target in the range spectrum. This spread mainly occurs due to a change in radial velocity, or whenever the length of the target along the range axis of the radar is higher than the radar range resolution. Similarly, the velocity profile is the extension that the target presents along the Doppler spectrum. This extension is caused by the radial velocities of the target's reflection points.

In the developed system, the method presented in [62], [93] is adopted, which allows a very simple and direct approach to calculate the range and Doppler profiles of the targets. In this section, we discuss the methods used for extracting such features in single and multi-target situations. The range and radial velocity of the target are measured simultaneously and unambiguously even in multi-target situations (Figure 5.2). The measurement results of a single observation are shown in the Range-Doppler map.

5.2.1 Extraction of Spectrum Based Features:

- **Range Profile:**

The range profile is the extension that the target has along the size of the range within the range-Doppler map and the range spectrum. This extension is caused by the spatial distribution of the reflection points that the target possesses and based on [62], [93], and [94], it is computed as follows:

$$R_{profile} = R_{max} - R_{min} \tag{5.1}$$

where R_{max} and R_{min} are respectively the maximum and minimum distances of the peaks (corresponding to reflection points of the target) in the spectrum of the range-FFT. It is worth remarking that only the peaks that are over a predefined threshold th_{range} are used in the computation of $R_{profile}$. Figure 5.3 exemplifies this calculation when a single car is present. Note that $R_{profile}$ is 0 in the presence of a spectrum containing a single peak, as for example usually happens for pedestrians.

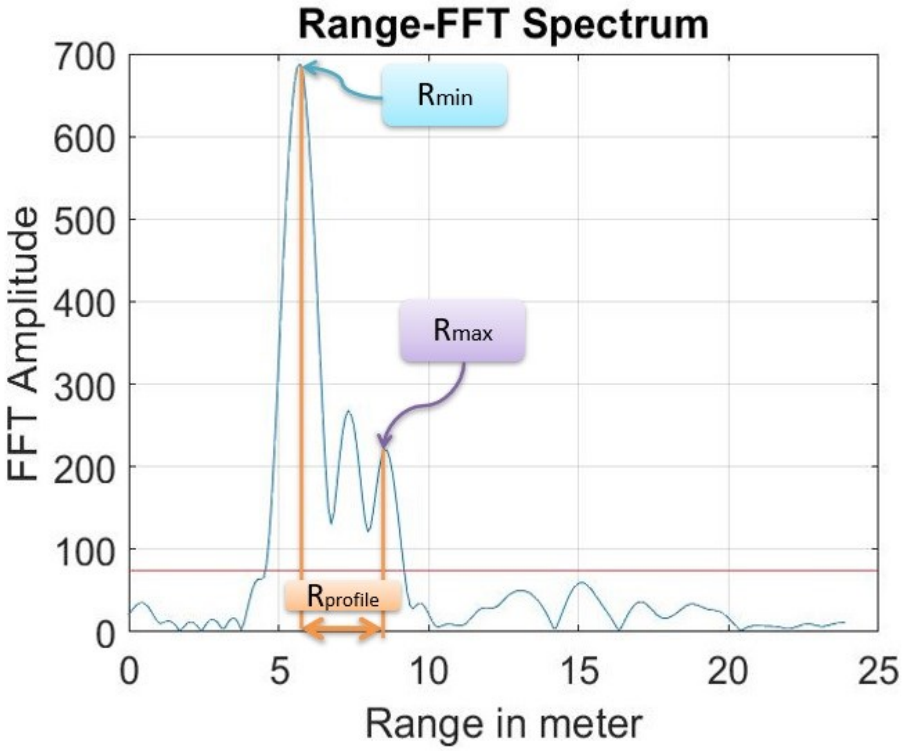


Figure 5.3 Example of computation of the range spread in the presence of a car. The horizontal red line indicates the threshold beyond which the peaks of the spectrum are considered.

- **Velocity Profile:**

The velocity profile is the extension that the target has along the dimension of the radial velocity within the Range-Doppler map and Doppler spectrum. This extension is caused by the radial velocities of the reflection points of the target. Since the maximum velocity measured by the radar is limited to $V_{max} = 5.4 \text{ km/h}$ (as indicated in Table 2.1), an aliasing phenomenon is present in the Doppler spectrum, especially in the case of vehicles; thus, the calculation of the velocity spread is not carried out directly on the output of the Doppler-FFT. First, it is verified in which Doppler half-axis (positive or negative) the greatest spectral content is present, after which only the 3/4 of the spectrum that contains the selected half-axis are considered. This process can be formulated as follows. Consider S_D the original Doppler spectrum obtained after the 2D FFT analysis, N_{pos} and N_{neg} are the numbers of positive and negative Doppler frequency samples contained in S_D respectively, and N_{DFFT} is the number of Doppler FFT points. The new Doppler spectrum S_D' after the deletion is:

$$S'_D = \begin{cases} S_D U(n + \frac{3 \times N_{DFFT}}{4}), & N_{pos} > N_{neg} \\ S_D U(\frac{3 \times N_{DFFT}}{4} - n), & N_{pos} < N_{neg} \end{cases} \tag{5.2}$$

where n is the discrete index on the Doppler frequency axis and $U(n)$ is the unit step function. Finally, the Doppler profile is computed by applying the following formula:

$$V_{profile} = V_{max} - V_{min} \tag{5.3}$$

where V_{max} and V_{min} are respectively the maximum and minimum radial velocities of the peaks, corresponding to the reflection points of the target, identified in the chosen spectrum portion. Figure 5.4 exemplifies this calculation when a single pedestrian is present in the scenario.

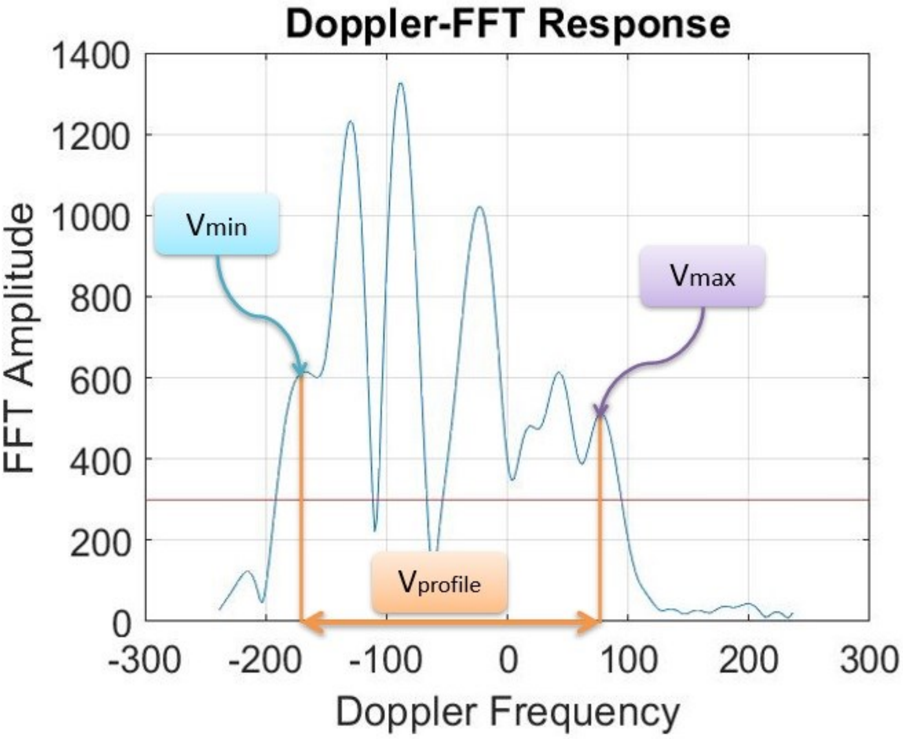


Figure 5.4 Example of computation of the velocity spread in the presence of a pedestrian. The red line indicates the threshold beyond which the peaks of the spectrum are considered.

5.2.2 Extraction of Spectrum Based Features in Multi-Target Scenario:

Using (5.1) and (5.3) to measure the range and Doppler profiles works fine for a single target situation. However, applying (5.1) directly in a multi-target scenario leads to wrong range profile estimations. This is because in the multi-target case it is always expected to have multiple peaks in the range spectrum that are coming from different targets. Hence, we should determine first the indices of the peaks corresponding to each target in the spectrum before applying (5.1).

In the developed procedure, this is achieved by using the results of the DBSCAN clustering algorithm [82]. However, before the calculation of the range profile, it is necessary to perform "peaks to clusters assignments". This step assigns the reflections of the targets on the range spectrum to their respective clusters on the Range-Doppler map. This step was discussed in detail in section 3.4.

Unlike the range profile, the calculation of the velocity profile in the multi-target case follows the same procedure as in the single target case. This is because the Doppler spectrum is obtained at each range bin. In other words, each target will generate its own Doppler spectrum which is usually distinct from the others. In fact, it may occur in some cases that many targets move in the same range bin. However, this will not be problematic as it is not the common case, and it doesn't usually last for more than one frame.

In order to create a more robust feature vector, the previous quantities, which are just related to the extent of the target response in the range and Doppler spectra, have been integrated with other global information. In particular, the mean and variance of the parts of the spectrum relevant to each tracked target have been included in the feature vector. Indeed, both range and Doppler spectra exhibit some random variations, due to target movements, residual clutter components that survived the removal method, and various sources of noise. For this reason, such contributions can be considered as random variables. Based on this consideration, the following preparatory quantities for the definition of the feature vector are also introduced (only the spectral components of amplitude greater than a threshold th_{range} are considered) [62], [95]:

- R_1, the variance of the target spectrum contained in the range-FFT;
- R_2, the standard deviation of the target spectrum contained in the range-FFT;
- R_3, an average of the target spectrum contained in the range-FFT;
- V_1, the variance of the target spectrum contained in the Doppler-FFT;
- V_2, the standard deviation of the target spectrum contained in the Doppler-FFT;
- V_3, an average of the target spectrum contained in the Doppler-FFT.

It is worth remarking that the standard deviations and variances used in the feature vector are closely related quantities. However, as often done in machine learning classification [52], [62], a feature vector containing redundant information has been adopted, since it may strengthen the classification.

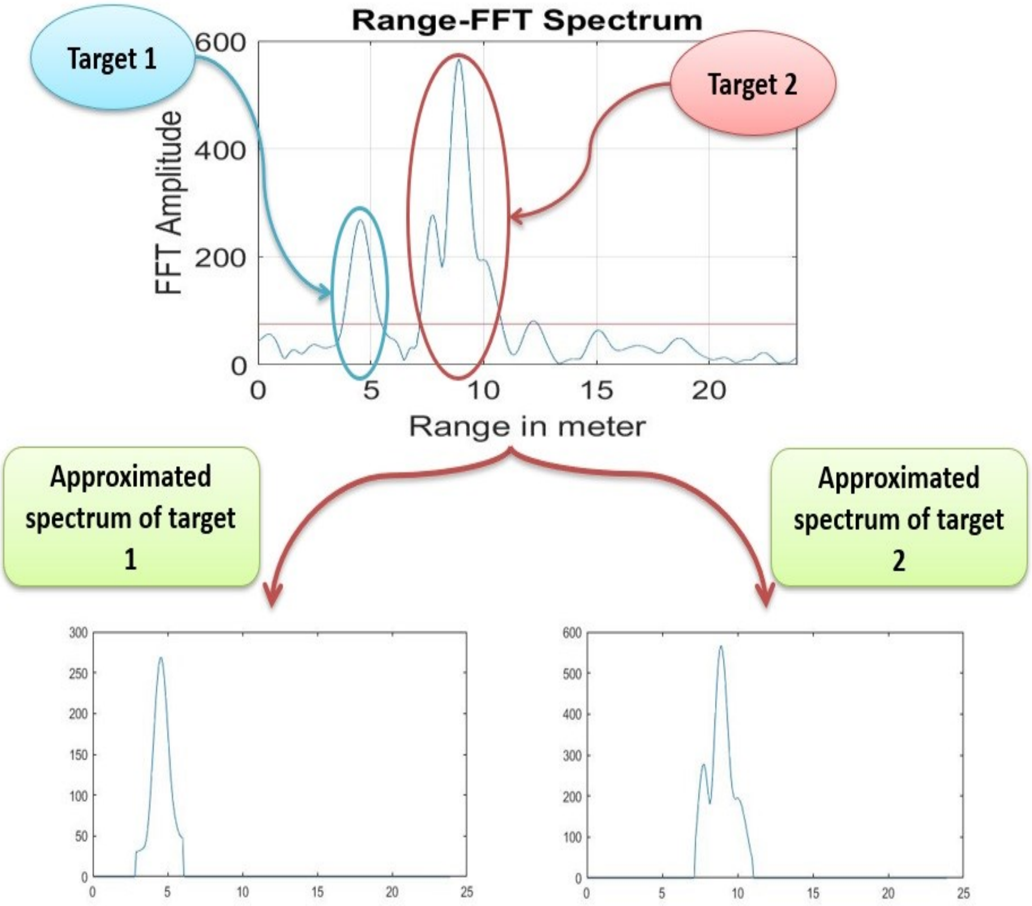


Fig. 5.5. Approximation of the range-FFT spectrum of each target separately from the contribution of the other targets in the range spectrum.

The measurements of V_1, V_2, and V_3 are simple, since, at each bin, a single target is present with a single Doppler spectrum, thus targets at different range bins will have their own separate Doppler spectrums. However, computing R_1, R_2, and R_3 in the multi-target case is more complicated because of the mixed contributions of multiple targets in the range spectrum. Consequently, to calculate the values of R_1, R_2, and R_3, an approximated range spectrum of each target is obtained depending on the range bins indices obtained from the DBSCAN clustering. Only the part that is above the range threshold was approximated for each target while the other parts are set to zero. This process is illustrated in Fig. 5.5.

To clarify things up, the process can be formulated as follows: Let j be the index of the points on the range axis, i be the cluster index, and S_{RFFT} is the range-FFT spectrum. Now consider $S_R^{(i)}$ as the i^{th} approximated range spectrum of the i^{th} detection based on the i^{th} cluster C_i, i.e.,

$$S_R^{(i)} = \begin{cases} S_{RFFT}, & \forall\, R_j \in C_i \\ 0, & otherwise \end{cases} \tag{5.4}$$

Finally, two more features are added to the previous ones, namely the measure of reflectivity and the average target radial speed estimated by the tracking algorithm. The reflectivity measure is proportional to the Radar Cross Section (RCS), which is a measure of the ability of the target to reflect the radar signal in the direction of the radar receiver. The reflectivity γ was measured following the procedure presented in [96]. The radial speed v_{avg} of a target is estimated by averaging the set of radial velocities in the target track file obtained using (4.12).

The components of the feature vector for classification are given by the means of the quantities listed above which are calculated on the set of available frames. The tracking algorithm provides the proposed feature extraction block with the correct indices of the targets along with their target IDs across the different frames. Based on this information, the extracted features at each new time frame are assigned to their true respective target. When a target track file is deleted by the tracking algorithm, the information about that target is no more provided to the feature extraction block, and so no more features are assigned to that target ID. However, the collected feature set of that target is saved to be averaged and classified later on, after which it is also deleted.

To sum up, the entire feature set for a single measurement is shown in Table 5.1. This feature set will be used to determine the model parameters of the classifier.

Note that, the feature vector provided later to the classifier contains the average of each feature over several time frames taken recorded for each target. Using the average of the features is important to reduce the range of error that might occur in the measurements of the features over a single time frame.

