Unified Multi-Object Tracking and Sensing with mmWave Radar

by **Andre Pearce**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of

Prof. J. Andrew Zhang

A/Prof. Richard Yida Xu

School of Electrical and Data Engineering
Faculty of Engineering and IT
University of Technology Sydney

July 13, 2025

Certificate of Original Authorship

I, Andre Pearce, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: July 13, 2025

Abstract

Millimetre-wave (mmWave) radars have increasingly become more popular for short-range object tracking due to their high resolution and robustness in various environmental conditions. Although the literature on single object tracking with mmWave radar is well-established, multi-object tracking remains a developing field. The current leading implementations for mmWave multi-object tracking typically operate under assumptions that can either limit their adaptability, performance, or accuracy. The research presented in this thesis aims to enhance the capability and reliability of mmWave multi-object tracking systems, by unifying traditional tracking methodologies with advanced mmWave radar sensing.

To ultimately achieve this, the thesis addresses several key challenges. Firstly, the feasibility of improving mmWave multi-object tracking performance through environmental sensing is explored. A novel tracking algorithm is developed to leverage the high-resolution of the mmWave radar to define regional trajectory analysis patterns, improving object detection and tracking performance.

Secondly, an approach towards reducing the challenges associated with labelling mmWave radar data is presented, fundamentally achieved through a sensor fusion architecture. This proposed approach dramatically improves the accessibility and feasibility of constructing large scale datasets that can be used to train accurate mmWave radar deep learning systems.

Thirdly, the research proposes a generalised framework for a joint mmWave radar sensing and tracking system, incorporating our novel mmWave Convolutional LSTM Autoencoder (mmCLAE) for rain-induced noise reduction. This framework is designed to be inherently adaptable to various tracking scenarios and environmental conditions. The unified framework is validated through an implementation that estimates rainfall with mmWave radar, while jointly incorporating the impact of the rainfall and the noise reduction capabilities of mmCLAE to improve the overall tracking performance.

The results of the research conducted demonstrate a promising approach to yield significant improvements in tracking performance, adaptability, and robustness, compared to existing traditional multi-object tracking architectures. The research showcases mmWave multi-object tracking systems that can accurately track multiple objects, even in challenging conditions where objects intermittently leave the radar's field of view or the radar data frames are highly congested with complex noise profiles. The findings of this research have significant implications for a variety of differing applications, including autonomous vehicles, robotics and surveillance systems. This thesis contributes to the advancement of mmWave radar technology by providing a comprehensive study on enhancing the capabilities of mmWave radar systems for multi-object tracking. The unified tracking and sensing system proposed offers a foundation for future research towards more advanced and reliable tracking systems.

Dedicated to my Wife and Beloved Family

Acknowledgements

First and foremost, I would like to express my profound gratitude to my primary supervisor, Prof. J. Andrew Zhang, for his invaluable guidance, support, and encouragement throughout my PhD journey. His expertise and insights have been instrumental in shaping this thesis. His patience and willingness to share his vast knowledge have greatly contributed to my academic and personal growth. I am deeply appreciative of the countless hours he has dedicated towards me and providing constructive feedback.

I am also grateful to my secondary supervisor, A/Prof. Richard Yida Xu, for his constructive feedback and support. His contributions have significantly enriched my research.

I wish to extend my deepest gratitude to my parents and my brother for the unwavering support they have provided throughout my studies. Their constant encouragement and belief in my potential have been a source of strength and inspiration. The mentorship and reassurance they offered have been pivotal in helping me navigate through my PhD. Their faith in my abilities has been a beacon of hope, and for this, I thank them sincerely.

Lastly, I owe my deepest appreciation to my wife, Melissa Pearce. Her unwavering patience, immense sacrifices, and continuous belief in me have been the foundation of my success. She has been my pillar of strength and my greatest supporter. Her boundless love and encouragement have given me the strength to pursue my aspirations and overcome the numerous challenges encountered along the way. This thesis stands as a testament to her dedication and support, for which I am eternally grateful.

List of Publications

Journal Papers

- **Pearce, Andre** & Zhang, J. Andrew & Xu, Richard. (2025). Joint mmWave Rainfall and Object tracking with Rain Noise Reduction and Rain Intensity Classification using Deep Learning. *IEEE Internet of Things*, [Pending Submission].

- **Pearce, Andre** & Zhang, J. Andrew & Xu, Richard & Wu, Kai. (2023). Multi-Object Tracking with mmWave Radar: A Review. *Electronics*, 12(2), 308, doi: 10.3390/electronics12020308.

- **Pearce, Andre** & Zhang, J. Andrew & Xu, Richard. (2022). A Combined mmWave Tracking and Classification Framework Using a Camera for Labeling and Supervised Learning. *Sensors*, 22, 8859, doi: 10.3390/s22228859.

Conference Papers

- **Pearce, Andre** & Zhang, J. Andrew & Xu, Richard. (2022). Regional Trajectory Analysis through Multi-Person Tracking with mmWave Radar. *2022 IEEE Radar Conference (RadarConf22)*, 1-6, doi: 10.1109/RadarConf2248738.2022.9764343.

- Liu, Jingwei & Zhang, J. Andrew & Xu, Richard & **Pearce, Andre** & Ni, Wei & Hedley, Mark. (2020). Gaussian Mixture Model based Convolutional Sparse Coding for Radar Heartbeat Detection. *2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 1-6, doi:

10.1109/ICSPCS50536.2020.9310063.

- Shi, Zhenguo & Zhang, J. Andrew & Xu, Richard & Cheng, Qingqing & **Pearce, Andre**. (2020). Towards Environment-Independent Human Activity Recognition using Deep Learning and Enhanced CSI. *2020 IEEE Global Communications Conference*, 1-6, doi: 10.1109/GLOBECOM42002.2020.9322627.

Contents

Abstract ii

Acknowledgements v

List of Publications vi

List of Figures xiii

List of Tables xiv

Abbreviations xvii

1 Introduction 1

1.1 Background . . . 2

1.2 Problem Statement . . . 3

1.3 Major Contributions . . . 4

1.4 Thesis Structure . . . 7

2 Literature Review 10

2.1 History of Multi-Object Tracking . . . 10

2.1.1 Foundations of Multi-Object Tracking . . . 11

2.1.2 Early Developments in Multi-Object Tracking . . . 11

2.1.3 Recent Advances in Multi-Object Tracking . . . 16

2.2 mmWave Radar Multi-Object Tracking . . . 17

2.2.1 Discontinuous Multi-Object Tracking . . . 17

2.2.2 Applications of mmWave Multi-Object Tracking and Sensing . 18

2.3 Typical mmWave Tracking System Architecture . . . 19

2.3.1 Radar Architecture 20

2.3.2 Position and Velocity Estimation 22

2.3.3 Association and Tracking 25

2.3.4 Sensing and Identification 28

2.4 Advanced Technologies and Methodologies 29

2.4.1 Object Detection Enhancements 31

2.4.2 Sensing Methodologies 33

2.4.3 Identification Strategies 39

2.5 Review Summary 44

3 Multi-Object Regional Trajectory Analysis 47

3.1 Introduction 47

3.2 RDTP Multi-Object Tracking Architecture 49

3.2.1 Problem Statement 50

3.2.2 Proposed Framework 50

3.3 Methodology and Implementation 52

3.3.1 Multi-Object Tracking Trajectory Collection 52

3.3.2 Pre-processing and Normalisation 55

3.3.3 RDTP Analysis 57

3.3.4 Environmental Association and Bounding 59

3.4 Experimental Results and Analysis 60

3.5 Conclusion 62

4 Combined mmWave Tracking and Classification Framework Using Camera for Labelling and Supervised Learning 64

4.1 Introduction 65

4.2 Sensor Fusion Architectures 70

4.3 Radar Training with Camera Labelling and Supervision Methodology 75

4.3.1 Problem Space 75

4.3.2 Proposed Approach 77

4.4 System Design and Implementation 79

4.4.1 Radar Pipeline 81

4.4.2 Camera Pipeline 83

4.4.3 Fused Pipeline 90
4.5 Results 91
4.6 Conclusion 97

5 Joint mmWave Sensing and Tracking with mmCLAE 99
5.1 Introduction 100
5.2 Classical Methodologies 101
5.2.1 Noise Reduction 101
5.2.2 Rainfall Sensing 103
5.3 Unified System Architecture 104
5.3.1 Integration and Workflow 104
5.3.2 Advantages of the Unified System 105
5.4 Proposed Rain-induced Noise Reduction: mmCLAE 106
5.4.1 mmCLAE Architecture 106
5.4.2 mmCLAE Training Process 109
5.5 Proposed Sensing Method: CNN Rainfall Intensity Classification . . . 112
5.5.1 Rainfall Classification Architecture 113
5.5.2 Rainfall Classification Training Process 116
5.6 Experimental Results 117
5.6.1 Multi-Object Tracking Performance Comparison 119
5.6.2 Evaluation Metrics 122
5.6.3 Rain Intensity Classification Evaluation 124
5.7 Conclusion 125

6 Conclusions and Future Work 127
6.1 Summary of Contributions 127
6.2 Recommended Future Work 129
6.2.1 Advanced Environmental Characterisation 129
6.2.2 Improved Sensor Fusion Techniques 129
6.2.3 Enhanced Noise Reduction Methods 131

Bibliography 132

List of Figures

1.1 Thesis organisation. 7

2.1 Illustration of the FINST model for human visual field multi-object tracking. 12

2.2 Illustration of discontinuous tracking in mmWave radar. 18

2.3 mmWave tracking architecture block diagram. 19

2.4 Single board mmWave radar sensor architecture. 21

2.5 Illustration of a FMCW chirp signal showing the linear frequency sweep over time. 23

2.6 Generalised stages of association and tracking in an mmWave tracking architecture system. 25

2.7 Areas explored and discussed in Section 2.4 in contrast to the typical multi-object mmWave tracking architecture block diagram presented in Figure 2.3. 30

2.8 Walking classification system designs explored in [60]; **a)** PCA combined with SVM classification; **b)** PCA combined with k-NN classification; **c)** t-SNE combined with SVM classification; **d)** t-SNE combined with k-NN classification. 36

2.9 System breakdown of the gait identification methodology presented in [66]. 41

3.1 Block diagram of overall stages involved in the RDTP multi-object tracking architecture. 50

3.2 TI mmWave IWR6843ISK with MMWAVEICBOOST and DCA1000EVM. 53

3.3 Positioning of the mmWave sensor for RDTP. 53

3.4 Trajectory taxonomy. 58

3.5 Training and test error rate percentage across epochs. 61

3.6 RDTP multi-object tracking plot with projections and scene. 62

3.7 Precision, recall, and F1-score for entry and exit classification. 63

4.1 Radar training with camera labelling and supervision methodology processing chain. 78

4.2 Radar and camera time alignment bias. 79

4.3 Radar tracking system design with a human movement pattern classifier trained with camera labelled frames. 80

4.4 Radar processing pipeline design. 81

4.5 Generated radar range-Doppler heatmap. 81

4.6 Camera processing pipeline design. 83

4.7 Faster R-CNN model design, used for camera object detection. 85

4.8 Camera trained radar system accuracy in contrast to a trained standalone camera system. 94

4.9 Manually labelled trained radar system accuracy in contrast to the camera trained radar system. 96

5.1 Unified system overview architecture for joint mmWave tracking and sensing with mmCLAE. 104

5.2 Architecture of mmCLAE for rain-induced noise reduction in mmWave radar. 107

5.3 ConvLSTM cell architecture. 108

5.4 Architecture of the rainfall simulator for generating mmWave radar data with varying levels of rain-induced noise. 112

5.5 Physical setup of the rainfall simulator for generating mmWave radar data with varying levels of rain-induced noise. 113

5.6 Architecture of the rainfall intensity classification model. 114

5.7 Training pipeline for rainfall intensity classification. 116

5.8 Range-Doppler heatmap illustrating no rainfall. 119

5.9 Range-Doppler heatmap illustrating light rainfall. 120

5.10 Range-Doppler heatmap illustrating medium rainfall. 120

5.11 Range-Doppler heatmap illustrating medium rainfall with a single person walking. 121

5.12 Range-Doppler heatmap illustrating medium rainfall with two people walking. 122

5.13 Multi-object tracking mmCLAE and standalone EKF comparison of a simple movement path. 123

5.14 Accuracy graph of the rainfall intensity classification model. 125

5.15 Loss graph of the rainfall intensity classification model. 126

6.1 Mind map illustrating potential future research opportunities in mmWave radar multi-object tracking and sensing. 130

List of Tables

2.1 A comparison of methodologies explored for the enhancement of object detection in an mmWave tracking architecture. 34

2.2 A comparison of sensing methodologies explored for the enhancement of tracking reliability in an mmWave tracking architecture. 40

2.3 A comparison of identification methodologies explored for the enhancement of tracking objects discontinuously in an mmWave tracking architecture. 45

4.1 Types of mmWave radar and camera fusion system architectures. . . 66

4.2 Benefits and limitations of fusion architectures. 67

4.3 Types of mmWave radar and camera fusion depths. 68

4.4 Distribution of activities in the recorded dataset. 93

4.5 Accuracy similarity between the standalone camera system and camera trained radar system. 95

5.1 Rainfall classification types. 115

5.2 mmCLAE multi-object tracking performance comparison. 121

5.3 Rain intensity classification evaluation metrics. 124

Abbreviations

ADC Analog-to-Digital Converter.