Table 5.1 Features Set

Feature	Description
$R_{profile}$	Extension in range spectrum
R_1	Variance in range spectrum
R_2	STD in range spectrum
R_3	Mean in range spectrum
$V_{profile}$	Extension in Doppler spectrum
V_1	Variance in Doppler spectrum
V_2	STD in Doppler spectrum
V_3	Mean in Doppler spectrum
γ	Reflectivity
V_{est}	Estimated target velocity

5.2.3 Results and Discussion:

- **Pedestrians Vs. Vehicles:**

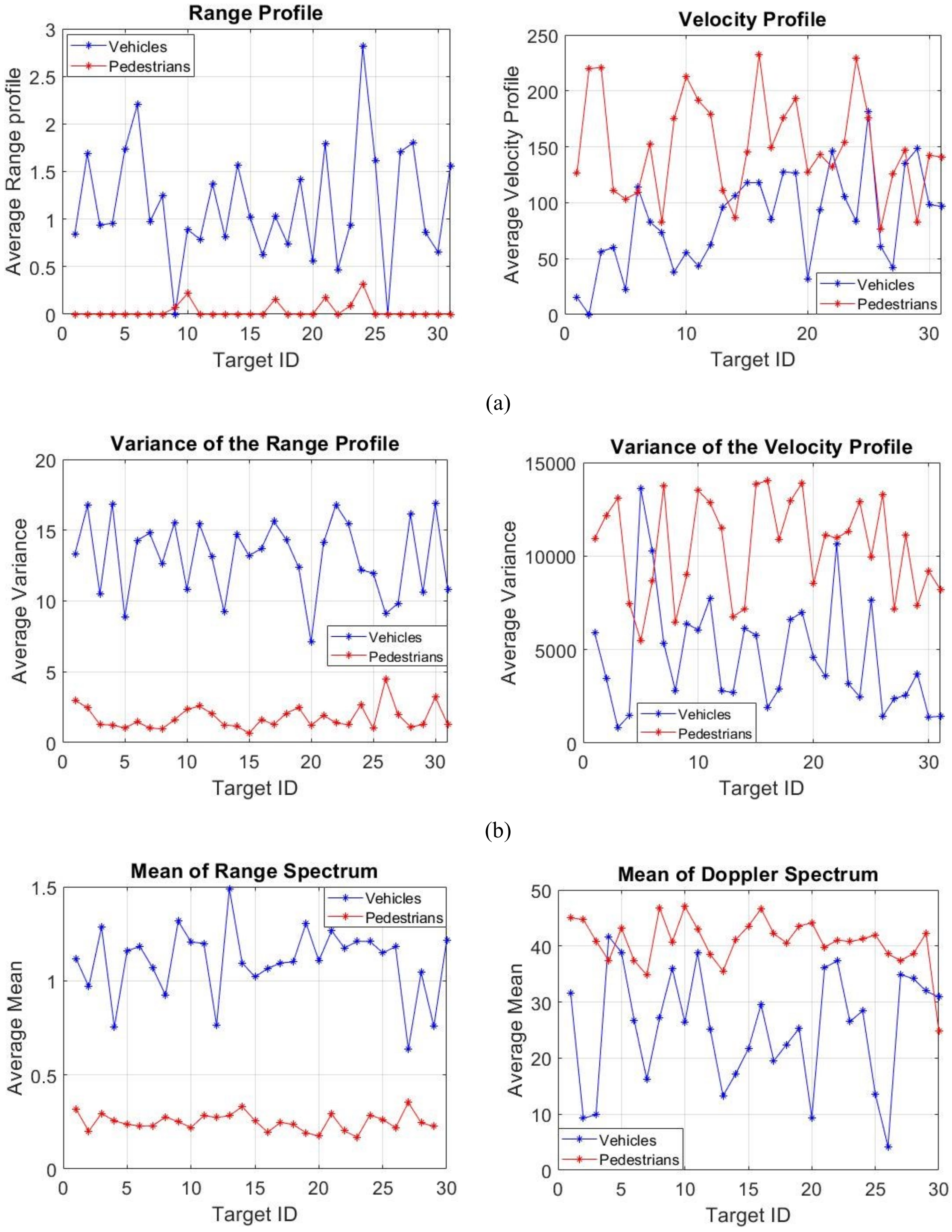
The measurement results of a single observation are shown in the Range-Doppler map of each target (Figure 5.2). Three different cases for targets have been recorded. A Doppler spread exists in the case of pedestrians and a Range spread in the case of an along-range moving vehicles. However, a cross-range moving vehicle showed a simultaneous range and Doppler spread. The features proposed in section 5.2.1 are measured according to a certain detection algorithm. Since, in some rare cases, the vehicle could have an echo signal like that of the pedestrian, especially for the case of a cross-range moving vehicle, the measurements of the features were calculated by taking the average of several measurements for each target instead of using only a single measurement. The features measurements were done for 31 samples taken from each class (31 humans and 31 vehicles). Figure 5.6 shows the obtained results of pedestrians vs. vehicles regarding the spectrum-based features, while Figure 5.7 shows the features related to the speed and RCS.

The difference in the average range profile of pedestrians and vehicles is clearly visible (Figure 5.6(a)), where most of the time the range profile of the vehicles is a lot higher, except in two cases in which the measurements were zero. This is mainly because the number of obtained measurements for those two cases was little. In fact, the range profile of the pedestrians should always be zero, however in some rare cases some clutters could intervene in the measurement of the range profile, but still, their effect can be hugely neglected.

On the other hand, Figure 5.6(a) also shows that the average velocity profile of pedestrians is greater than that of the vehicles in most of the cases. However, in many cases, it appears that the vehicle can have a similar velocity profile as that of pedestrians. This is mainly because the vehicle sometimes can have a Doppler spread like in the case of pedestrians, especially when the vehicle is moving in the cross-range position, but still, the average velocity profile of the pedestrian is notably higher.

Figure 5.6 (b), and (c), show that the Doppler distribution of pedestrians is totally different from that of vehicles as they have different variances and mean, and this is clear as the variance and STD of pedestrians is always higher than that of the vehicles. This is explained by the presence of multiple peaks above the detection threshold in the case of pedestrians due to the movement of the legs and arms in addition to the main body. This case doesn’t apply for vehicles, resulting in lower peaks compared to pedestrians and so lower variance and mean.

Finally, Figure 5.6 (b), and (c), also illustrate the difference in the range distribution of pedestrians and vehicles. In this case, vehicles always have a higher variance and mean, which



(a)

(b)

(c)

Figure 5.6 Extracted spectrum-based features of pedestrians vs. vehicles: (a) average range and Doppler profiles, (b) average variance variations of the range and Doppler spread, (c) average mean variation of the range and Doppler spread

is expected because of the expected range spread in the range-FFT of the vehicle. In the case of pedestrians, no range spread is expected since the size of the human main body is mainly smaller than the range resolution of the radar (Table 2.1), so the low values obtained in Figure 5.6 (b), and (c), for the variance and mean with respect to the values obtained for vehicles are in total agreement with what is expected.

In Figure 5.7 it can be clearly seen how the features for speed and RCS are much lower in the case of pedestrians with respect to vehicles. This is accompanied by lower variations in those features for pedestrians. It is worth remarking that the reflectivity may be affected by the state of the targets, especially for pedestrians. Specifically, the height of a pedestrian can significantly vary the RCS; in [97] an average reduction of 5 dBsm has been observed between a child and an adult in the range [23,28] GHz. However, in our settings, the class of pedestrians is the one already showing the lower RCS, and, consequently, it is expected that a child would be still classified as a pedestrian. On the contrary, the variability caused by different clothes is quite low: For an adult, it has been found to be less than ±1 dBsm, especially when the clothes are electrically thin and dry [97], [98]. Finally, concerning the posture, although the different poses during the walk may introduce fluctuations in the RCS [98], such behaviour is implicitly taken into account in building the feature vector.

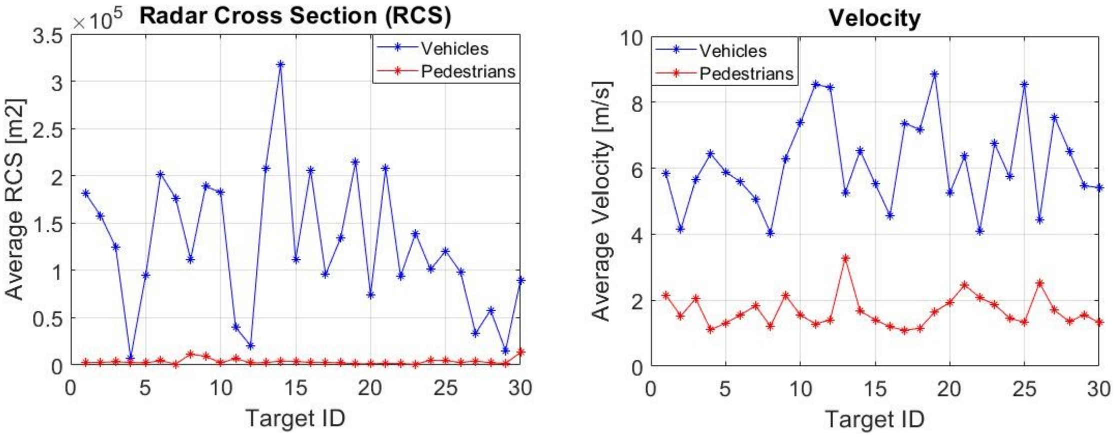


Figure 5.7 Extracted RCS and velocity features of pedestrians vs. vehicles

- **Vehicle Vs. Vehicle:**

Figure 5.8 presents the most definitive features that were used for the differentiation between the different targets of the vehicle class. As can be seen, these features are mostly range profile-

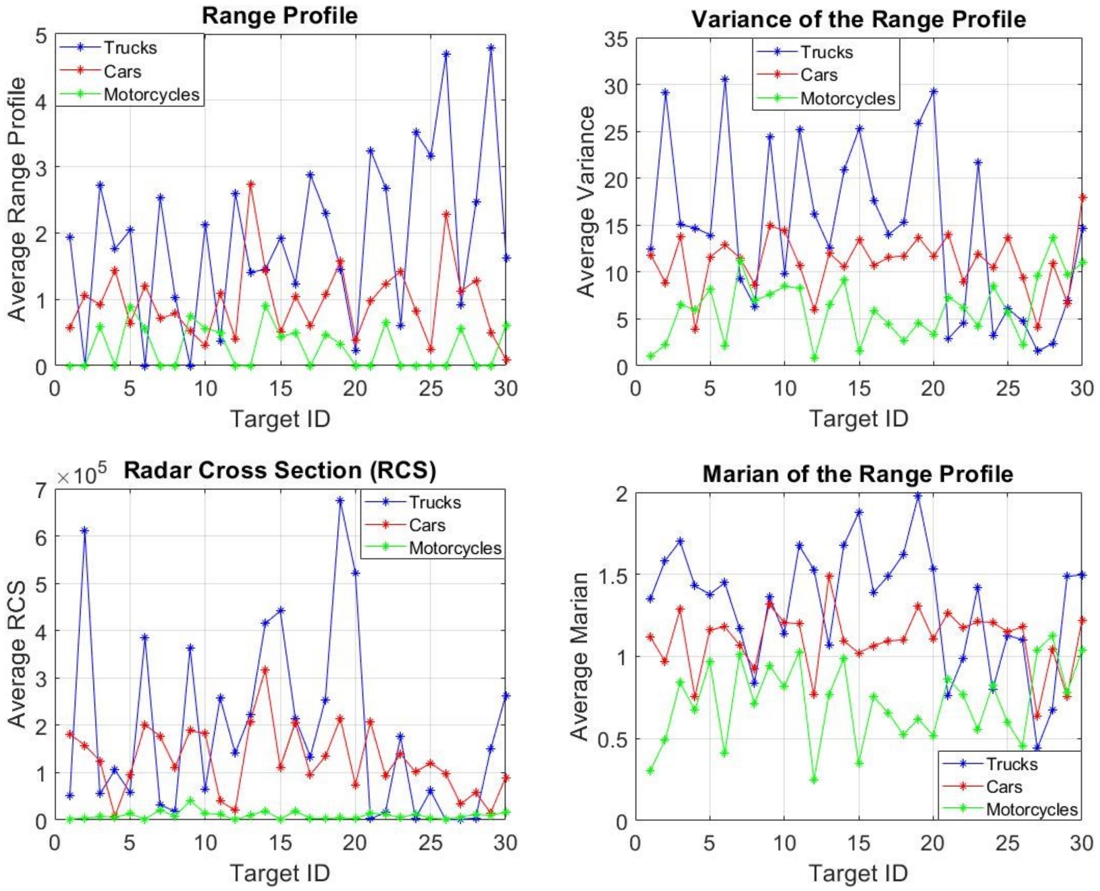


Figure 5.8 The most definitive features used in the differentiation of the vehicle class

based features, and the RCS measurements. The truck class shows to have, in most of the cases, higher measurements in regard to the range profile-based features, while the lowest measurement is mostly for the motorcycles. The car class measurements fall in between. This indeed agrees with the expectations as the truck class, in general, has bigger dimensions when compared to the other two classes. The dimension of the car then comes second, and the motorcycles are third with the lowest dimensions.

On the other hand, the dimension of the target is directly related to the measure of the reflectivity of the radar signal by the target. In other words, the bigger the dimension of the target the higher it can reflect the radar signal. This amount is also related to the RCS, which explains the results given in Figure 5.8 for the RCS measurements, where motorcycles have the lowest RCS measurements while trucks and cars have approximately equivalent measures with a slightly higher measurement for trucks in some cases. The separation between the three-

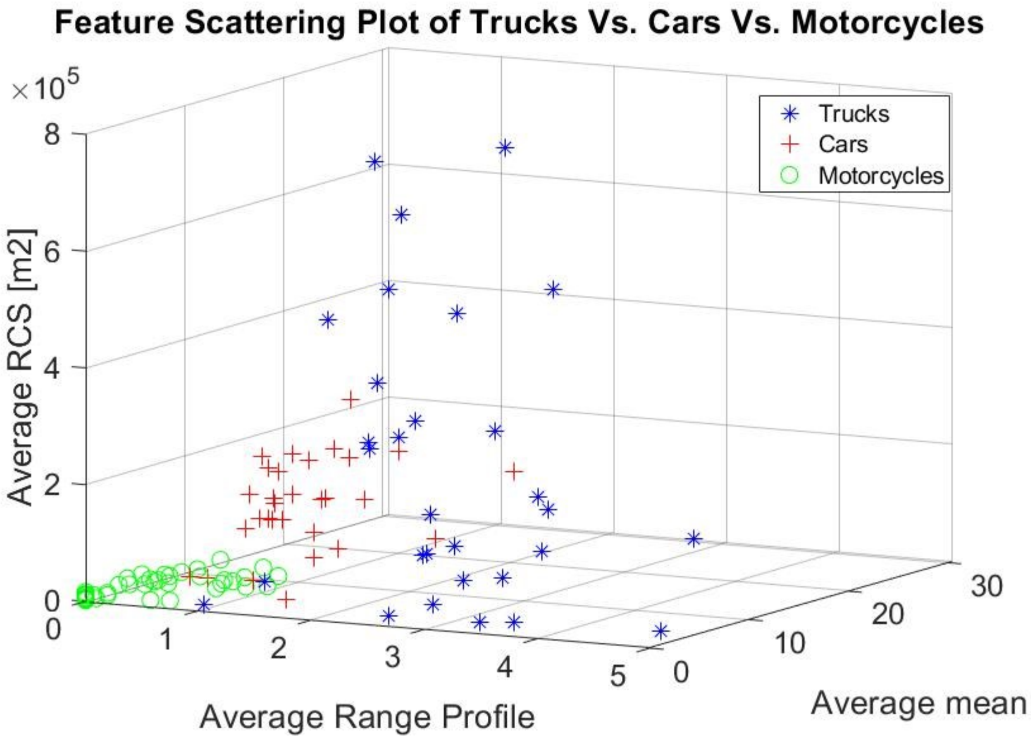


Figure 5.9 Comparison of the scattering plot of the three vehicle classes using three different features (RCS, range profile, and mean of the range profile)

vehicle classes is shown in Figure 5.9 using three distinctive features, i.e., RCS, range profile, and the mean of the range profile. A clear separation can be noticed between the motorcycle class and the two others, as the motorcycle scored the lowest values for the presented features. Also, a significant separation can be noticed between cars and trucks, as the car scores are mostly grouped in the mid-low ranges of the scattering plot, while the truck scores are scattered on the higher ranges of the scatter plot. The results shown in Figure 5.9 prove the robustness and the effectiveness of the chosen features to be used in the classification.