AKF Adaptive Kalman Filter.

ANN Artificial Neural Network.

AoA Angle of Arrival.

CFAR Constant False Alarm Rate.

CNN Convolutional Neural Network.

ConvLSTM Convolutional Long Short-Term Memory.

CRF-Net CameraRadarFusionNet.

DBSCAN Density-based Spatial Clustering of Applications with Noise.

DSP Digital Signal Processor.

EKF Extended Kalman Filter.

EnKF Ensemble Kalman Filter.

Faster R-CNN Faster Region-based Convolutional Neural Network.

FC Fully Connected.

FFT Fast Fourier Transform.

FINST Fingers of Instantiation.

FMCW Frequency Modulated Continuous Wave.

GAN Generative Adversarial Network.

GCN Graph Convolutional Network.

HAR Human Activity Recognition.

IF Intermediate Frequency.

IIR Infinite Impulse Response.

IMU Inertial Measurement Unit.

IPM Inverse Perspective Mapping.

IR-UWB Impulse Radio Ultra-Wideband.

JPDA Joint Probabilistic Data Association.

k-NN K-Nearest Neighbour.

LiDAR Light Detection and Ranging.

LMS Least Mean Squares.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

MCU Microcontroller Unit.

MIMO Multiple Input Multiple Output.

MLP Multi-Layer Perceptron.

mmCLAE Millimetre Wave Convolutional Long Short-Term Memory Autoencoder.

mmWave Millimetre Wave.

MSE Mean Squared Error.

NMS Non-Maximum Suppression.

PCA Principal Component Analysis.

RDTP Regional Dominant Trajectory Pattern.

ReLU Rectified Linear Unit.

RFID Radio Frequency Identification.

RIS Reconfigurable Intelligent Surface.

RLS Recursive Least Squares.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

ROI Region of Interest.

Rx Receiver.

SNR Signal-to-Noise Ratio.

SVM Support Vector Machine.

t-SNE t-Distributed Stochastic Neighbour Embedding.

TI Texas Instruments.

Tx Transmitter.

UCB Upper Confidence Bound.

UKF Unscented Kalman Filter.

VAE Variational Autoencoder.

WCE Weighted Cross-Entropy.

Chapter 1

Introduction

Millimetre Wave (mmWave) radars have been a topic of great interest over recent years in the field of multi-object tracking and sensing. The potential and motivation for mmWave radars in this field is primarily driven by the micro-Doppler information that can be extrapolated. Micro-Doppler simply refers to the Doppler information generated by movements of individual parts of a particular target [1]. The micro-Doppler features can then be exploited to determine characteristics of multiple targets for tracking and sensing purposes. The characteristics extracted can ultimately translate to sub-millimeter individual movements of the targets, due to the heightened sensitivity of the radar caused by the short wavelength of mmWave.

In the context of this thesis, the term tracking refers to the ability to perform object detection on multiple targets, as well as maintaining a correlation between the targets currently detected status and previous detections. Sensing in the context of this thesis refers to the ability to extrapolate characteristics of multiple targets for classification and quantification purposes. This ultimately means extracting target information such as, but not limited to, behavioural patterns, patterns regarding movement across the field of view, and object signatures that could ultimately equate to profiling an object [2].

1.1 Background

mmWave refers to the electromagnetic spectrum with wavelengths between 1 millimetre and 10 millimetres, corresponding to frequencies between 30 GHz and 300 GHz. This frequency range is particularly attractive for various applications due to its ability to provide high-resolution data and its relatively short wavelength, which allows for the detection of fine details [3]. The use of mmWave technology has grown rapidly in recent years, driven by advancements in semiconductor technology and the increasing demand for high-bandwidth communication systems. The unique properties of mmWave signals, such as their ability to penetrate materials like clothing and their sensitivity to small movements, make them ideal for a wide range of sensing and tracking applications.

One of the primary applications of mmWave technology is in the field of automotive radar systems. These systems utilise mmWave sensors to detect and track objects around a vehicle, providing critical information for advanced driver assistance systems and autonomous driving [4]. The high resolution of mmWave radar allows for the detection of small objects and the precise measurement of their speed and distance, enabling features such as adaptive cruise control, collision avoidance, and lane change assistance. Additionally, while mmWave radar is still affected by adverse weather conditions, it is less impacted by environmental factors such as rain, fog, and dust compared to other sensing modalities like optical cameras and Light Detection and Ranging (LiDAR), making it a reliable choice for automotive applications [5].

Another significant application of mmWave technology is in the area of indoor and outdoor surveillance. mmWave sensors can be used to monitor the movement of people and objects in various environments, such as airports, shopping malls, and public transportation systems. The ability to detect micro-Doppler signatures, which are the unique patterns of motion generated by different parts of a moving object, allows for the identification and classification of different types of targets. This capability is particularly useful for security and surveillance applications, where it is important to distinguish between different types of objects, track them, and precisely extract interesting surveillance information [6]–[9].

In the healthcare sector, mmWave technology is being explored for applications such as remote patient monitoring and fall detection [10]–[15]. The high sensitivity of mmWave sensors to small movements makes them suitable for monitoring vital signs such as respiration and heart rate without the need for direct contact with the patient. This non-invasive approach is particularly beneficial for elderly patients or those with chronic conditions, as it allows for continuous monitoring without causing discomfort. Additionally, mmWave sensors can be used to detect falls and other sudden movements, providing timely alerts to caregivers and improving the overall safety of patients.

The use of mmWave technology is also expanding into the field of industrial automation and robotics. mmWave sensors can be integrated into robotic systems to enhance their perception capabilities, enabling them to navigate complex environments and perform tasks with greater precision [16], [17]. For example, mmWave radar can be used to detect and track objects in a manufacturing facility, allowing robots to avoid collisions and optimise their movements. The ability to operate in harsh environments and under challenging conditions makes mmWave sensors a valuable tool for improving the efficiency and safety of industrial processes.

The unique properties of mmWave signals, specifically their high resolution and sensitivity to small movements, make them ideal for tracking and sensing applications. As the technology continues to advance, it is expected that the use of mmWave sensors will become increasingly prevalent, driving further innovation and development in these areas.

1.2 Problem Statement

The research and solutions surrounding multi-object tracking with mmWave are not as mature as single object tracking. This is primarily due to the complexities that are involved in the identification of multiple targets, and the ability to persist a correlated track for each of the individual targets. The current state-of-art mmWave multi-object tracking systems only focus on continuous tracking, and typically through conventional signal processing techniques. Continuous multi-object tracking in this context refers to the tracking and association of multiple targets only whilst the target is within the field of view of the radar. Discontinuous multi-object tracking, on the other hand, is one in which the targets can ultimately be uniquely identified, associated, and tracked despite whether they interrupt their presence in the field of view of the radar. Discontinuous multi-object tracking has a much broader range of applications in comparison to continuous tracking. Specifically, discontinuous multi-object tracking systems would be applicable in extracting personalised movement characteristics amongst groups of people. In a generalised sense, a system of this nature can produce applications capable of providing a feedback mechanism to dynamically tailor an experience for a group, or individual.

In order to accomplish discontinuous multi-object tracking, unique mmWave features corresponding to targets need to be reliably correlated with the respective target movement. Inherently, there is a gap in current literature surrounding combined mmWave sensing and mmWave multi-object tracking systems. As a result, the information present in the scene that can be extrapolated from sensing methodologies is not being considered in typical mmWave multi-object tracking architectures. In order to accomplish a combined mmWave sensing and mmWave multi-object system, challenges surrounding the unification of the data and projection onto a single plane will need to be addressed. The research presented in this thesis aims to propose an approach that provides the theoretical and practical foundations for a unified mmWave multi-object tracking and sensing system, with a primary goal of accomplishing discontinuous multi-object tracking. The proposed work has a key focus on ensuring the reliant features are generalised to the extent in which environmental factors result in minimal impact to the performance and accuracy.

1.3 Major Contributions

The primary contributions of this thesis aim to address the problem statement discussed in the previous section, Section 1.2. One of the main challenges is the accurate and reliable tracking of multiple objects in dynamic and complex environments, where occlusions and disturbances can significantly degrade performance. Another challenge is the effective fusion of data from different sensor modalities, such as mmWave radar and cameras, to improve the accuracy and robustness of derived tracking systems. Additionally, adverse weather conditions, such as rain, can introduce noise and artefacts that further complicate the tracking process. This thesis aims to tackle these challenges by developing novel frameworks and methodologies that enhance the capabilities of mmWave radar systems.

To address these challenges, this thesis makes several key contributions. Firstly, we introduce a novel framework for extracting and utilising environmental characteristics from multi-object tracking trajectory data. This framework includes the generation of regional activity heatmaps and the classification of entry and exit points using Convolutional Neural Networks (CNNs). Secondly, we present an innovative sensor fusion framework that integrates mmWave radar and camera data to improve multi-object sensing and classification capabilities. Lastly, we then extend on our learnings from the first two contributions to propose a comprehensive approach to enhancing mmWave multi-object tracking systems in adverse weather conditions, focusing on rain-induced noise reduction and rain intensity classification. The main contributions presented by this thesis are summarised as follows:

- We develop a framework for incorporating environmental characteristics from multi-object tracking trajectory data. This framework leverages the capabilities of mmWave radar to collect detailed trajectory data of multiple objects within a given environment. The collected data undergoes a pre-processing and normalisation process to generate regional activity heatmaps, which serve as the foundation for further analysis. By dividing the observed environment into a grid and assigning tracked objects to their respective regions, the framework surfaces the regional trajectory information from the data. A CNN is then designed and trained to classify the regional activity heatmaps into predefined classes, specifically identifying entry and exit points within the environment. This classification is based on a taxonomy of movement patterns, allowing the framework to extract significant environmental characteristics from the trajectory data. The classified entry and exit points are then projected onto the multi-object tracking plane, providing a visual representation of the environmental layout. This systematic approach addresses the challenges posed by occlusions and disturbances. The proposed framework not only improves the tracking performance in dynamic and complex environments but also provides a foundation for future advancements in multi-object tracking technologies by integrating environmental understanding into the tracking process.

- We propose a sensor fusion framework that integrates mmWave radar and camera data for enhanced multi-object tracking and classification. The framework combines the strengths of both sensor modalities to improve the accuracy and robustness of tracking systems. By fusing mmWave radar data with camera images, the framework enables accurate object detection, tracking, and classification in various environments. The integration of camera data provides additional context and visual information that can be used to enhance the tracking performance of mmWave radar systems. The proposed sensor fusion framework addresses the challenges associated with labelling and training deep learning models for mmWave radar data by leveraging the complementary strengths of mmWave radar and camera sensors. The framework is designed to be adaptable to different tracking scenarios and environmental conditions, providing a unified approach to multi-object tracking and sensing.

- We research and propose an approach to enhance mmWave multi-object tracking systems in adverse weather conditions, focusing on rain-induced noise reduction and rain intensity classification. The proposed approach includes a convolutional Long Short-Term Memory (LSTM) autoencoder, referred to as mmCLAE, for noise reduction in mmWave radar signals. mmCLAE effectively removes rain-induced artefacts from the radar data, improving the tracking accuracy and resilience in adverse weather conditions. Additionally, we introduce a CNN-based model for rain intensity classification, which accurately classifies the intensity of rainfall based on mmWave radar data. By jointly addressing noise reduction and rain intensity classification, the proposed approach significantly improves the performance of mmWave multi-object tracking systems in challenging weather conditions. The developed methods provide a practical implementation of deep learning techniques to mitigate the impact of adverse weather on tracking performance, demonstrating the potential for real-world applications in dynamic environments.