5.3 Summary:

In this chapter, a description of the collected data in the testing scenario was presented. In addition, some examples of some measurements taken from a pedestrian, vehicle, and multi-target situation were shown. The pedestrian's echo signal showed to have a single peak in the range spectrum and multiple peaks on the Doppler spectrum. This was translated to a spread on the Doppler axis of the Range-Doppler map. On the other hand, the measurements for the vehicle showed that the characteristics of the reflected echo signal differ depending on the position of the vehicle in the FOV of the radar (cross-range or along-range). In the case of a cross-range moving vehicle, a range and Doppler spread in observed, while a point-shaped target was detected in the case of an along-range moving vehicle. The multi-target measurements presented showed that the proposed system performs well in detecting multiple targets when presented at the same time in the illuminated scene. In addition, the extracted features that will be used for building the classification module, later on, were also discussed in detail. The results obtained showed the good performance of the presented features in discriminating between pedestrians and vehicles, and so they can be used for building classification module.

Chapter 6: Experimental Classification Results

To identify the types of objects in transit through the gate, the Support Vector Machines (SVM) classifier was adopted in this study. In particular, the One-Vs-All SVM was used which is a multi-class SVM classifier. The system was trained on a single target dataset collected by the system itself and tested on two different data sets collected in two different environments. The performance of the system was first tested in human-vehicle Classification, then in multi-class classification (pedestrian, motorcycle, car, and truck), and finally in multi-target classification. This chapter presents the evaluation phase of the proposed system in a real-world operation, where the system was tested in classifying the multiple targets passing in the FOV of the radar. The complete set of the parameters used during the training phase is presented in Table 6.1.

TABLE 6.1 SET OF PARAMETERS ADOPTED IN THE PROCESSING CHAIN

Symbol	Quantity	Value
Th_R	Range detection threshold	75 mV
Th_D	Doppler detection threshold	300 mV
r_{min}	Minimum detection range	1 m
r_{max}	Maximum detection range	25 m
K_R	Shape factor for the Kaiser window on the range dimension	3.2
K	Shape factor for the Kaiser window on the Doppler dimension	4
Th_{DBSCAN}	Threshold for DBSCAN clustering	505 mV
M	Minimum number of neighbour points for DBSCAN clustering	115
ε	Neighbourhood search radius for DBSCAN clustering	1.2
C_{NA}	Cost of non-assignment for the Hungarian algorithm	2.8
α	Alpha tracking parameter	0.9
β	Beta tracking parameter	0.3
T_f	Frame sampling time	200 ms

6.1 O-V-A SVM Classifier:

Support Vector Machine (SVM) [63] is a well-known algorithm that behaves well in many classification problems. On this basis, for our problem of the classification of objects passing through the monitored gate, we chose to adopt the One-vs-All multi-class SVM [61], [99]. This is a generalization of binary SVM capable of handling the multi-class scenario. Binary SVM is based on the construction of a hyperplane within the data space that maximizes the distance from the points of both classes that are closest to the plane itself (maximizing the margin). An example of binary SVM is shown in Figure 6.1.

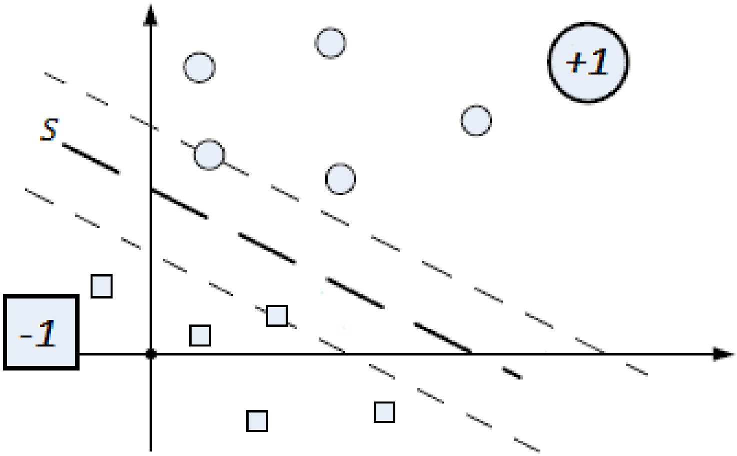


Figure 6.1 Example of binary SVM. The separation hyperplane is indicated with S.

The hyperplane can be formulated as follows:

$$\hat{\mathbf{w}} \cdot \mathbf{x} + \hat{b} = 0 \tag{6.1}$$

where $\hat{\mathbf{w}}$ is the normal vector to the hyperplane, $\mathbf{x}$ is an element belonging to the data space and $\hat{b}$ is the bias of the plane. Assuming the availability of a training set $X = \{\mathbf{x}_i\}_{i=1}^{N}$ with the relative labels $Y = \{y_i \in \{-1,1\}\}_{i=1}^{N}$, the maximization of the margin is obtained by determining the hyperplane parameters according to the following optimization problem:

$$\hat{\mathbf{w}}, \hat{b}, \hat{\boldsymbol{\xi}} = \arg\min_{\mathbf{w},b,\boldsymbol{\xi}} \left[\frac{1}{2}\|\mathbf{w}\|_2^2 + \delta \sum_{i=1}^{N} \xi_i\right], y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \geq 0, \xi_i \geq 0 \tag{6.2}$$

where $\boldsymbol{\xi} = [\xi_1 \quad \cdots \quad \xi_N]^T$ is the vector of the slack variables and δ is a hyperparameter which together with the sum of the slack variables forms the so-called penalty term. Moving on to the dual formulation of the optimization problem and introducing the use of non-linear kernels to manage classes that cannot be linearly separated in the data space, we arrive at the following reformulation:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \left[\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N} \alpha_i\right], 0 \leq \alpha_i \leq \gamma, \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{6.3}$$

where α_i are Lagrange multipliers and $K(\cdot,\cdot)$ is the kernel function. The belonging of a new element $\mathbf{x}$ to one of the two classes is determined as follows:

$$\text{sign}[s(\mathbf{x})] = \begin{cases} > 0 \Rightarrow y = 1 \\ < 0 \Rightarrow y = -1 \end{cases} \tag{6.4}$$

where the function

$$s(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{6.5}$$

is called the classification score. The generalization to the multi-class scenario can take place through the adoption of approaches using error-correcting output codes (ECOC), which allow transforming a multi-class classification problem into a set of binary classification problems. In particular, the One-Vs-All (OVA) method was adopted in this project, which starts from the construction of a coding matrix. By way of example, in the case of a scenario with 3 classes, this matrix is equal to:

	Classifier 1	**Classifier 2**	**Classifier 3**
Class 1	1	-1	-1
Class 2	-1	1	-1
Class 3	-1	-1	1

When the ij-th element takes the value 1 or -1 then the j-th classifier will consider the elements of the training set belonging to the i-th class as having respectively 1 or -1 label. More generally, in the presence of C classes, C binary classifiers are identified, and the coding matrix indicates to each of these which elements of the dataset must be considered for training. Once the C optimization problems (6.3) have been solved, a new datum **x** (new observation) is classified by interrogating all the binary classifiers, then collecting the scores $s_j(\mathbf{x})$, with $j = 1, \ldots, C$, and attributing to the new element the i-th class that minimizes the following quantity, called aggregation loss:

$$\frac{\sum_{j=1}^{C} |m_{ij}| h[m_{ij}, s_j(\mathbf{x})]}{\sum_{j=1}^{C} |m_{ij}|} \tag{6.6}$$

where m_{ij} is the ij $-$th element of the coding matrix and

$$h[m_{ij}, s_j(\mathbf{x})] = \frac{1}{2}\{1 - m_{ij}[2s_j(\mathbf{x}) - 1]\}^2 \tag{6.7}$$

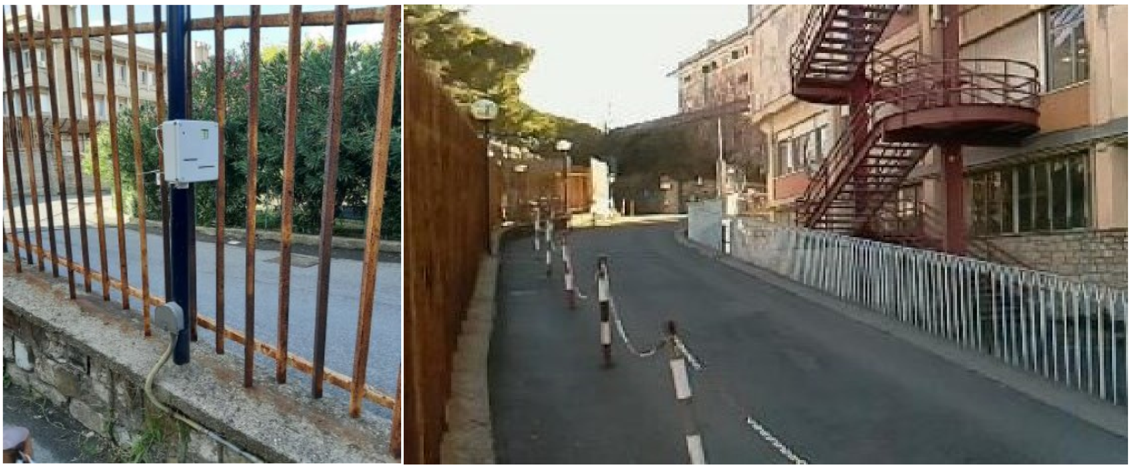
is called the quadratic binary loss function.

6.2 Single Target Classification:

This section presents the first evaluation of the system in classifying single targets passing in the FOV of the radar. At first, the performance of the system was tested in a human-vehicle classification where two approaches were followed in that regard, i.e., transfer learning and binary SVM. After, the performance of the system was tested in a multi-class classification where the detected single targets were classified as either: pedestrian, car, truck, and motorcycle.

6.2.1 Measurement Setup:

Our data acquisition system is shown in Figure 6.2. The data collection was done using the “Distance2Go” radar board by Infineon Technologies [80]. The system has been put into a box and mounted on a pole, beside an internal road of the department at a height of approximately 1.5 m. The box has been oriented in such a way that the main beam of the antenna forms an angle of 30 degrees with respect to the direction of the road. Moreover, the Half Power Beamwidths (HPBWs) on the horizontal and vertical planes are 42 degrees and 20 degrees, respectively [80]. Only the targets moving in the FOV defined by such HPBWs have been considered. The camera was only used for observation purposes and for marking the type of the passing targets. A MATLAB script was used to collect the radar raw data wirelessly with the help of a Raspberry Pi 3B+ embedded PC. The Raspberry PI collects the radar data by means of a USB cable and then transmits them to MATLAB via a WiFi network (Figure 2.1 & Figure 2.2). The data were collected for two classes: humans and vehicles. The data acquisition procedure was triggered only when a target passes in the FOV of the radar with a peak in the range-FFT spectrum higher than a previously fixed threshold.



(a) (b)

Figure 6.2 (a) Positioning of the external unit in the first test scenario. (b) Picture of the environment of the first test scenario.

6.2.2 Human-Vehicle Classification Using Transfer Learning:

For the sake of the best classification performance, the R-D maps obtained by the radar system should be clean and the target signature in the image must be self-evident. This indicates the importance of using the data collection system in its most optimized state (Table 2.1) and applying a detection and calibration technique, able to suppress noise and clutters as much as possible. The resolution and size of the Range-Doppler image depend on the number of FFT points used to apply the 2D FFT processing. We used $RFFT = 512$ points for range-FFT, and $DFFT = 256$ points for Doppler-FFT, producing a (256×256) Range-Doppler image.

Figure 5.2 illustrates some examples of R-D maps obtained for pedestrians and vehicles. In general, both R-D maps for humans and vehicles suffer from a speed ambiguity problem. The problem of the ambiguity arises from the fact that the speed of the target in many cases, especially for cars, is higher than the maximum speed the radar can measure, which is about $5\ Km/hr$. This ambiguity problem causes the presence of R-D maps that are incapable of reflecting the true Doppler signature since they have incorrect measurements on the speed axis. In addition, the low resolution of the R-D maps cannot be improved due to the inability to use higher values of N_c and N_s.

The Convolutional Neural Networks (CNNs), proved over the years to be very efficient in images classification. Since R-D maps can actually be considered as images [77], applying CNNs for classification is consistent in this case. Consequently, feature selection was not introduced in the proposed method: instead, the Range-Doppler maps obtained from the 2D FFT processing of the radar data were transformed into images and fed to a convolutional neural network (CNN) trained on ImageNet [100] - a dataset containing millions of visually identified labeled images categorized into more than one thousand classes - in order to benefit from feature extraction layers. Only the last classification layer is retrained using the data set collected with the developed system. Figure 6.3 illustrates this transfer learning concept from domain A to domain B [101], where the same feature extraction layers trained for domain A are used in the case of domain B, and then only the classification layer is re-trained with domain B data. The re-trained CNN is then used in the proposed solution.

As we have previously shown in Figure 5.2, the pedestrian's R-D map has a vertical distribution along the speed axis indicating the presence of different velocities for the different parts of the body. An along-range moving vehicle R-D map shows only a point-shaped target which represents one beat and Doppler frequencies. However, a cross-range moving vehicle's R-D map has a broad distribution along the range and speed axis, which represents the presence of

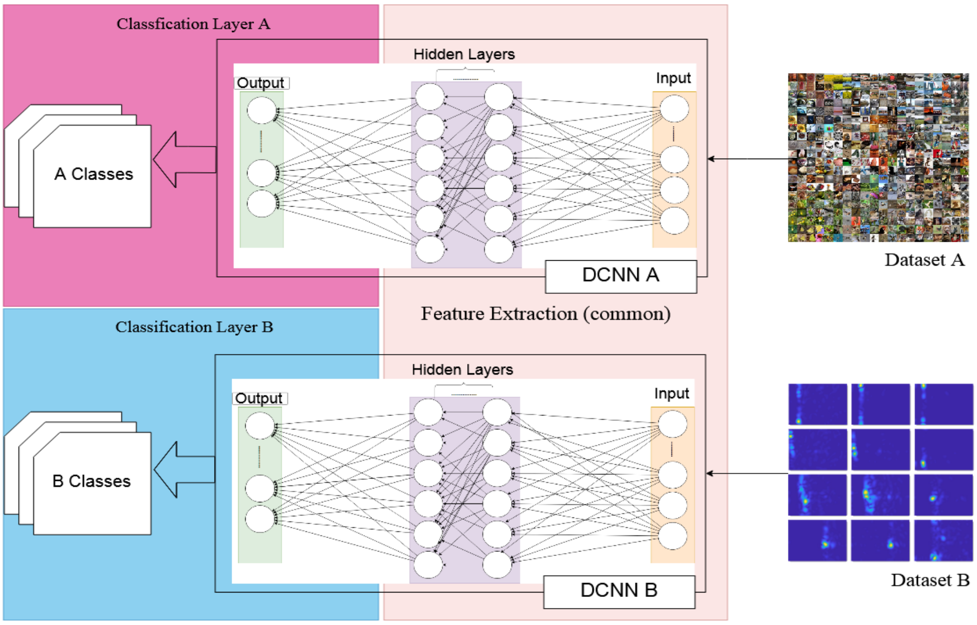


Figure 6.3 Flow of transfer learning from Dataset A to Dataset B

a variety of beat and Doppler frequencies. The difference between the patterns of pedestrians and vehicles (Figure 5.2) is very clear to human eyes. Consequently, a CNN model is also expected to be able to differentiate between them.

For the used CNN model, the input image size used is 256 × 256 pixels. The images in this experiment have been represented in RGB color mode. The experiments for the proposed solution were made under the following setup:

- Hardware: Retraining the network and the inference were made on the Jetson TX2 development kit donated by Nvidia Corporation [102].
- Software: Tensor flow 1.8.0, Operating System: Ubuntu 16.04 LTS.
- Dataset: 500 images distributed evenly among two classes: car and human, each image is of size: 256 × 256.
- Neural network: Inception Resnet [103].

The experiment was made using 80% of the dataset for training, 10% for validation, and 10% for testing. The training/ validation/ testing split was randomly selected.

The used CNN model achieved a great performance in discriminating between humans and vehicles. The training and test accuracies scored 97.8% and 96.5% respectively. The 96.5% represents the overall accuracy of the system. The misclassification for both classes is 3.5%, which indicates the effectiveness of such a classifier for our application.

6.2.3 Human-Vehicle Classification Using Binary SVM:

In order to give a better insight into the robustness of our radar classification system, a binary classification test involving human-vehicle classification was carried out. The kernel function used for SVM in this study is the Gaussian kernel [104]. The feature set in Table 5.1 was used to form the feature vector. The feature vector fed into the SVM classifier is obtained by averaging the data samples collected for each passing target. The averaging process is done also for both the training set and the testing set. In this study, the classification is only done for a single target scenario.

As detailed in the previous section, the feature vector $\mathbf{x}_j$ at the j^{th} time frame upon which the classification was carried on is defined as:

$$\mathbf{x}_j = \left[R_{profile}, R_1, R_2, R_3, V_{profile}, V_1, V_2, V_3, \gamma, v_{av}\right]^t \tag{6.8}$$

The feature vector was normalized using a mean normalization scheme. In particular, the l-th component of the normalized feature vector $\mathbf{z}_j$ is obtained as follows:

$$z_j^l = \frac{x_j^l - \bar{x}_j^l}{x_{j,max}^l - x_{j,min}^l} \tag{6.9}$$

where $\bar{x}_j{}^l$, $x_{j,max}^l$, and $x_{j,min}^l$ are the mean, maximum, and minimum value of the l-th component of $\mathbf{x}_j$ in the training set.

As for the binary and OVA SVM method, the Gaussian kernel has been adopted, which is defined as follows:

$$K(\mathbf{x}_a, \mathbf{x}_b) = e^{-\rho\|\mathbf{x}_a - \mathbf{x}_b\|_2^2} \tag{6.10}$$

where ρ is a parameter to be tuned for the problem at hand.

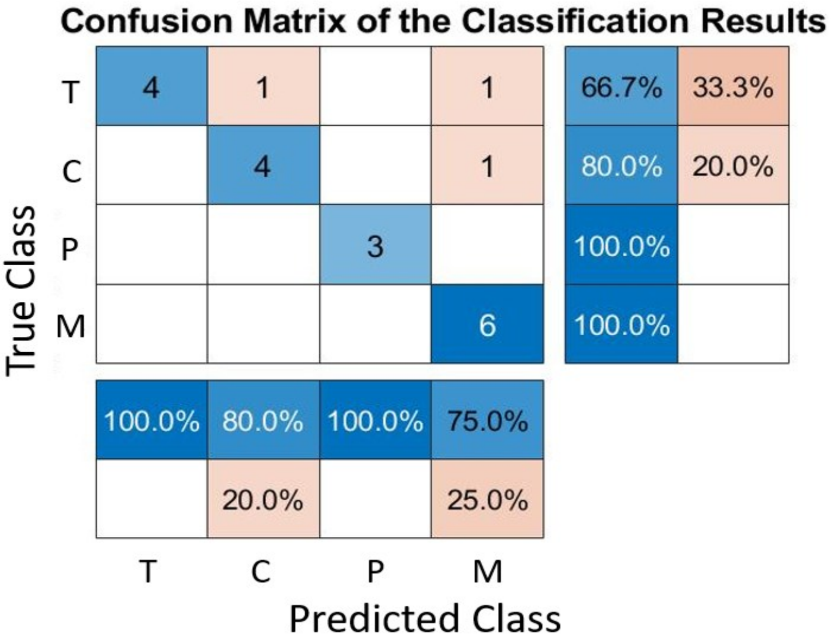
For human-vehicle binary classification, 90 samples were collected for each class forming a dataset of 180 samples. The dataset was separated as 80% of the data were used for training and the other 20% for testing. The training data and the testing data were both randomly selected. The cross-validation criterion was done using 10% of the training data.
The classification performance in this scenario is perfect, which shows the robustness of the proposed system in classifying pedestrians from vehicles using the proposed feature vector.

6.2.4 Multi-Class Classification Using OVA SVM:

The four target classes are pedestrians, trucks, cars, and motorcycles. In this test, since only 30 samples were collected for class trucks, we were forced to equilibrate the samples for the other classes with the truck class. In that regard, the dataset for the four target classes consists of 120 samples, in which each target has 30 samples. The dataset again was separated into training and testing, where 100 randomly chosen samples were used for training (25 for each) and the remaining 20 samples were used for testing (used after the optimization of the classifier to assess the results). The cross-validation here was also done using 10% of the training set. Knowing that the maximum distance that can be measured by the radar is 24 m, the distance where the radar starts to detect and collect data samples for pedestrians ranges between 12 to 16 m, whereas it ranges between 20 to 24 m for vehicles. No specific conditions were considered in this experiment other than placing the radar perpendicular to the road and directly facing the passing targets.

The One-vs-All Multi-class SVM version [61], [99], [105] was used in this test. This is a generalization of binary SVM capable of handling the multi-class scenario.

The OVA-SVM has been trained by using the Bayesian global optimization algorithm [106] available in the MATLAB “Statistics and Machine Learning Toolbox”. Moreover, a K-fold cross-validation scheme, with $K = 10$, has been adopted to find suitable values of the hyper parameters C and ρ, where C is the SVM regularization parameter [107], [108]. Figure 6.4 shows the confusion matrix obtained by applying the SVM obtained after this optimization on the test set, where class T stands for trucks, C stands for cars, P stands for pedestrians and M


Confusion Matrix of the Classification Results
True Class
Predicted Class

	T	C	P	M		
T	4	1		1	66.7%	33.3%
C		4		1	80.0%	20.0%
P			3		100.0%	
M				6	100.0%	
	100.0%	80.0%	100.0%	75.0%		
		20.0%		25.0%		

Figure 6.4 Confusion matrix obtained by applying the trained OVA-SVM to the single-target test data.

for motorcycles. These must give the method good generalization skills, i.e., that it may be able to obtain good classification performances even on elements that are not part of the training set.

The obtained classifier is then used for the multi-target classification, as detailed in the following Section.

6.2.5 Pedestrians and Multi-Class Vehicle Classification using Rulex Software:

Rulex is a machine learning software that supports various machine learning algorithms that can be easily applied in a user-friendly environment [109]. The Rulex Graphical User Interface (GUI) provides a means of importing training data and manipulating them before applying machine learning algorithms. The main proprietary algorithm for Rulex is the logic learning machine (LLM) [110], [111] which implements explainable AI. LLM was the main algorithm used for most of the classifications, where a tree-based structure, which combines vehicle classes to achieve more accurate results, was adopted. Then, these new combined classes are split recursively until all vehicles have been classified in their respective sub-classes.

- **Classification Methodology:**

The same four classes mentioned earlier were also considered here. First, running forecasts by relying on multiple algorithms in one overall simulation was performed. Rulex possesses various algorithms to choose from, such as neural networks (NN), k-nearest neighbor (KNN), decision trees, support vector machines, and logic learning machine (LLM), all of which can be used for classification. However, since the adopted methodology applies multiple ML algorithms in a cascaded setup and tests multiple arrangements, which can have a large number of forecasts, only NN and LLM were included. This was adopted since NN is a widely used ML algorithm and because LLM is the commercial algorithm of Rulex. Furthermore, for NN, a multi-layer perceptron (MLP) arrangement was applied using the back propagation algorithm [112]. However, since the vehicle class can be split into three sub-classes, namely, cars, trucks, and motorcycles, and since there is no need to differentiate between them in real-world scenarios, it was possible because of this fact to apply a sub-class-based tree structure, where the classes are nodes, and the algorithms are branches. The forecasts of the tree-based method

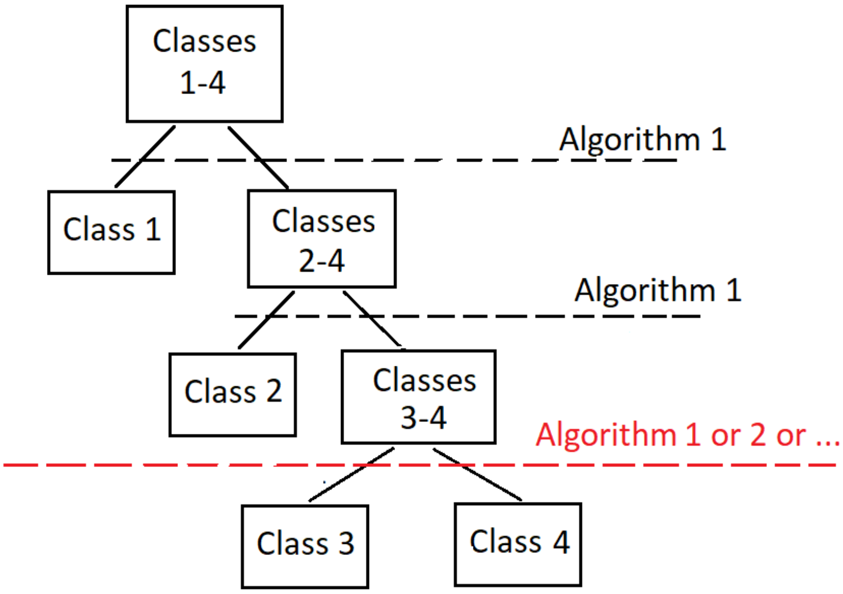


Figure 6.5 Sub-Class based Tree Structure.

can lead to improved validation results by setting different weights in each forecast. Therefore, we can choose a convenient split ratio between training and testing data, separately for each forecast, where the split method used is holdout validation. It is also possible to use different classification algorithms in each forecast on each branch. This is useful in case the adopted algorithm is not generating the expected results for the dataset at hand. Figure 6.5 presents such a tree structure, where the leaves are the final classification outputs.

As presented, different types of algorithms were applied to the final branches of the tree since they possess the lowest success rate. Therefore, the prediction accuracy is optimized by varying weights, data splitting, and, more critically, the algorithm used for classification. The further we go down the tree, the harder it gets to differentiate between classes.

- **Results and Discussion:**

The machine learning system used consists of the Rulex Engine running on the Raspberry Pi as an application server, which is where forecasts are applied. This engine is accessed through a graphical interface running on a Windows client, while a PostgreSQL server is used as the common storage point between both nodes.

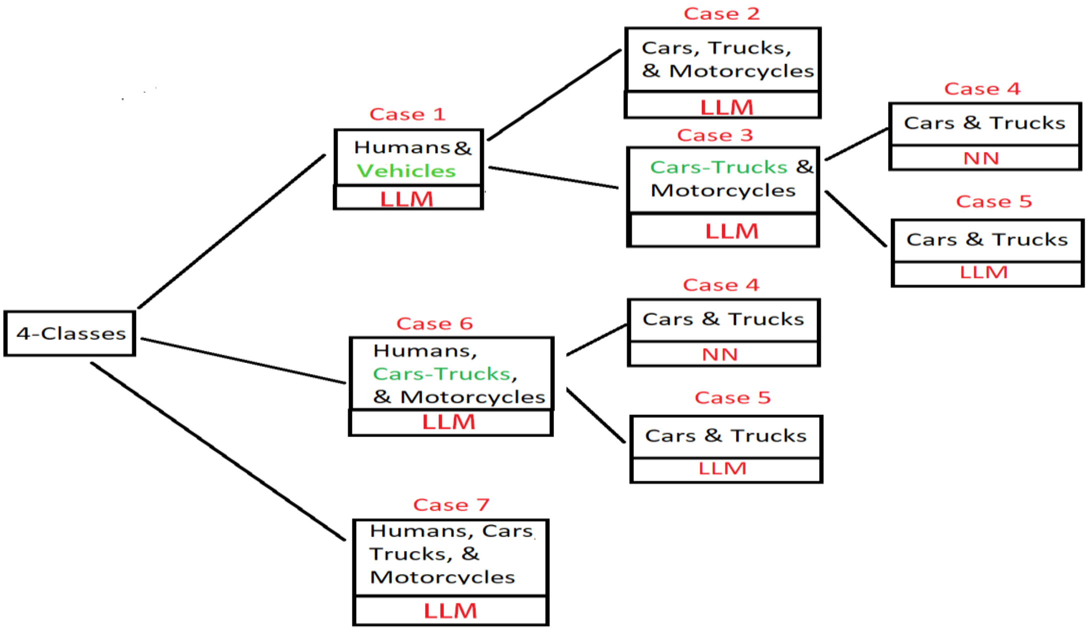


Figure 6.6 Cases 1 – 7 featuring all applied forecasts presented in this study.

The dataset collected by the data acquisition system described before consists of 120 rows equally distributed into 4 classes with 30 patterns for every class. The features set mentioned earlier in Table 5.1 was adopted. In order to maximize forecast accuracy, multiple tree-based sub-class arrangements were applied to simulate using Rulex as a client/server setup [113].

As stated earlier, multiple cascaded simulations were applied with a varying number of classes as well as a cascaded order. Thus, a summary of all the applied forecasts is presented in Figure 6.6 and described in detail in this section. In Figure 6.6, the red labels stand for the cases and machine learning algorithms used in that particular simulation, and the green labels represent classes that will be split into sub-classes in the upcoming simulation.

In Case 1, the data was split into two classes: humans, and vehicles where LLM was used.

The accuracy is shown in Table 6.2.

In Case 1, the machine learning algorithm used is LLM for classification. However, in Case 2, only vehicle sub-classes are considered. The simulation was applied using LLM where the prediction accuracies are found in Table 6.3.

TABLE 6.2 HUMANS AND VEHICLES TRAINING AND TESTING PREDICTION ACCURACY.

Class	Training	Testing
Humans	100%	100%
Vehicles	95%	100%

In Case 3, a forecast using LLM was performed for vehicle classes by splitting the data into two classes as shown in Table 6.4. In Cases 4 and 5, the cars/trucks class has been split into two sub-classes, cars, and trucks. In Case 4, the neural network was used, whereas LLM has been used in Case 5. The results of these can be found in Table 6.5 and Table 6.6, respectively.

TABLE 6.3 VEHICLES, THREE CLASSES TRAINING AND TESTING PREDICTION ACCURACY.

Class	Training	Testing
Cars	96%	75%
Motorcycles	94%	91%
Trucks	100%	72%

TABLE 6.4 CARS/TRUCKS AND MOTORCYCLES TRAINING AND TESTING PREDICTION ACCURACY.

Class	Training	Testing
Motorcycles	100%	91%
Cars/Trucks	98%	100%

TABLE 6.5 CARS AND TRUCKS WITH NEURAL NETWORKS TRAINING AND TESTING PREDICTION ACCURACY.

Class	Training	Testing
Cars	95%	87%
Trucks	100%	70%

TABLE 6.6 CARS AND TRUCKS WITH LLM TRAINING AND TESTING PREDICTION ACCURACY.

Class	Training	Testing
Cars	95%	100%
Trucks	100%	66%

Cases 1 to 5 were processed separately to get a glimpse of how LLM would perform with this given dataset. From the outputs generated in Tables 1–5, it is possible to estimate the overall prediction accuracy for a cascaded setup. Furthermore, it should be noted that misclassified records in preceding forecasts will be treated as correctly classified in upcoming forecasts, which leads to the overall accuracy of the cascaded system being incorrectly estimated. The preceding forecasts were all performed with a 70%/30% split for training and testing data, respectively, and with all weights being set to unity.

TABLE 6.7 CASES 1 THEN 2 PREDICTION ACCURACY.

Class	Forecast
Humans	100%
Motorcycles	91%
Cars	75%
Trucks	72%

Furthermore, multiple cascaded setups can be applied which are based on the previous forecasts. When cascading cases 1 and 2, the projected output is presented in Table 6.7. In the case where cases 1, 3, and 4, cascaded the projected output is provided in Table 6.8.

TABLE 6.8 CASES 1, 3 AND THEN 4 PREDICTION ACCURACY.

Class	Forecast
Humans	100%
Motorcycles	91%
Cars	87%
Trucks	70%

When cases 1, 3, and 5 are cascade, the obtained results are shown in Table 6.9. Other variations of initializing the cascaded system with LLM can be found in Table 6.10, which is case 6, where one class for humans along with two classes for vehicles is taken.

TABLE 6.9 CASES 1, 3 AND THEN 5 PREDICTION ACCURACY.

Class	Forecast
Humans	100%
Motorcycles	91%
Cars	100%
Trucks	66%

TABLE 6.10 HUMANS AND VEHICLES 2-CLASSES TRAINING AND TESTING PREDICTION ACCURACY.