1.4 Thesis Structure

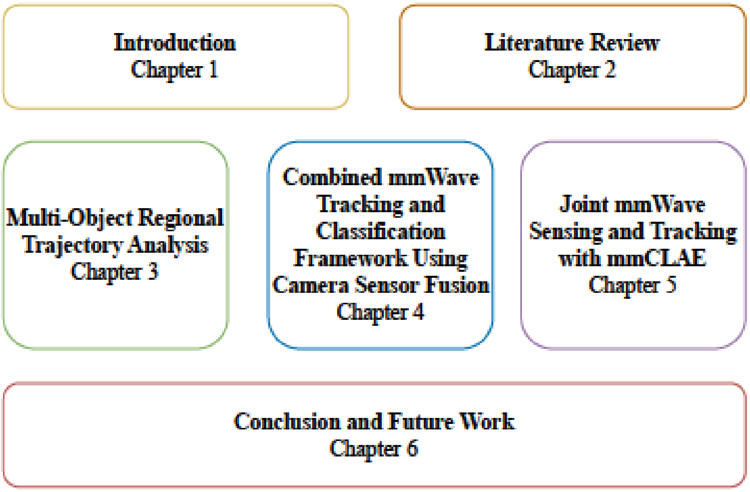

Introduction
Chapter 1
Literature Review
Chapter 2
Multi-Object Regional Trajectory Analysis
Chapter 3
Combined mmWave Tracking and Classification Framework Using Camera Sensor Fusion
Chapter 4
Joint mmWave Sensing and Tracking with mmCLAE
Chapter 5
Conclusion and Future Work
Chapter 6

Figure 1.1: Thesis organisation.

This thesis is structured into several chapters that address different aspects of mmWave radar multi-object tracking and sensing, as illustrated in Figure 1.1. The figure presents the overall thesis organisation, with our three major contributions forming the core pillars of this research.

In this chapter, Chapter 1, we introduced the fundamental concepts and motivations behind mmWave radar technology. We provided a detailed background in Section 1.1, discussing the current applications and advantages of mmWave technology in various fields. In Section 1.2, we defined the problem statement, highlighting the challenges and gaps in existing multi-object tracking systems using mmWave radar. Following this, we outlined the major contributions of this thesis in Section 1.3, summarising the key innovations and methodologies proposed to address the identified challenges. Finally, we present the overall thesis structure in this section, Section 1.4.

Chapter 2 is structured to provide a comprehensive review of the current state of the art in mmWave radar multi-object tracking and sensing. The chapter begins with Section 2.1, which traces the history of multi-object tracking from its inception to modern advancements. This is followed by Section 2.1.1, which addresses the foundational theories and methodologies that have shaped the field. Section 2.1.2 discusses the early developments in multi-object tracking, focusing on the implementation of Kalman filters and particle filters. The chapter then explores recent advances in the field in Section 2.1.3, highlighting the impact of deep learning techniques and sensor fusion. Section 2.2 specifically addresses the challenges and methodologies related to mmWave radar multi-object tracking. The chapter concludes with Section 2.5, which summarises the findings and provides concluding remarks on future directions in this field.

In Chapter 3, we present the first major contribution of this thesis, focusing on the development of a framework for incorporating environmental characteristics from multi-object tracking trajectory data. The chapter begins with Section 3.1, which provides an overview of the proposed framework and its objectives. Section 3.2 details the architecture of the Regional Dominant Trajectory Pattern (RDTP) multi-object tracking system, explaining the purpose and function of each stage. Section 3.3 describes the methodology and implementation of the proposed framework, including the data collection, pre-processing, and analysis steps. In Section 3.4, we present the experimental results and analysis, evaluating the performance of the framework in various scenarios. We conclude the chapter with Section 3.5.

Chapter 4 is structured to present a framework for combining mmWave radar and camera data to enhance tracking and classification capabilities. The chapter begins with an introduction in Section 4.1, which outlines the motivation and challenges associated with sensor fusion. Following this, Section 4.2 reviews existing sensor fusion architectures, discussing various approaches and their benefits and limitations. Section 4.3 details the proposed methodology for radar training with camera labelling and supervision, including the problem space and the proposed approach. The system design and implementation of the proposed framework are discussed in Section 4.4, providing a practical demonstration of the methodology. Finally, the chapter concludes with an evaluation of the system's performance and a discussion of the results in Section 4.5 and Section 4.6.

In Chapter 5, we discuss our final major contribution to enhance the performance of mmWave multi-object tracking systems in adverse weather conditions. An introduction to the chapter is provided in Section 5.1, outlining the motivation and objectives of the proposed approach. Section 5.2 reviews classical methodologies for noise reduction and rainfall sensing in mmWave radar systems, providing a foundation for understanding the proposed techniques. The unified system architecture is detailed in Section 5.3, describing the integration and workflow of the noise reduction and rain intensity classification modules. Section 5.4 presents the proposed noise reduction approach, including the architecture of mmCLAE and its training process. Section 5.5 describes the CNN-based method for rain intensity classification, including the model architecture and training process. The experimental results are discussed in Section 5.6, demonstrating the effectiveness of the proposed approaches in enhancing multi-object tracking performance and robustness to rain artefacts. Finally, the chapter concludes with Section 5.7, summarising the findings and discussing future research directions.

Lastly, in our final chapter, Chapter 6, we summarise the key findings and contributions of the thesis and provide recommendations for future work in the field of mmWave radar multi-object tracking and sensing. The chapter begins with Section 6.1, which provides a detailed summary of the significant contributions made by this thesis, highlighting the advancements in environmental characterisation, sensor fusion, and noise reduction techniques. Following this, Section 6.2 outlines the recommended future work, categorising potential research directions into advanced environmental characterisation, improved sensor fusion techniques, enhanced noise reduction methods, and real-world deployment.

Chapter 2

Literature Review

This chapter provides a comprehensive review of the current state of the art in mmWave radar multi-object tracking and sensing. The purpose of this chapter is to lay the foundations for the research conducted in this thesis, by identifying the primary components and processes involved in mmWave object tracking and sensing systems. This chapter is structured as follows: Section 2.1 provides a historical overview of multi-object tracking, detailing the foundational theories and early developments. Section 2.2 focuses on mmWave radar multi-object tracking, outlining the specific challenges and methodologies associated with this technology. Section 2.3 presents a typical mmWave tracking system architecture, describing the key components and processes involved. Section 2.4 explores advanced technologies and methodologies that enhance the capabilities of mmWave tracking systems. Finally, Section 2.5 summarises the findings and provides concluding remarks on future directions in this field.

2.1 History of Multi-Object Tracking

Multi-object tracking can be traced back from the 1960s to the present. The foundational theories proposed by Pylyshyn and Storm [18] in 1988 introduced the concept of parallel tracking processes in humans, ultimately laying the foundations to recognising multi-object tracking as its own field. Early developments in the field saw the implementation of using Kalman filters and particle filters, which fundamentally improved tracking accuracy and reliability, along with data association techniques like the Joint Probabilistic Data Association (JPDA) algorithm. As we approach more modern implementations of multi-object tracking, a notable trend can be observed in recent advances being heavily inspired by deep learning techniques, such as CNNs and Recurrent Neural Networks (RNNs). The remainder of this section will explore each of these historical developments in more detail.

2.1.1 Foundations of Multi-Object Tracking

Although multi-object tracking has been a concept for several decades, the theory behind it was first formally introduced as its own field of study following the publication in 1988 by Pylyshyn and Storm [18]. In Pylyshyn and Storm's work, they introduce a model that attempts to describe how multi-object tracking in the visual field takes place for humans simultaneously. Their work proposes the idea that suggests that tracking multiple targets does not exclusively happen in a serial fashion and instead, humans make use of a parallel tracking process [18]. Their work describes a process whereby humans are able to track multiple objects in the field of view by assigning a unique identifier to each object. This unique identifier is then cognitively used to maintain a track for the object as it moves through the field of view. This process is ultimately termed by Pylyshyn and Storm as the Fingers of Instantiation (FINST), otherwise known as the Visual Indexing Theory [18]. Figure 2.1 illustrates the concept of the FINST model and how cognitively each object is associated with its own identity that maintains a respective track for the object. The FINST model was later further refined in future work by Pylyshyn [19][20][21] and is a key underlying foundation to the domain agnostic research behind multi-object tracking systems.

2.1.2 Early Developments in Multi-Object Tracking

The early developments in multi-object tracking primarily focused on the implementation of Kalman filters and particle filters to improve the tracking accuracy and reliability of systems. Kalman filters, also known as linear quadratic estimation, were introduced in the 1960s and provided a robust framework for estimating the state of a dynamic system from a sequence of noisy measurements [22]. In a gener-

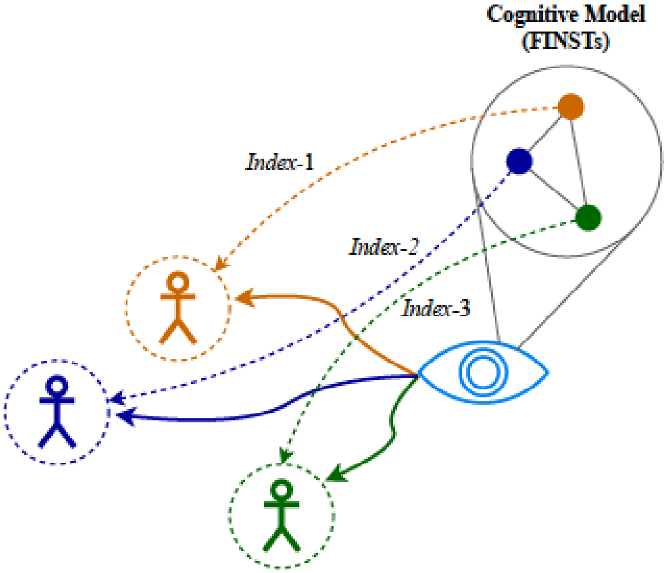


Figure 2.1: Illustration of the FINST model for human visual field multi-object tracking.

alised form, the Kalman filter can be expressed in two main steps: the prediction stage and the update stage. In the prediction stage, the filter estimates the state of the system based on the previous state and dynamics of the system [22]. The predicted state estimate at time k, denoted by $\hat{x}_{k|k-1}$, is given by:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k, \tag{2.1}$$

where $\hat{x}_{k|k-1}$ is the predicted state estimate at time k, F_k is the state transition matrix, $\hat{x}_{k-1|k-1}$ is the previous state estimate, B_k is the control input matrix, and u_k is the control input at time k.

Similarly, the predicted state covariance at time k, denoted by $P_{k|k-1}$, is expressed as:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k, \tag{2.2}$$

where $P_{k|k-1}$ is the predicted state covariance at time k, and Q_k is the process noise covariance matrix.

In the update stage, the filter utilises new measurements to refine the state estimate and reduce uncertainty [22]. The Kalman gain at time k, which determines the optimal weighting between the predicted state and the measurement, is expressed as:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}, \tag{2.3}$$

where K_k is the Kalman gain at time k, H_k is the observation matrix, and R_k is the measurement noise covariance matrix.

The updated state estimate at time k, incorporating the new measurement information, is given by:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H_k \hat{x}_{k|k-1}), \tag{2.4}$$

where $\hat{x}_{k|k}$ is the updated state estimate at time k, and z_k is the measurement at time k.

Finally, the updated state covariance at time k, reflecting the reduced uncertainty after incorporating the measurement, is expressed as:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}, \tag{2.5}$$

where $P_{k|k}$ is the updated state covariance at time k, and I is the identity matrix.

Kalman filters are particularly effective for linear systems with Gaussian noise and largely still used in tracking systems to date. However, in complex tracking problems involving non-linear systems and non-Gaussian noise, particle filters, or otherwise known as sequential Monte Carlo methods, were proven to be an effective solution. Particle filters represent the probability distribution of the object's state with a set of random samples, ultimately proving to be a more flexible approach to tracking multiple objects [23].

A generalised approach to particle filters is represented in Algorithm 1, where N is the number of particles, $x_k^{(i)}$ represents the state of the i-th particle at time k, and $p(x_k|x_{k-1}^{(i)})$ is the state transition model. The weight of the i-th particle at time k is denoted as $w_k^{(i)}$, while $p(y_k|x_k^{(i)})$ represents the likelihood of the observation y_k given the state $x_k^{(i)}$. The normalised weight of the i-th particle at time k is $\tilde{w}_k^{(i)}$, and $p(x_0)$ is the initial state distribution.

Algorithm 1 Generalised Particle Filter

```
Initialise particles $\{x_0^{(i)}\}_{i=1}^N$ from $p(x_0)$.
for time step $k$ do
    Prediction: $x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)})$
    Update: $w_k^{(i)} = p(y_k|x_k^{(i)})$
    Normalise: $\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^N w_k^{(j)}}$
    Resample: $\{x_k^{(i)}\}_{i=1}^N$ from $\{x_k^{(i)}, \tilde{w}_k^{(i)}\}_{i=1}^N$
end for
```

In addition to the early adoption of Kalman and particle filters, the development of data association techniques were pivotal in the advancement of multi-object tracking. The JPDA algorithm, introduced in the 1970s, provided an approach to associate measurements with multiple targets, ultimately improving tracking performance in environments with a high density of targets and/or clutter [24]. The JPDA algorithm ultimately relies on the basic concept of maintaining multiple hypotheses about the association of targets to the measurements, fundamentally providing more robust tracking in complex environments [25].