Class	Training	Testing
Cars/Trucks	97%	94%
Motorcycles	94%	100%
Humans	94%	69%

Finally, a single forecast for all four classes which is applied using LLM is presented in Table 6.11, namely, case 7, which consists of forecasting all classes in a single block. With the variation added in Table 6.10 and Table 6.11, two additional combinations can be applied to cascade. Thus, case 6 with case 4 can be cascaded, which employs neural networks, or it can be cascaded with case 5 which uses LLM. These last two combinations include a situation where the previous prediction was not 100% accurate, which needs to be considered when theoretically estimating the overall accuracy. When combining case 6 with case 4, the cars/trucks class has an accuracy of 94%, so, naturally, the neural network predictions in case 4 will be multiplied by 0.94. The same can be said for case 5, where the cars and trucks classes success rates are multiplied by the same factor. Table 6.12 and Table 6.13 provide the projected output forecast accuracy for the last two scenarios.

TABLE 6.11 DEFAULT LLM FORECAST TRAINING AND TESTING PREDICTION ACCURACY.

Class	Training	Testing
Humans	100%	73%
Cars	100%	100%
Motorcycles	84%	91%
Trucks	90%	72%

TABLE 6.12 CASES 6 THEN 4 PREDICTION ACCURACY.

Class	Forecast
Humans	100%
Motorcycles	69%
Cars	$0.94 * 87 = 82\%$
Trucks	$0.94 * 70 = 66\%$

All the preceding simulations only provide an estimation of actual results when cascading multiple engines. This is due to not considering the false-positive cases of the forecast. The entire dataset was taken into consideration for each algorithm block while ignoring some of the rows which were correctly classified in the abandoned class. In case two algorithms are cascaded, the first block should be followed by a Rulex data management block which will filter out the true and false positives in the abandoned class and remove them from the table.

TABLE 6.13 CASES 6 THEN 5 PREDICTION ACCURACY.

Class	Forecast
Humans	100%
Motorcycles	69%
Cars	$0.94 * 100 = 94\%$
Trucks	$0.94 * 66 = 62\%$

However, the multiplication of the proceeding blocks with their parent class's success rate still must be done to calculate the overall accuracy. For case 1, the data was split $70/30$, with 70% used for training and 30% being used for testing. The same was applied for case 3. However, for case 4, with the reduced number of rows, the data were split $65/35$, with 65% used for training and the remaining 35% being used for testing. The main reason for changing the split ratio in case 4 is due to the fact the prediction is applied to half of the dataset, and it was found that increasing the size of the test set can lead to higher accuracy for the given data.

As for weights, the only way to set them and optimize results is by trial and error and intuition. There is no universal method to select weights accordingly. The unity gain in the case 1 block should already provide very good results, so there is no need to change the weights. With a unity gain, in case 3, the cars/trucks class, which will be used in the proceeding block, should be accurate while keeping the motorcycles class forecast precise enough. A gain of 1.5 was chosen for the cars/trucks field and 1.0 for motorcycles. As for the final block, which is case 4, both trucks and cars classes, which originate in case 3, have an equal true positive rate of 80% in testing. Therefore, weights are left at unity.

Table 6.14 represents the accuracy for training and testing of cases 1, 3, and 5, respectively, and as predicted using Rulex. All the forecasts present good results for testing. Humans were detected with a rate of 100% and vehicles overall at a rate of 96.67%. In case 3, which is block 2, the cars/trucks class has a true positive rate of 93.75% and motorcycles at 90%. As for the cars and trucks block, which is case 5, the success rate is 80% for both trucks and cars. Table 6.15 and Table 6.16 present the overall output true and false-positive rates for the chosen All-LLM forecast. Humans are detected without any errors for the test dataset. The overall forecasts of the motorcycles, cars, and trucks have been calculated based on the preceding forecasts to become 90.63% for motorcycles and 77.34 for both cars and trucks.

TABLE 6.14 TRAINING AND TESTING ACCURACIES FOR CASES 1, 3, AND 5 AS PREDICTED USING RULEX BEFORE COMPUTING THE ACTUAL ACCURACIES.

Case	**Training**		**Testing**	
Case 1	**Humans**	**Vehicles**	**Humans**	**Vehicles**
	100%	95.82%	100%	96.67%
Case 3	**Cars/Trucks**	**Motorcycles**	**Cars/Trucks**	**Motorcycles**
	100%	100%	100%	93.75%
Case 5	**Cars**	**Trucks**	**Cars**	**Trucks**
	95%	100%	80%	80%

TABLE 6.15 MAIN FORECAST.

Class	**Forecast**
Humans	100%
Vehicles	96.67%

TABLE 6.16 VEHICLES FORECAST.

Class	**Forecast for Vehicles**
Motorcycles	$0.9667 * 93.75 = 90.63\%$
Cars	$0.9667 * 80 \ = \ 77.336\%$
Trucks	$0.9667 * 80 \ = \ 77.336\%$

6.3 Multi-Target Classification:

This section presents the evolution of the system performance in classifying multiple targets passing in the FOV of the radar. The system is first tested on a multi-target dataset collected in the first testing environment. After that, its performance was validated using a multi-target dataset collected in a different environment with different environmental conditions.

6.3.1 Multi-Target Validation in the Testing Scenario:

After training the OVA-SVM, the performance of the system in classifying multiple targets has been evaluated by considering the first testing scenario of Figure 6.2. The experimental validation was performed on 93 multi-target samples, which belong to a different dataset from the one used for training. Figure 6.7 shows the classification performance of the proposed method on the whole dataset. The perfect classification score indicates that the proposed method was able to classify all the targets correctly, while the imperfect classification indicates that not all the targets were classified correctly. The bad classification score indicates that none of the targets were correctly classified. Some specific examples that were presented in section 4.4.7 are considered here to validate the performance of the system in multi-target classification.

The first test case consisted of a pair of pedestrians moving away from the radar and a car also moving away. Figure 4.3 illustrates some moments of the recorded event and the related

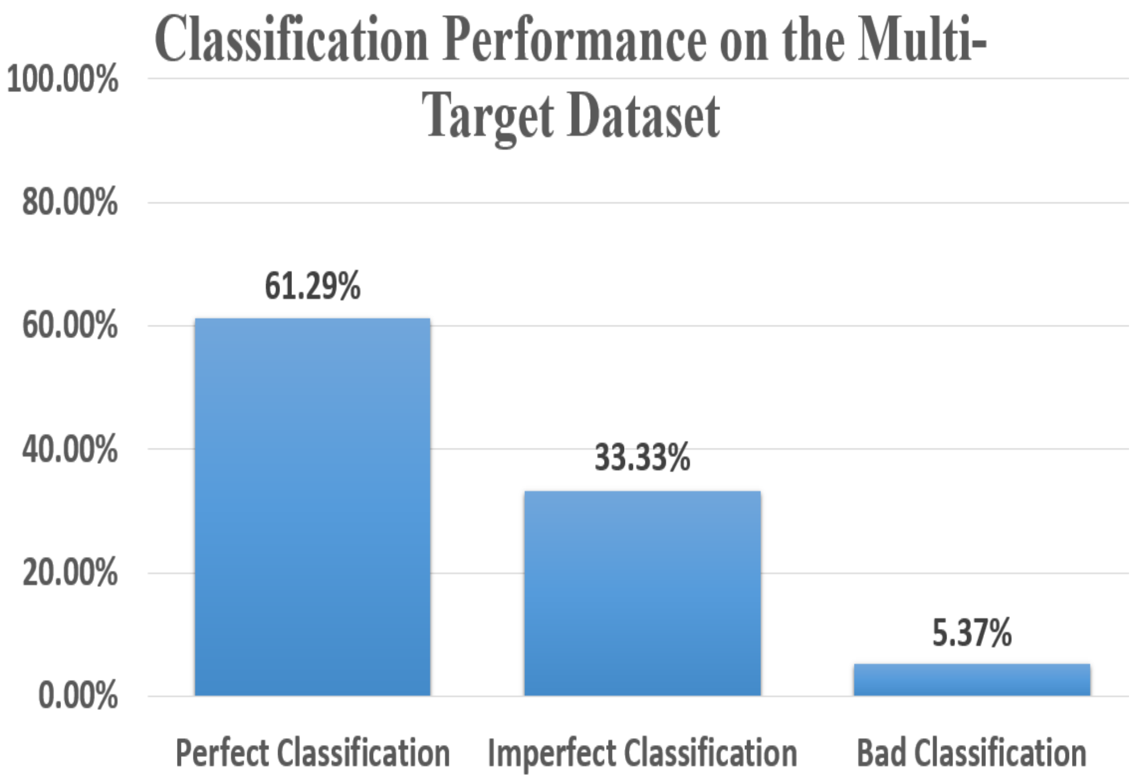


Figure 6.7 Performance of the proposed method in classifying targets from the multi-target dataset.

tracking results (the different colors are used to denote different classes of targets in the image). The two pedestrians were detected and tracked as a single target. All the targets are correctly classified as shown in Table 6.17, which also indicates the aggregation losses calculated for each class and the estimated average radial speeds. Note that recognition occurs with wide margins compared to competing classes. The estimated average radial speeds are compatible with the motion of the targets in the scene.

The second test case consisted of a pedestrian and a truck (Figure 4.4). The aggregation losses and the estimated average radial speeds reported in Table 6.17 highlight the robustness of the classification and the goodness of the tracking.

The third test case contained two pedestrians, one of whom is approaching and one moving away, and subsequently an approaching motorcycle. From Figure 4.5(b) and Figure 4.5(d) it

Table 6.17 Recognition of the type of target made on the stored tracks, aggregation loss and estimated average radial speeds. P = pedestrian, M = Motorcycle, T = Truck, C = Car.

Real class	***Predicted class***	***Aggregation loss***				***Average radial velocity*** **[km/h]**
		C	***P***	***T***	***M***	
Test 1						
P	*P*	*0.55*	*0.06*	*0.68*	*0.44*	*3.2*
P	*P*	*0.57*	*0*	*0.76*	*0.65*	*-3.4*
M	*M*	*0.67*	*0.5*	*0.71*	*0.03*	*-33.7*
Test 2						
P	*P*	*0.66*	*0*	*0.62*	*0.82*	*2.3*
C	*C*	*0.01*	*0.79*	*0.49*	*0.84*	*26.5*
Test 3						
P	*P*	*0.63*	*0*	*0.7*	*0.56*	*-5.8*
T	*T*	*0.46*	*1.06*	*0.04*	*0.94*	*-15.3*
Test 4						
P	*P*	*0.52*	*0.01*	*0.53*	*0.66*	*2.55*
C	*C*	*0.22*	*0.73*	*0.55*	*0.28*	*19.36*
M	*M*	*0.43*	*0.37*	*0.42*	*0.17*	*16.92*
P	*P*	*0.6*	*0*	*0.81*	*0.73*	*4.4*

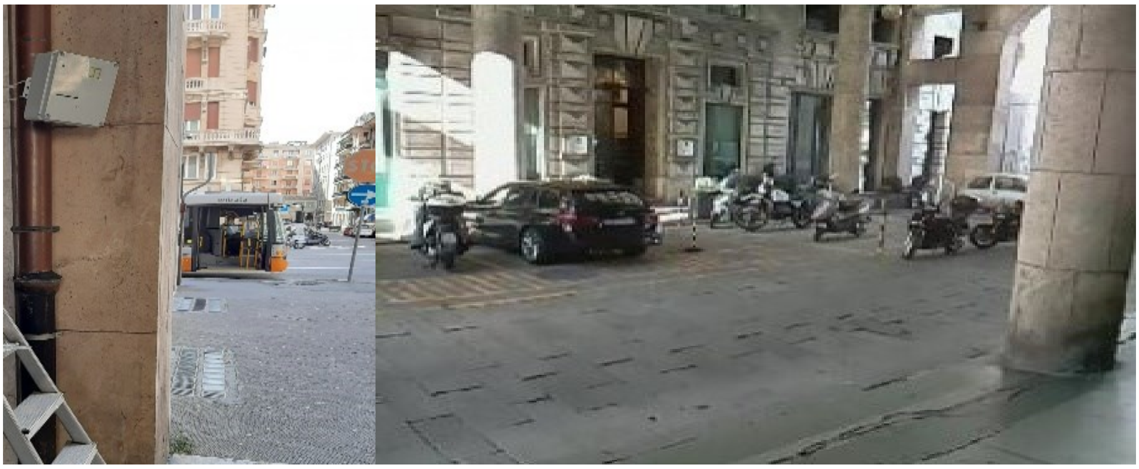

(a) (b)

Figure 6.8 (a) Positioning of the external unit in the second test scenario. (b) Picture of the environment of the second test scenario.

can be seen how the method is able to follow the trend at the constant speed of their trajectories. The result of the classification, the aggregation losses, and the estimated average radial speeds are shown in Table 6.17.

In the last example case of Figure 4.6 (test case 4), the scene started with the presence of two pedestrians very close to each other, they were recognized and tracked as a single target. In Figure 4.6(b) the movement of the pedestrians was disrupted by a passing car followed by a passing motorcycle. Following the disappearance of the motorcycle from the FOV of the radar in Figure 4.6(d), a new track was created for the newly detected pedestrians. The aggregation losses and the estimated average radial speeds reported in Table 6.17 highlights the robustness of the classification.

6.3.2 Multi-Target Validation in a Different Scenario:

To confirm the robustness of the proposed method, another experiment was done in a totally different environment and under different conditions. The radar was mounted on a pole near a side public road at 3 m height and at about 5 m distance from the street, tilted with a 45-degree angle toward the street (Figure 6.8).

A total of 37 samples were collected for this test. The training of the OVA-SVM was not performed, since the old training parameters found for the first scenario were used. However, the different environmental conditions required to retune some of the parameters of the algorithms used in the developed method. In particular, the higher distance of the target from the radar compared to the first scenario required to decrease the range and DBSCAN detection thresholds for obtaining an improved detection of the target contribution in the range spectrum and the resulting DBSCAN clusters. On the other hand, the higher cross-range movement of the targets with respect to the radar compared to the along-range one in the first scenario imposed higher variations in the Doppler spectrum. This is mainly due to the greater variation in the radial velocity of the targets, and thus a higher Doppler threshold should be adopted to avoid detections caused by noise. Such considerations, together with preliminary empirical tests on some targets, have been used to retune the aforementioned parameters. The values of