A generalised approach to JPDA is represented in Algorithm 2, where α_{ij} represents the prior association probabilities for each measurement-track pair (z_j, T_i), and $\beta_{k|k-1}^{[i,j]}$ denotes the association probabilities for each measurement-track pair (z_j, T_i). The predicted state estimates for each target i are denoted as $x_{k|k-1}^{[i]}$, while $P_{k|k-1}^{[i]}$ represents the predicted state covariances for each target i. The measurement is represented by z_j, and T_i denotes the track for the i-th target. The state estimates for each target i are denoted as $x_{k|k}^{[i]}$, and $P_{k|k}^{[i]}$ represents the state covariances for each target i. The combined state estimate is represented by $\hat{x}_k^{[i]}$, and the combined state covariance is denoted as $P_{k|k}^{[i]}$.

These early developments discussed in the context of multi-object tracking were foundational in the development of the multi-object tracking systems present in modern day. The introduction of Kalman filters and particle filters provided stable mechanisms for state estimation in dynamic systems, ultimately enhancing tracking

Algorithm 2 Generalised JPDA

```
Initialisation:
Initialise prior association probabilities $\alpha_{ij}$ for each measurement-track pair $(z_j, T_i)$, where $z_j$ is the measurement and $T_i$ is the track for the $i$th target.
Prediction:
Predict the state estimates $x^{[i]}_{k|k-1}$ and covariances $P^{[i]}_{k|k-1}$ for each target $i$.
Association:
for each measurement $z_j$ do
    for each track $T_i$ do
        Calculate the association probability $\beta^{[i,j]}_{k|k-1} = \frac{\alpha^{[i,j]} p(z^{[j]}_k | x^{[i]}_{k|k-1})}{\sum_{i=1}^{N} \alpha^{[i,j]} p(z^{[j]}_k | x^{[i]}_{k|k-1})}$
    end for
    Normalise the probabilities $\beta^{[i,j]}_{k|k-1} \leftarrow \frac{\beta^{[i,j]}_{k|k-1}}{\sum_{i=1}^{N} \beta^{[i,j]}_{k|k-1}}$
end for
Combined Update:
for each measurement-track pair $(z_j, T_i)$ do
    Update the state estimates $x^{[i]}_{k|k-1}$ and covariances $P^{[i]}_{k|k-1}$
    Compute the combined state estimate $\hat{x}^{[i]}_k \leftarrow \hat{x}^{[i]}_k + \beta^{[i,j]}_{k|k-1} x^{[i,j]}_{k|k}$
    Compute the combined covariance $P^{[i]}_{k|k} \leftarrow P^{[i]}_{k|k} + \beta^{[i,j]}_{k|k-1} (P^{[i,j]}_{k|k} + (x^{[i,j]}_{k|k} - \hat{x}^{[i]}_k)(x^{[i,j]}_{k|k} - \hat{x}^{[i]}_k)^T)$
end for
Resample:
Generate new tracks based on the updated state estimates $\hat{x}_i$ and covariances $P_i$.
```

accuracy and reliability. Data association techniques, such as the discussed JPDA algorithm, further improved tracking performance in complex environments with multiple targets and clutter. These advancements have been pivotal in shaping the direction of multi-object tracking research, more recent techniques that ultimately build upon these foundations will be discussed in the following section.

2.1.3 Recent Advances in Multi-Object Tracking

In the more recent years, multi-object tracking has seen significant advancements, specifically in the field of computer vision, with the rapid adoption of deep learning techniques. Deep learning methodologies, such as CNNs and RNNs, have been employed to improve object detection, feature extraction, and tracking persistence [26]. These methods have demonstrated reliability in handling complex scenarios involving occlusions, varying object appearances, and dynamic environments. For example, the use of Siamese networks for similarity learning has demonstrated promising results in the potential of re-identifying objects across frames, ultimately improving the tracking continuity [27][28][29]. Additionally, with the development of attention mechanisms being included in these models, the networks can be better focussed on more relevant features to improve tracking accuracy [30].

Another significant direction in the recent research is the integration of sensor fusion techniques, which combine data from multiple sensors such as cameras, LiDAR, and mmWave radars [31]. This multi-sensor approach takes advantage of the strengths of each independent sensor type to provide a more comprehensive perspective of the environment. As an example, cameras provide high-resolution information in the visual spectrum, mmWave radars on the other hand reliably provide distance and velocity measurements, even during conditions that are considered low-visibility, such as nighttime. The fusion of these data streams can lead to a more reliable and precise multi-object tracking system that not only improves tracking performance, but also improves the ability for the system to function in diverse and challenging environmental conditions.

2.2 mmWave Radar Multi-Object Tracking

Multi-object tracking, although advanced in some domains such as computer vision, is still a developing field in mmWave radar. The research and techniques available for achieving robust and reliable multi-object tracking and sensing, specifically with mmWave radar, are yet to be consolidated into a unified architecture. Complications, such as harsh signal propagation environments, make the task of multi-object tracking and sensing quite difficult [32]. However, it should be highlighted that tracking and sensing, unspecific to mmWave, is not a new concept in regard to radio in general. This concept has been proven successful in other types of radios, such as Impulse Radio Ultra-Wideband (IR-UWB) [33]. Therefore, the findings from multi-object tracking and sensing with alternate types of radios can be assessed for the potential to apply similar techniques with mmWave.

mmWave radar related literature can be categorised into continuous and discontinuous multi-object tracking. Continuous tracking refers to the ability to track multiple targets in an environment only whilst it is in the current field of view of the radar.

2.2.1 Discontinuous Multi-Object Tracking

Discontinuous tracking, on the other hand, is an extension of continuous tracking, whereby the targets can be tracked whilst in the current field of view and also correlated to a previous track if it re-appears in the future field of view of the radar. To clarify the difference between the two types of tracking, consider an individual, who is currently not in the field of view of the radar, performing the following sequence of events, which can also be correlated against Figure 2.2 for further clarity:

1. Moving into the radar's field of view;

2. Leaving the radar's field of view;

3. Moving back into the radar's field of view.

In the described scenario, a solution that is capable of continuous tracking is one that is capable of detecting and tracking multiple individuals in both event 1 and 3. However, a continuous tracking solution would not be capable of correlating