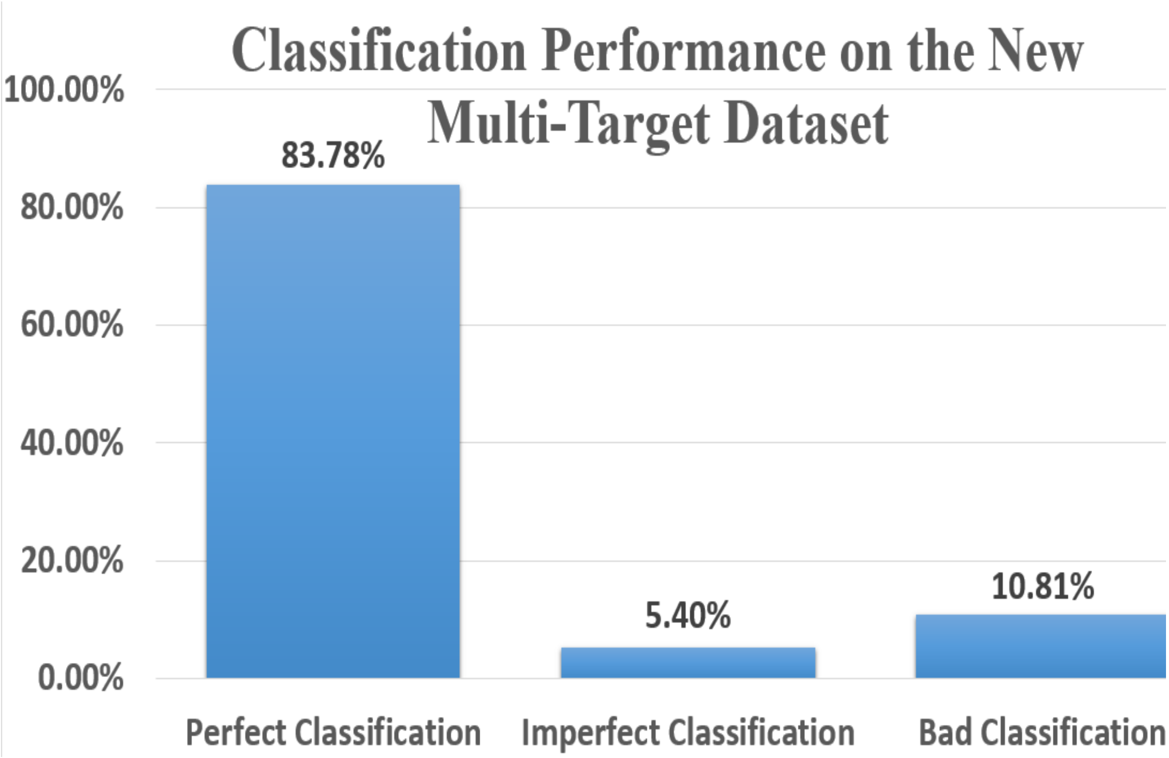
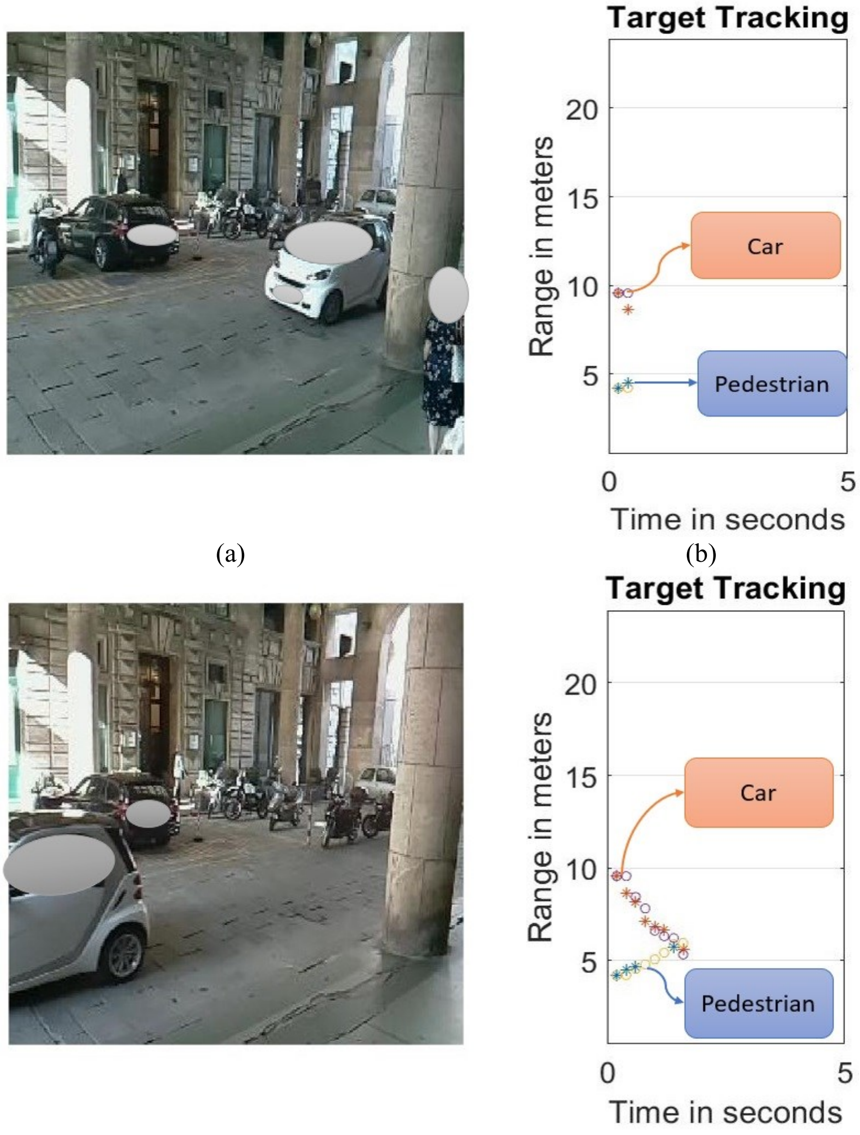


Figure 6.9 Performance of the proposed method in classifying targets from the second multi-target dataset.

TABLE 6.18 MODIFIED PARAMETERS ADOPTED IN THE PROCESSING CHAIN (EXPERIMENT 2)

Symbol	Quantity	Value
Th_R	Range detection threshold	50 mV
Th_D	Doppler detection threshold	500 mV
Th_{DBSCAN}	Threshold for DBSCAN clustering	260 mV

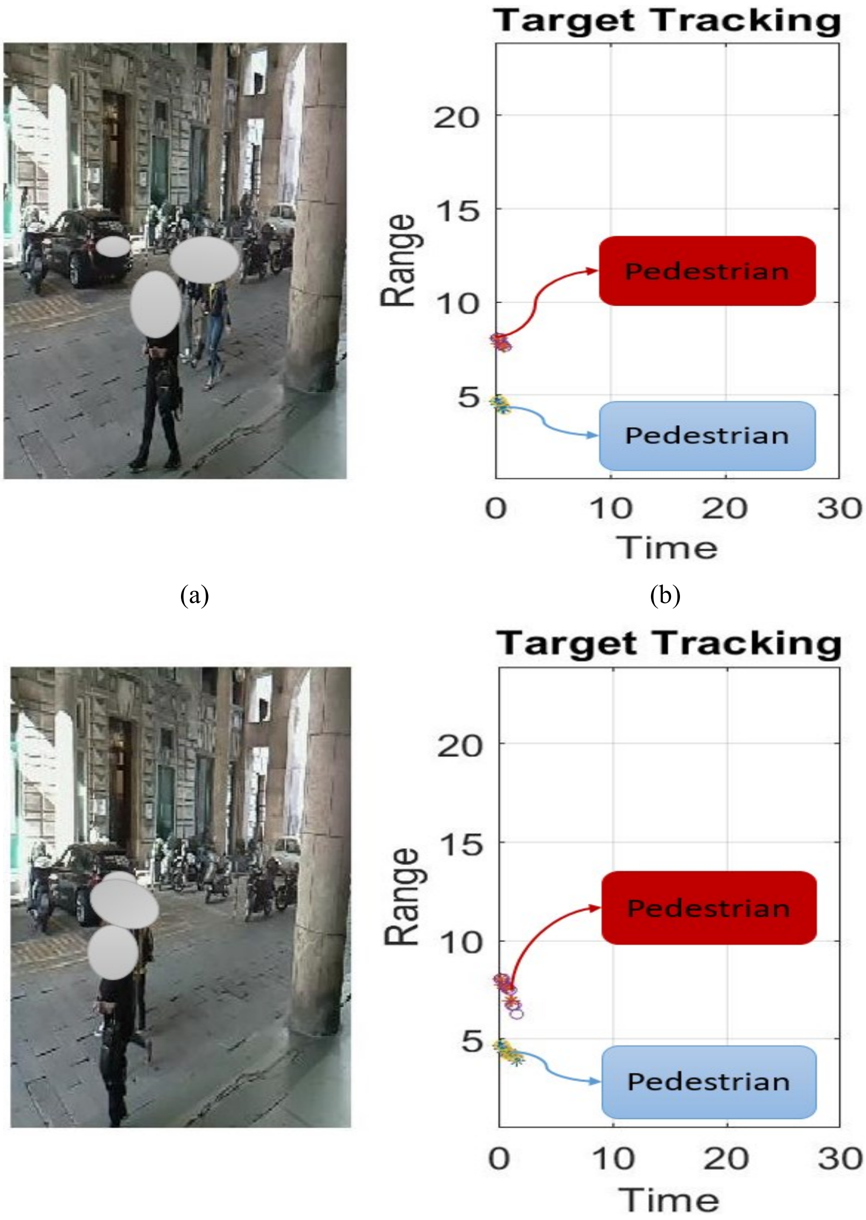


(a) (b)

(c) (d)

Figure 6.10 Camera images and tracking obtained from the second scenario (first test case). The circle symbol indicates the predicted path by the alpha-beta filter, while the star symbol indicates the measured path by the radar.

the modified parameters are shown in Table 6.18. As in the previous test, Figure 6.9 shows the performance of the proposed method on the whole dataset. Perfect, imperfect, and bad classification has the same meaning as presented before.



(a) (b)

(c) (d)

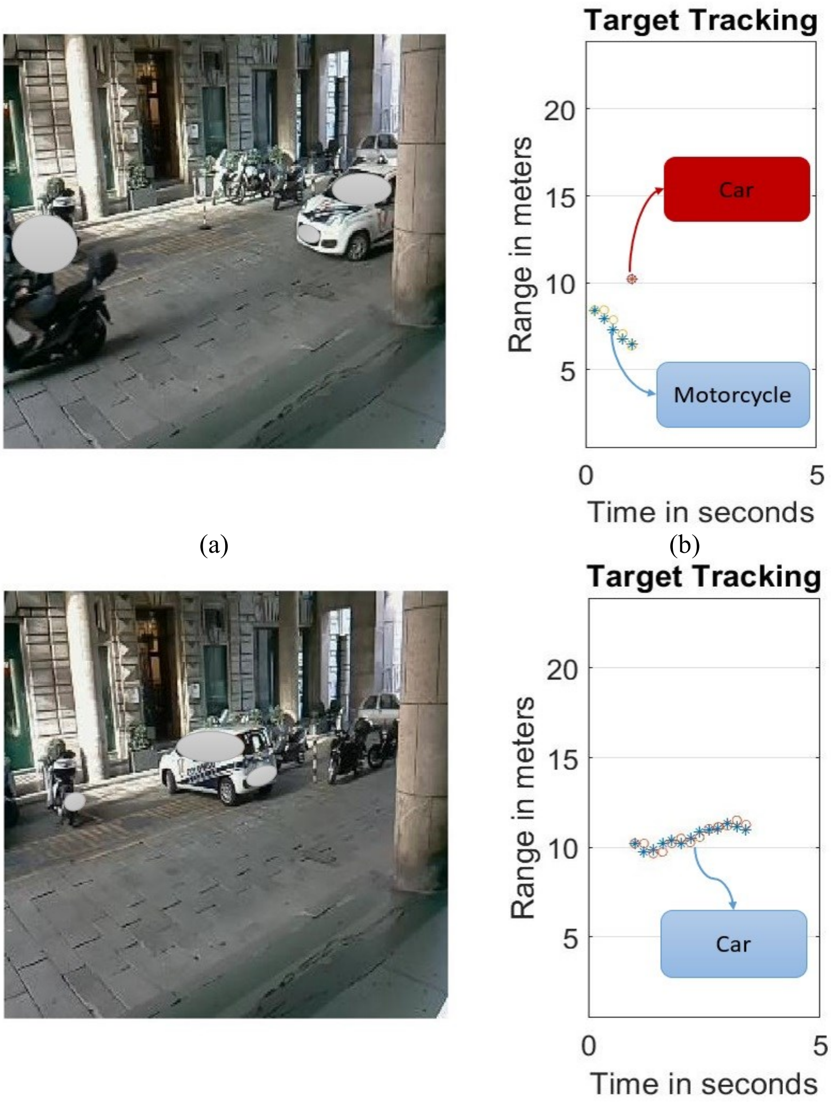
Figure 6.11 Camera images and tracking obtained from the second scenario (second test case). The circle symbol indicates the predicted path by the alpha-beta filter, while the star symbol indicates the measured path by the radar.

Also in this second scenario, the behaviour of the method has been analysed in detail for some significant cases. The first test case (Test 1) concerns the presence of two targets, of which one is a pedestrian departing and the other is an approaching car (Figure 6.10 (a)-(c)). From Figure 6.8 it can be clearly seen the narrow-angle of the radar in this new setup. However, the presence of the targets can be still clearly seen. The method was able to track the two targets perfectly and follow the trend even after the occlusion of the two targets (Figure 6.10 (b)-(d)). The aggregation losses and the classification results are shown in Table 6.19. The reason behind the lower difference margin of the aggregation loss compared with the previous tests is because the model used here is the previously trained model, built on a different scenario. However, the method still shows sufficient robustness to perform the classification.

In the subsequent test case (test 2), three pedestrians are approaching with two separate paths. Observing Figure 6.11(b)-(d), the pedestrians were tracked very well for some frames before they exit the scene. Looking at Figure 6.11(c)- (d), a similar case as in test 2 of the first scenario occurs, where the two pedestrians that were very close to each other were recognized and tracked as one target. Finally, the robustness of the classification is shown in Table 6.19. As in the first test case, a lower margin in the aggregation loss compared to the first scenario is present.

TABLE 6.19 RECOGNITION OF THE TYPE OF TARGET MADE ON THE STORED TRACKS, AGGREGATION LOSS AND ESTIMATED AVERAGE RADIAL SPEEDS. P = PEDESTRIAN, M = MOTORCYCLE, T = TRUCK, C = CAR.

Real class	***Predicted class***	***Aggregation loss***				***Average radial velocity* [km/h]**
		C	***P***	***T***	***M***	
TEST 1						
P	*P*	*0.7*	*0.14*	*0.52*	*0.36*	*1.94*
C	*C*	*0.18*	*0.37*	*0.34*	*0.4*	*-6.4*
TEST 2						
P	*P*	*0.5*	*0.22*	*0.31*	*0.28*	*-1.18*
P	*P*	*0.3*	*0.2*	*0.45*	*0.42*	*-2.22*
TEST 3						
M	*M*	*0.45*	*0.255*	*0.63*	*0.25*	*-5.16*
C	*C*	*0.1*	*0.71*	*0.4*	*0.82*	*2.1*



(a) (b)

(c) (d)

Figure 6.12 Camera images and tracking obtained from the second scenario (third test case). The circle symbol indicates the predicted path by the alpha-beta filter, while the star symbol indicates the measured path by the radar.

Finally, the last test case (Test 3) shown in Figure 6.12 concerns a motorcycle and a car approaching the radar (Figure 6.12 (a)). The two targets are successfully detected and tracked, as can be seen from Figure 6.12 (b). The motorcycle track was deleted after it left the scene for several frames, while the car was still tracked until it parked on the side of the road (Figure 6.12 (c)-(d)). The two targets were correctly classified by the system and the robustness of the classification is shown in. It is however worth noting that also in this second case the classifier margins are lower than in the first scenario. In particular, the motorcycle is quite close to being classified as a pedestrian. This behavior can be related to the diversity of the scenario used to train the classifier. Indeed, since the positioning of the radar has been changed, the range and Doppler features may be different. Nevertheless, the approach seems to be quite robust with respect to such changes. The aggregation losses and the classification results are shown in Table 6.19.

6.4 Summary:

This chapter presented the experimental phase that was followed to test the performance of the developed system. The performance of the system was firstly tested in human-vehicle classification, where two approaches were followed in that case, i.e., transfer learning and binary SVM. The SVM classifier showed a slightly better performance with an almost perfect classification result, while the result obtained with transfer learning was 96.5% classification accuracy. After that, the performance of the system was tested in the multi-class classification of the single targets passing in the FOV of the radar. the classes considered in this study are trucks, cars, pedestrians, and motorcycles. The OVA SVM was used in this test and the system showed a good performance in differentiating the different target classes with an overall accuracy of around 85% using a very small dataset for training and testing. Another multi-class test was carried out using the machine learning software Rulex. Multiple forecasts were cascaded in a tree-like structure while tuning the parameters of every forecast. This approach achieves higher accuracy over the classical approach of applying a single forecast for all classes. The last test was a multi-target validation of the system performance. The system was trained on the collected single target data and tested using the OVA SVM on two multi-target datasets. The system was firstly tested on the multi-target dataset collected in the first testing environment. Later, its performance was validated on a second multi-target dataset collected in a different environment with a different environmental condition. The obtained results were very good in both single and multi-target situations which proves the robustness of the proposed processing chain and the goodness of the classification.

Chapter 7: Conclusions and Future Works

7.1 Conclusions:

People’s safety, sometimes, can be at the heart of target classification systems. For example, in autonomous vehicles, such systems are used to ensure pedestrian's safety. Radars and vision sensors are the main competitors in this field of application. However, radar sensors are proved to be more robust in bad weather and bad lighting conditions. Among the different available radar technologies, FMCW radars are the most widely used for such applications. This is indeed for their abilities to provide simultaneous range and Doppler information. In addition, they require a lower sampling rate and lower peak to average ratio compared to pulse and UWB radar sensors.

In this study, a cost-efficient multi-target classification system for smart gate monitoring was proposed. The system adopts a novel short-range surveillance technique based on low-cost FMCW radar technology. The data processing method is based on an ad-hoc chain composed of several blocks, which perform clutter and leakage removal by means of a frame subtraction technique, clustering with the DBSCAN algorithm applied on the Range-Doppler maps, then a peak to cluster assignment step is introduced before tracking of targets with an α-β filter, features extraction, and finally classification using OVA SVM classifier.

A prototype of the system, equipped with a low-cost Distance2Go radar module from Infineon and a Raspberry PI 3 Model B+ mini-PC, has been assembled and used to experimentally test the developed procedure in real operating environments. The target data were collected by the system and processed using custom MATLAB software. In this study, four classes of targets were considered (trucks, cars, pedestrians, and motorcycles), and the classification is performed using features extracted from the range and Doppler spectra. The system was trained only on the single target data set, then the performance of the system was tested in two different scenarios with different conditions.

First, a human-vehicle classification was carried out following two approaches, i.e., transfer learning and binary SVM, which both yielded good classification results. After a multi-class classification test was done using OVA SVM to differentiate between the four classes of targets under study, i.e., cars, trucks, motorcycles, and pedestrians, and a good result with an overall accuracy of 85% was obtained.

The machine learning software Rulex ported to a Raspberry Pi in a client/server setup for edge computing applications has also been used to make forecasts on a pedestrian and vehicle classification dataset for urban security applications. Multiple forecasts were cascaded in a tree-like structure while tuning the parameters of every forecast. Classes were split into sub-classes and single process simulations were applied, where we estimated the overall accuracy for various cascaded setups. After exhausting all the possible arrangements, the setup with the best-projected output was simulated in a cascaded configuration, which provides an improved prediction outcome. This approach achieves higher accuracy over the classical approach of applying a single forecast for all classes. Further, combining classes into a parent class can be useful in practice, such is the case with the vehicle's parent class. However, this approach is exhaustive and time-consuming and may require setting parameters for different forecasts and various ML algorithms. Moreover, the tree-based method for improving machine learning forecasts can be used in various configurations. After applying the proposed method, humans were classified with an accuracy of 100% and vehicles with an accuracy of 96.67%. The final vehicle sub-classes forecast accuracies are 90.63% for motorcycles and 77.34% for the cars and trucks classes.

The final test was a multi-target classification test that was done on two different multi-target datasets that were collected in two different environments with different conditions. The obtained results show that the developed technique, although relying on data provided by low-cost components with limited radar resolutions and computational capabilities, can effectively recognize the targets, even in complex situations involving multiple simultaneous objects with different velocities and moving directions. In the first scenario, the system was tested on a multi-target dataset that was collected in the same environmental conditions as the training data and the system showed good accuracy in classifying the multiple targets. In the second scenario, no training has been done and the already trained system was tested on a new multi-target dataset that was collected in a different environmental condition. Also, in this case, the system showed a good performance in classifying the different passing targets.

7.2 Future Works:

As for future works, further developments will be aimed at including more advanced processing blocks, e.g., tracking schemes based on probabilistic models of associations. Moreover, the integration of enhanced radar boards, e.g., with the possibility of providing the positions of the targets through directions of arrival of the reflected signals or working at higher frequencies, will be also pursued to increase the classification capabilities especially in the presence of multiple targets with small distances. In addition, other classification techniques need to be tested, i.e., deep learning classification algorithms. In particular, Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Recurrent Neural Networks (RNN), which can deal directly with radar images and time sequence frames without the need for a feature extraction step. Such techniques also require much more training data, which indicates the need to build a much bigger dataset for future works.

References

[1] P. Withington, H. Fluhler, and S. Nag, “Enhancing homeland security with advanced UWB sensors,” *IEEE Microw. Mag.*, vol. 4, no. 3, pp. 51–58, Sep. 2003, doi: 10.1109/MMW.2003.1237477.

[2] F. Colone, D. Pastina, P. Falcone, and P. Lombardo, “WiFi-based passive ISAR for high-resolution cross-range profiling of moving targets,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3486–3501, Jun. 2014, doi: 10.1109/TGRS.2013.2273099.

[3] G. Gennarelli, G. Vivone, P. Braca, F. Soldovieri, and M. G. Amin, “Multiple extended target tracking for through-wall radars,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6482–6494, 2015, doi: 10.1109/TGRS.2015.2441957.

[4] A. Randazzo *et al.*, “A two-step inverse-scattering technique in variable-exponent Lebesgue spaces for through-the-wall microwave imaging: Experimental results,” *IEEE Trans. Geosci. Remote Sens.*, 2021, doi: 10.1109/TGRS.2021.3052608.

[5] B. Pottier, L. Rasolofondraibe, and S. Kerroumi, “Pedestrian detection strategy in urban area: capacitance probes and pedestrians’ signature,” *IEEE Sens. J.*, vol. 17, no. 17, pp. 5663–5668, Sep. 2017, doi: 10.1109/JSEN.2017.2718734.

[6] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, “Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications,” *IEEE Sens. J.*, vol. 20, no. 9, pp. 4901–4913, May 2020, doi: 10.1109/JSEN.2020.2966034.

[7] T. Gandhi and M. M. Trivedi, “Pedestrian protection systems: Issues, survey, and challenges,” *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 413–430, Sep. 2007, doi: 10.1109/TITS.2007.903444.

[8] A. Rasouli and J. K. Tsotsos, “Autonomous vehicles that interact with pedestrians: A survey of theory and practice,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 900–918, Mar. 2020, doi: 10.1109/TITS.2019.2901817.

[9] G. Gennarelli, I. Catapano, X. Dérobert, and F. Soldovieri, “A ground penetrating radar imaging approach for a heterogeneous subsoil with a vertical permittivity gradient,” *IEEE Trans. Geosci. Remote Sens.*, 2020, doi: 10.1109/TGRS.2020.3024831.

[10] A. Benedetto and L. Pajewski, *Civil Engineering Applications of Ground Penetrating Radar*. Cham, Switzerland: Springer, 2015.

[11] S. Lambot and F. André, “Full-wave modeling of near-field radar data for planar layered media reconstruction,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2295–2303, May 2014, doi: 10.1109/TGRS.2013.2259243.

[12] M. Pastorino *et al.*, “A microwave tomographic system for wood characterization in the forest products industry,” *Wood Mater. Sci. Eng.*, vol. 10, no. 1, pp. 75–85, 2015, doi: http://dx.doi.org/10.1080/17480272.2014.898696.

[13] T. Negishi, G. Gennarelli, F. Soldovieri, Y. Liu, and D. Erricolo, “Radio frequency tomography for nondestructive testing of pillars,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3916–3926, Jun. 2020, doi: 10.1109/TGRS.2019.2959589.

[14] J. M. Munoz-Ferreras, F. Perez-Martinez, J. Calvo-Gallego, A. Asensio-Lopez, B. P. Dorta-Naranjo, and A. Blanco-del-Campo, “Traffic surveillance system based on a high-resolution radar,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1624–1633, Jun. 2008, doi: 10.1109/TGRS.2008.916465.

[15] S. Saponara, M. S. Greco, and F. Gini, “Radar-on-chip/in-package in autonomous driving vehicles and intelligent transport systems: opportunities and challenges,” *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 71–84, Sep. 2019, doi: 10.1109/MSP.2019.2909074.

[16] S. Fontul, A. Paixão, M. Solla, and L. Pajewski, “Railway track condition assessment at network level by frequency domain analysis of GPR data,” *Remote Sens.*, vol. 10, no. 4, 2018, doi: 10.3390/rs10040559.

[17] R. Zhang and S. Cao, “Extending reliability of mmwave radar tracking and detection via fusion with camera,” *IEEE Access*, vol. 7, pp. 137065–137079, 2019, doi: 10.1109/ACCESS.2019.2942382.

[18] G. L. Charvat, *Small and Short-Range Radar Systems*. Boca Raton: CRC Press, Taylor & Francis Group, 2014.

[19] C. Li *et al.*, “A review on recent progress of portable short-range noncontact microwave radar systems,” *IEEE Trans. Microw. Theory Tech.*, vol. 65, no. 5, pp. 1692–1706, May 2017, doi: 10.1109/TMTT.2017.2650911.

[20] Pao-Jen Wang, Chi-Min Li, Cheng-Ying Wu, and Hsueh-Jyh Li, “A channel awareness vehicle detector,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 339–347, Jun. 2010, doi: 10.1109/TITS.2010.2043100.

[21] S. Saponara and B. Neri, “Radar sensor signal acquisition and multidimensional FFT processing for surveillance applications in transport systems,” *IEEE Trans. Instrum. Meas.*, vol. 66, no. 4, pp. 604–615, Apr. 2017, doi: 10.1109/TIM.2016.2640518.

[22] A. Mukhtar, L. Xia, and T. B. Tang, “Vehicle detection techniques for collision avoidance systems: A review,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2318–2338, Oct. 2015, doi: 10.1109/TITS.2015.2409109.

[23] D.-H. Jung, H.-S. Kang, C.-K. Kim, J. Park, and S.-O. Park, “Sparse scene recovery for high-resolution automobile FMCW SAR via scaled compressed sensing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10136–10146, Dec. 2019, doi: 10.1109/TGRS.2019.2931626.

[24] C. Ding *et al.*, “Continuous human motion recognition with a dynamic range-Doppler trajectory method based on FMCW radar,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6821–6831, Sep. 2019, doi: 10.1109/TGRS.2019.2908758.

[25] Y. Nan, X. Huang, and Y. J. Guo, “A millimeter-wave GCW-SAR based on deramp-on-receive and piecewise constant Doppler imaging,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 680–690, Jan. 2020, doi: 10.1109/TGRS.2019.2939004.

[26] C. Q. Mayoral *et al.*, “Water content continuous monitoring of grapevine xylem tissue using a portable low-power cost-effective FMCW radar,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5595–5605, Aug. 2019, doi: 10.1109/TGRS.2019.2900565.

[27] B. Vandersmissen *et al.*, "Indoor person identification using a low-power FMCW radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3941–3952, Jul. 2018, doi: 10.1109/TGRS.2018.2816812.

[28] B. Yektakhah and K. Sarabandi, "All-directions through-the-wall imaging using a small number of moving omnidirectional bi-static FMCW transceivers," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2618–2627, May 2019, doi: 10.1109/TGRS.2018.2875695.

[29] A. Ganis *et al.*, "A portable 3-D imaging FMCW MIMO radar demonstrator with a 24x24 antenna array for medium-range applications," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 298–312, Jan. 2018, doi: 10.1109/TGRS.2017.2746739.

[30] D. Jasteh, E. G. Hoare, M. Cherniakov, and M. Gashinova, "Experimental low-terahertz radar image analysis for automotive terrain sensing," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 4, pp. 490–494, Apr. 2016, doi: 10.1109/LGRS.2016.2518579.

[31] F. Fioranelli, S. Salous, and X. Raimundo, "Frequency-modulated interrupted continuous wave as wall removal technique in through-the-wall imaging," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6272–6283, Oct. 2014, doi: 10.1109/TGRS.2013.2295835.

[32] A. Anghel, G. Vasile, R. Cacoveanu, C. Ioana, and S. Ciochina, "Short-range wideband FMCW radar for millimetric displacement measurements," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5633–5642, Sep. 2014, doi: 10.1109/TGRS.2013.2291573.

[33] R. Wang, Yun-Hua Luo, Yun-Kai Deng, Zhi-Ming Zhang, and Yue Liu, "Motion compensation for high-resolution automobile FMCW SAR," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 5, pp. 1157–1161, Sep. 2013, doi: 10.1109/LGRS.2012.2234435.

[34] T. Lee, S. Jeon, J. Han, V. Skvortsov, K. Nikitin, and M. Ka, "A simplified technique for distance and velocity measurements of multiple moving objects using a linear frequency modulated signal," *IEEE Sens. J.*, vol. 16, no. 15, pp. 5912–5920, Aug. 2016, doi: 10.1109/JSEN.2016.2563458.

[35] T. Lee, V. Skvortsov, M. Kim, S. Han, and M. Ka, “Application of W-band FMCW radar for road curvature estimation in poor visibility conditions,” *IEEE Sens. J.*, vol. 18, no. 13, pp. 5300–5312, Jul. 2018, doi: 10.1109/JSEN.2018.2837875.

[36] M. Rameez, M. Dahl, and M. I. Pettersson, “Autoregressive model-based signal reconstruction for automotive radar interference mitigation,” *IEEE Sens. J.*, vol. 21, no. 5, pp. 6575–6586, Mar. 2021, doi: 10.1109/JSEN.2020.3042061.

[37] T. Yang, H. Shi, J. Guo, and M. Liu, “FMCW ISAR autofocus imaging algorithm for high-speed maneuvering targets based on image contrast-based autofocus and phase retrieval,” *IEEE Sens. J.*, vol. 20, no. 3, pp. 1259–1267, Feb. 2020, doi: 10.1109/JSEN.2019.2947559.

[38] M. Ash, M. Ritchie, and K. Chetty, “On the application of digital moving target indication techniques to short-range FMCW radar data,” *IEEE Sens. J.*, vol. 18, no. 10, pp. 4167–4175, May 2018, doi: 10.1109/JSEN.2018.2823588.

[39] F. Alimenti *et al.*, “Noncontact measurement of river surface velocity and discharge estimation with a low-cost Doppler radar sensor,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 5195–5207, Jul. 2020, doi: 10.1109/TGRS.2020.2974185.

[40] M. A. Richards, *Fundamentals of Radar Signal Processing*. New York, NY: McGraw-Hill, 2014.

[41] M. Kronauge and H. Rohling, “New chirp sequence radar waveform,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 4, pp. 2870–2877, Oct. 2014, doi: 10.1109/TAES.2014.120813.

[42] J.-J. Lin, Y.-P. Li, W.-C. Hsu, and T.-S. Lee, “Design of an FMCW radar baseband signal processing system for automotive application,” *SpringerPlus*, vol. 5, no. 1, p. 42, Dec. 2016, doi: 10.1186/s40064-015-1583-5.

[43] T. Mitomo, N. Ono, H. Hoshino, Y. Yoshihara, O. Watanabe, and I. Seto, “A 77 GHz 90 nm CMOS transceiver for FMCW radar applications,” *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 928–937, Apr. 2010, doi: 10.1109/JSSC.2010.2040234.

[44] A. Asensio Lopez *et al.*, “Coherent signal processing for traffic flow measuring radar sensor,” *IEEE Sens. J.*, vol. 18, no. 12, pp. 4803–4813, Jun. 2018, doi: 10.1109/JSEN.2017.2757699.

[45] Y. Zhao and Y. Su, “Vehicles detection in complex urban scenes using gaussian mixture model with FMCW radar,” *IEEE Sens. J.*, vol. 17, no. 18, pp. 5948–5953, Sep. 2017, doi: 10.1109/JSEN.2017.2733223.

[46] Y. Lee, H. Choo, S. Kim, and H. Kim, “RCS based target recognition with real FMCW radar implementation,” *Microw. Opt. Technol. Lett.*, vol. 58, no. 7, pp. 1745–1750, Jul. 2016, doi: 10.1002/mop.29901.

[47] H.-T. Chou, H.-K. Ho, and T.-P. Chang, “Effectiveness of moving signal-average method in K band FMCW radar for short-range vehicle detection using antennas with narrow beam widths,” *Radio Sci.*, vol. 53, no. 3, pp. 344–356, Mar. 2018, doi: 10.1002/2017RS006460.

[48] H.-J. Cho and M.-T. Tseng, “A support vector machine approach to CMOS-based radar signal processing for vehicle classification and speed estimation,” *Math. Comput. Model.*, vol. 58, no. 1–2, pp. 438–448, Jul. 2013, doi: 10.1016/j.mcm.2012.11.003.

[49] R. Perez, F. Schubert, R. Rasshofer, and E. Biebl, “Single-frame vulnerable road users classification with a 77 GHz FMCW radar sensor and a convolutional neural network,” in *Proceedings of the 19th International Radar Symposium*, Bonn, Germany, Jun. 2018, pp. 1–10. doi: 10.23919/IRS.2018.8448126.

[50] A. Mazzinghi, L. Facheris, F. Cuccoli, and A. Freni, “2.5D physical optics based algorithm for vehicles classification through a FM-CW radar,” in *Proceedings of the 15th European Radar Conference*, Madrid, Spain, Sep. 2018, pp. 257–260. doi: 10.23919/EuRAD.2018.8546575.

[51] T. Gao, Z. Lai, Z. Mei, and Q. Wu, “Hybrid SVM-CNN classification technique for moving targets in automotive FMCW radar system,” in *Proceedings of the 11th International Conference on Wireless Communications and Signal Processing*, Xi’an, China, Oct. 2019, pp. 1–6. doi: 10.1109/WCSP.2019.8928051.

[52] S. Heuel and H. Rohling, “Pedestrian recognition based on 24 GHz radar sensors,” in *Ultra-Wideband Radio Technologies for Communications, Localization and Sensor Applications*, R. Thom, Ed. InTech, 2013. doi: 10.5772/53007.

[53] S. Lee, Y.-J. Yoon, J.-E. Lee, and S.-C. Kim, “Human–vehicle classification using feature-based SVM in 77-GHz automotive FMCW radar,” *IET Radar Sonar Navig.*, vol. 11, no. 10, pp. 1589–1596, Oct. 2017, doi: 10.1049/iet-rsn.2017.0126.

[54] R. Prophet *et al.*, “Pedestrian classification with a 79 GHz automotive radar sensor,” in *Proceedings of the 19th International Radar Symposium*, Bonn, Germany, Jun. 2018, pp. 1–6. doi: 10.23919/IRS.2018.8448161.

[55] O. Schumann, J. Lombacher, M. Hahn, C. Wohler, and J. Dickmann, “Scene understanding with automotive radar,” *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 188–203, 2020, doi: 10.1109/TIV.2019.2955853.

[56] I. Urazghildiiev *et al.*, “Vehicle classification based on the radar measurement of height profiles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 2, pp. 245–253, Jun. 2007, doi: 10.1109/TITS.2006.890071.

[57] S.-L. Jeng, W.-H. Chieng, and H.-P. Lu, “Estimating speed using a side-looking single-radar vehicle detector,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 607–614, Apr. 2014, doi: 10.1109/TITS.2013.2283528.

[58] D. Felguera-Martin, J.-T. Gonzalez-Partida, P. Almorox-Gonzalez, and M. Burgos-García, “Vehicular traffic surveillance and road lane detection using radar interferometry,” *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 959–970, Mar. 2012, doi: 10.1109/TVT.2012.2186323.

[59] L. Du, L. Li, B. Wang, and J. Xiao, “Micro-Doppler feature extraction based on time-frequency spectrogram for ground moving targets classification with low-resolution radar,” *IEEE Sens. J.*, vol. 16, no. 10, pp. 3756–3763, May 2016, doi: 10.1109/JSEN.2016.2538790.

[60] A. Angelov, A. Robertson, R. Murray-Smith, and F. Fioranelli, “Practical classification of different moving targets using automotive radar and deep neural networks,” *IET Radar Sonar Navig.*, vol. 12, no. 10, pp. 1082–1089, Oct. 2018, doi: 10.1049/iet-rsn.2018.0103.

[61] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[62] S. Heuel and H. Rohling, “Pedestrian classification in automotive radar systems,” in *Proceedings of the 13th International Radar Symposium*, Warsaw, Poland, May 2012, pp. 39–44. doi: 10.1109/IRS.2012.6233285.

[63] M. Awad and R. Khanna, “Support Vector Machines for Classification,” in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, M. Awad and R. Khanna, Eds. Berkeley, CA: Apress, 2015, pp. 39–66. doi: 10.1007/978-1-4302-5990-9_3.

[64] S. Riegger and W. Wiesbeck, “Wide-band polarimetry and complex radar cross section signatures,” *Proc. IEEE*, vol. 77, no. 5, pp. 649–658, May 1989, doi: 10.1109/5.32055.

[65] S. M. Herman, “Joint passive radar tracking and target classification using radar cross section,” San Diego, CA, Dec. 2003, pp. 402–417. doi: 10.1117/12.505793.

[66] H. Jiang, L. Xu, and K. Zhan, “Joint tracking and classification based on aerodynamic model and radar cross section,” *Pattern Recognit.*, vol. 47, no. 9, pp. 3096–3105, Sep. 2014, doi: 10.1016/j.patcog.2014.03.001.

[67] S. Gupta, P. K. Rai, A. Kumar, P. K. Yalavarthy, and L. R. Cenkeramaddi, “Target Classification by mmWave FMCW Radars Using Machine Learning on Range-Angle Images,” *IEEE Sens. J.*, vol. 21, no. 18, pp. 19993–20001, Sep. 2021, doi: 10.1109/JSEN.2021.3092583.

[68] R. Khanna, D. Oh, and Y. Kim, “Through-wall remote human voice recognition using doppler radar with transfer learning,” *IEEE Sens. J.*, vol. 19, no. 12, pp. 4571–4576, Jun. 2019, doi: 10.1109/JSEN.2019.2901271.

[69] A. Bhattacharya and R. Vaughan, “Deep learning radar design for breathing and fall detection,” *IEEE Sens. J.*, vol. 20, no. 9, pp. 5072–5085, May 2020, doi: 10.1109/JSEN.2020.2967100.

[70] X. Huang, J. Ding, D. Liang, and L. Wen, “Multi-person recognition using separated micro-Doppler signatures,” *IEEE Sens. J.*, vol. 20, no. 12, pp. 6605–6611, Jun. 2020, doi: 10.1109/JSEN.2020.2977170.

[71] S. Kim, K. Lee, S. Doo, and B. Shim, “Automotive radar signal classification using bypass recurrent convolutional networks,” in *Proceedings of the IEEE/CIC International Conference on Communications in China*, Changchun, China, Aug. 2019, pp. 798–803. doi: 10.1109/ICCChina.2019.8855808.

[72] Y. Kim, I. Alnujaim, S. You, and B. J. Jeong, “Human Detection Based on Time-Varying Signature on Range-Doppler Diagram Using Deep Neural Networks,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 426–430, Mar. 2021, doi: 10.1109/LGRS.2020.2980320.

[73] M. E. Yanik and M. Torlak, “Near-Field 2-D SAR Imaging by Millimeter-Wave Radar for Concealed Item Detection,” in *2019 IEEE Radio and Wireless Symposium (RWS)*, Orlando, FL, USA, Jan. 2019, pp. 1–4. doi: 10.1109/RWS.2019.8714552.

[74] C. Özdemir, *Inverse Synthetic Aperture Radar Imaging with MATLAB Algorithms*. Hoboken, NJ: Wiley, 2012.

[75] V. C. Chen, Fayin Li, Shen-Shyang Ho, and H. Wechsler, “Micro-doppler effect in radar: phenomenon, model, and simulation study,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 2–21, Jan. 2006, doi: 10.1109/TAES.2006.1603402.

[76] V. C. Chen, *The micro-doppler effect in radar*. Boston: Artech House, 2011.

[77] S. Abdulatif, Q. Wei, F. Aziz, B. Kleiner, and U. Schneider, “Micro-Doppler based human-robot classification using ensemble and deep learning approaches,” in *Proceedings of the 2018 IEEE Radar Conference*, Oklahoma City, OK, Apr. 2018, pp. 1043–1048. doi: 10.1109/RADAR.2018.8378705.

[78] Jiajin Lei and Chao Lu, “Target classification based on micro-doppler signatures,” in *Proceedings of the 2005 IEEE International Radar Conference*, Arlington, VA, USA, 2005, pp. 179–183. doi: 10.1109/RADAR.2005.1435815.

[79] C. Clemente, J. J. Soraghan, A. Robinson, A. W. Miller, A. M. Kinghorn, and D. Greig, “Micro-Doppler based target classification using multi-feature integration,” in *IET Intelligent Signal Processing Conference 2013 (ISP 2013)*, London, UK, 2013, p. 2.1-2.1. doi: 10.1049/cp.2013.2042.

[80] Infineon Technologies, “Distance2Go – XENSIV^{TM} 24 GHz radar demo kit with BGT24MTR11 and XMC4200 32-bit ARM® Cortex^{TM}-M4 MCU for ranging, movement and presence detection,” Application Note AN543, 2020. [Online]. Available: https://www.infineon.com/cms/en/product/evaluation-boards/demo-distance2go/

[81] J. Wang, Y. Lu, L. Gu, C. Zhou, and X. Chai, “Moving object recognition under simulated prosthetic vision using background-subtraction-based image processing strategies,” *Inf. Sci.*, vol. 277, pp. 512–524, Sep. 2014, doi: 10.1016/j.ins.2014.02.136.

[82] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, Aug. 1996, pp. 226–231.

[83] B. R. Mahafza and A. Z. Elsherbeni, *MATLAB Simulations for Radar Systems Design*. Boca Raton, FL: CRC Press/Chapman & Hall, 2004.

[84] P. R. Kalata and K. M. Murphy, “α-β target tracking with track rate variations,” in *Proceedings of the Twenty-Ninth Southeastern Symposium on System Theory*, Cookeville, TN, USA, 1997, pp. 70–74. doi: 10.1109/SSST.1997.581581.

[85] M. I. Skolnik, *Introduction to Radar Systems*. McGraw-Hill, 2001.

[86] J. Mohinder, *Design of Multi-Frequency CW Radars*. Institution of Engineering and Technology, 2007. doi: 10.1049/SBRA004E.

[87] M. Ash, M. Ritchie, and K. Chetty, “On the application of digital moving target indication techniques to short-range FMCW radar data,” *IEEE Sens. J.*, vol. 18, no. 10, pp. 4167–4175, May 2018, doi: 10.1109/JSEN.2018.2823588.

[88] M. Jankiraman, *Design of Multi-Frequency CW Radars*. Institution of Engineering and Technology, 2007. doi: 10.1049/SBRA004E.

[89] A. G. Stove, “Linear FMCW radar techniques,” *IEE Proc. F Radar Signal Process.*, vol. 139, no. 5, p. 343, 1992, doi: 10.1049/ip-f-2.1992.0048.

[90] E. Hyun, Y.-S. Jin, and J.-H. Lee, “A Pedestrian Detection Scheme Using a Coherent Phase Difference Method Based on 2D Range-Doppler FMCW Radar,” *Sensors*, vol. 16, no. 1, p. 124, Jan. 2016, doi: 10.3390/s16010124.

[91] B. R. Mahafza, *Radar systems analysis and design using Matlab*. Boca Raton: Chapman & Hall/CRC, 2000.

[92] J. Munkres, “Algorithms for the assignment and transportation problems,” *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, Mar. 1957, doi: 10.1137/0105003.

[93] M. Drahansky *et al.*, “We are IntechOpen , the world ’ s leading publisher of Open Access books Built by scientists , for scientists TOP 1 %,” *Intech*, vol. i, no. tourism, p. 13, 2016, doi: http://dx.doi.org/10.5772/57353.

[94] S. Heuel and H. Rohling, “Pedestrian classification in automotive radar systems,” in *2012 13th International Radar Symposium*, Warsaw, Poland, May 2012, pp. 39–44. doi: 10.1109/IRS.2012.6233285.

[95] S. Liaqat, S. A. Khan, M. B. Ihsan, S. Z. Asghar, A. Ejaz, and A. I. Bhatti, “Automatic recognition of ground radar targets based on target RCS and short time spectrum variance,” in *Proceedings of the 2011 International Symposium on Innovations in Intelligent Systems and Applications*, Istanbul, Turkey, Jun. 2011, pp. 164–167. doi: 10.1109/INISTA.2011.5946127.

[96] S. Lee, S. Kang, S.-C. Kim, and J.-E. Lee, “Radar cross section measurement with 77 GHz automotive FMCW radar,” in *Proceedings of the 27th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications*, Valencia, Spain, Sep. 2016, pp. 1–6. doi: 10.1109/PIMRC.2016.7794738.

[97] European Commission. Joint Research Centre. Institute for the Protection and the Security of the Citizen., *Radar cross section measurements of pedestrian dummies and humans*

in the 24/77 GHz frequency bands: establishment of a reference library of RCS signatures of pedestrian dummies in the automotive radar bands. LU: Publications Office, 2013. Accessed: Sep. 11, 2021. [Online]. Available: https://data.europa.eu/doi/10.2788/80844

[98] E. Schubert, M. Kunert, W. Menzel, J. Fortuny-Guasch, and J.-M. Chareau, "Human RCS measurements and dummy requirements for the assessment of radar based active pedestrian safety systems," in *2013 14th International Radar Symposium (IRS)*, 2013, vol. 2, pp. 752–757.

[99] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognit.*, vol. 44, no. 8, pp. 1761–1776, Aug. 2011, doi: 10.1016/j.patcog.2011.01.017.

[100] "ImageNet." https://image-net.org/ (accessed Oct. 05, 2021).

[101] M. Alameh, A. Ibrahim, M. Valle, and G. Moser, "DCNN for Tactile Sensory Data Classification based on Transfer Learning," in *2019 15th Conference on Ph.D Research in Microelectronics and Electronics (PRIME)*, Lausanne, Switzerland, Jul. 2019, pp. 237–240. doi: 10.1109/PRIME.2019.8787748.

[102] "Sistemi integrati NVIDIA per macchine autonome di nuova generazione," *NVIDIA*. https://www.nvidia.com/it-it/autonomous-machines/embedded-systems/ (accessed Oct. 05, 2021).

[103] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Proc Thirty-First AAAI Conf. Artif. Intell.*, p. 7, 2017.

[104] P. K. Dash, S. R. Samantaray, and G. Panda, "Fault Classification and Section Identification of an Advanced Series-Compensated Transmission Line Using Support Vector Machine," *IEEE Trans. Power Deliv.*, vol. 22, no. 1, pp. 67–73, Jan. 2007, doi: 10.1109/TPWRD.2006.876695.

[105] Yi Liu and Y. F. Zheng, “One-against-all multi-class SVM classification using reliability measures,” in *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, Montreal, Canada, 2005, pp. 849–854. doi: 10.1109/IJCNN.2005.1555963.

[106] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian optimization of machine learning algorithms,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Red Hook, NY, Dec. 2012, pp. 2951–2959.

[107] A. Rojas-Dominguez, L. C. Padierna, J. M. Carpio Valadez, H. J. Puga-Soberanes, and H. J. Fraire, “Optimal Hyper-Parameter Tuning of SVM Classifiers With Application to Medical Diagnosis,” *IEEE Access*, vol. 6, pp. 7164–7176, 2018, doi: 10.1109/ACCESS.2017.2779794.

[108] E. Tuba, M. Tuba, and D. Simian, “Adjusted bat algorithm for tuning of support vector machine parameters,” in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation*, Vancouver, Canada, Jul. 2016, pp. 2225–2232. doi: 10.1109/CEC.2016.7744063.

[109] M. Muselli, “Extracting knowledge from biomedical data through Logic Learning Machines and Rulex,” *EMBnet.journal*, vol. 18, no. B, p. 56, Nov. 2012, doi: 10.14806/ej.18.B.549.

[110] M. Muselli, “Switching Neural Networks: A New Connectionist Model for Classification,” in *Neural Nets*, vol. 3931, B. Apolloni, M. Marinaro, G. Nicosia, and R. Tagliaferri, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 23–30. doi: 10.1007/11731177_4.

[111] M. Muselli and E. Ferrari, “Coupling Logical Analysis of Data and Shadow Clustering for Partially Defined Positive Boolean Function Reconstruction,” *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 37–50, Jan. 2011, doi: 10.1109/TKDE.2009.206.

[112] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Internal Representations by Error Propagation,” *No ICS-8506 Univ. Calif. San Diego Jolla Dep. Cogn. Sci. Fort Belvo CA USA*, p. 23, 1985, doi: 10.1016/B978-1-4832-1446-7.50035-2.

[113] K. Hajdarevic, S. Konjicija, and A. Subasi, “A low energy APRS-IS client-server infrastructure implementation using Raspberry Pi,” in *2014 22nd Telecommunications Forum Telfor (TELFOR)*, Belgrade, Serbia, Nov. 2014, pp. 296–299. doi: 10.1109/TELFOR.2014.7034409.