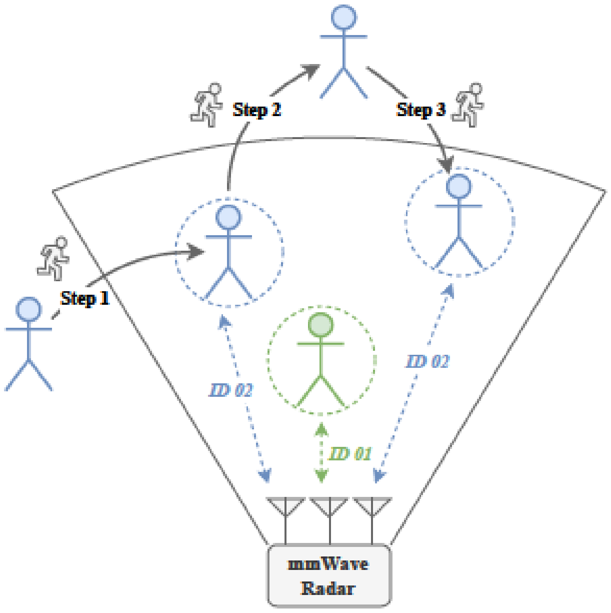


Figure 2.2: Illustration of discontinuous tracking in mmWave radar.

individuals being tracked in event 3 with previous tracks in event 1. On the other hand, a solution that is capable of discontinuous tracking is one that is capable of detecting and tracking individuals in both event 1 and 3, as well as recognising it is the same individual across the two events. Thus, a discontinuous tracking solution is one that can correlate and track multiple targets across a discontinuous sequence of events.

2.2.2 Applications of mmWave Multi-Object Tracking and Sensing

A sophisticated combined mmWave multi-object tracking and sensing system, capable of reliably discontinuously tracking, has numerous applications. Such a system could serve as a new level of security and surveillance by providing a foundation that detects threats or concerns not easily identified by vision-based systems, all while maintaining individual privacy. In the healthcare industry, this technology could enable mass patient monitoring, allowing for passive and continuous observation of vital metrics that typically require manual measurement by medical professionals. This has the potential to lead to earlier detection of patient complications and more timely interventions. Additionally, a mmWave multi-object tracking and sensing system could serve as an affordable, wide-scale analytical and auditing platform

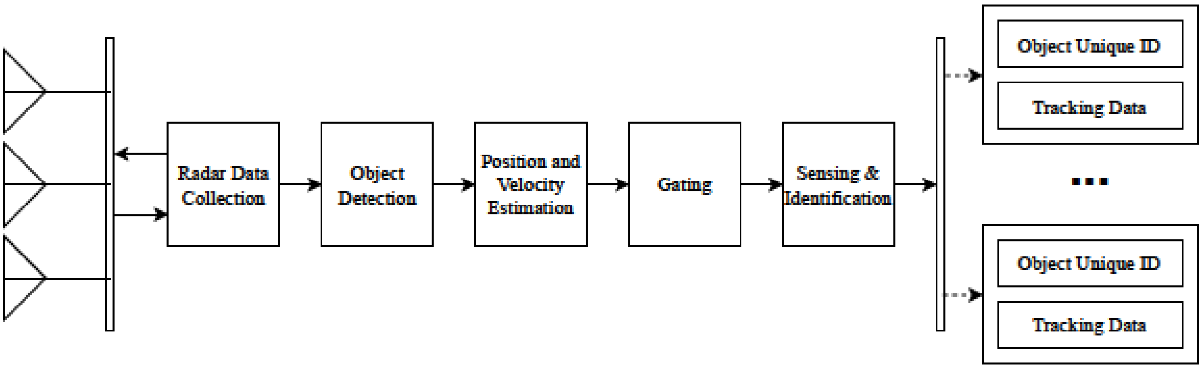


Figure 2.3: mmWave tracking architecture block diagram.

for public spaces like shopping centres and parks. It could provide insights into optimising space layouts, identifying congestion points, and understanding specific behaviours triggered by environmental events.

Furthermore, the integration of advanced sensing methodologies, such as micro-Doppler feature analysis and sensor fusion, can significantly enhance a system's reliability and accuracy. By combining data from multiple sensors, such as cameras and mmWave radars, the system can achieve a more comprehensive understanding of the environment. This multi-sensor approach not only improves tracking performance but also ensures the system's functionality in diverse and challenging conditions. The potential to incorporate identification strategies, such as gait recognition and shape profiling, further extends the system's capabilities, enabling it to uniquely identify and track individuals discontinuously. Collectively, these advancements serve as the potential to drive the way for innovative applications in various fields, from positively developing public safety to improving healthcare outcomes.

2.3 Typical mmWave Tracking System Architecture

An overview of the architectural model for multi-object tracking with mmWave radar, from data collection to tracked target information, is illustrated in Figure 2.3. The intention of this architecture depicted in Figure 2.3 is to provide a foundation for comparing and contrasting mmWave multi-object tracking research, encompassing both continuous and discontinuous tracking methodologies.

In order to help understand the events that take place to successfully perform discontinuous multi-object tracking with mmWave, the system can be illustrated as a series of five chained components. These five components and the sequence in which they are invoked is illustrated in Figure 2.3. The generalised aim of the system is to comprehend the influence multiple targets simultaneously have on radar chirps. This signal disturbance translates to information being exploited to initiate or resume a maintained track on an object, whilst it is in the radar's field of view. The system should ultimately produce a stream of uniquely identifiable objects along with their corresponding tracking context. The overall system architecture and sequence of components is a well established pattern in radar tracking literature. The uniqueness of an mmWave tracking system is ultimately held in the implementation of the system components and the mechanisms that are employed to characterise the tracked objects. The remainder of this section will explore and describe the purpose of each stage illustrated in the generalised architecture shown in Figure 2.3.

2.3.1 Radar Architecture

The radar architecture of a typical multi-object tracking system consists of the components required to ultimately collect the data describing the observed environment. Over the last couple of years, single board general-purpose mmWave radars have become readily available as off the shelf products, such as the Texas Instruments (TI) family of industrial and automotive mmWave radar sensors. A general architecture for a single board mmWave radar sensor is illustrated in Figure 2.4. The architecture of a single board mmWave radar sensor is usually comprised of 4 main component stacks, the radio components, analog components, digital components, and the software [34]. The radio components are responsible for ensuring the radar signals are sent and received. This usually includes Transmitter (Tx) and Receiver (Rx) antenna arrays, along with the necessary synthesisers and mixers to construct the Intermediate Frequency (IF) signal. The analog components are responsible for conditioning the IF signal, which usually involves various amplifiers and filters. The digital components are responsible for processing the signal, this will require an Analog-to-Digital Converter (ADC) to convert the IF signal to a digital signal, along with specialised signal processing units, such as a Digital Signal Processor

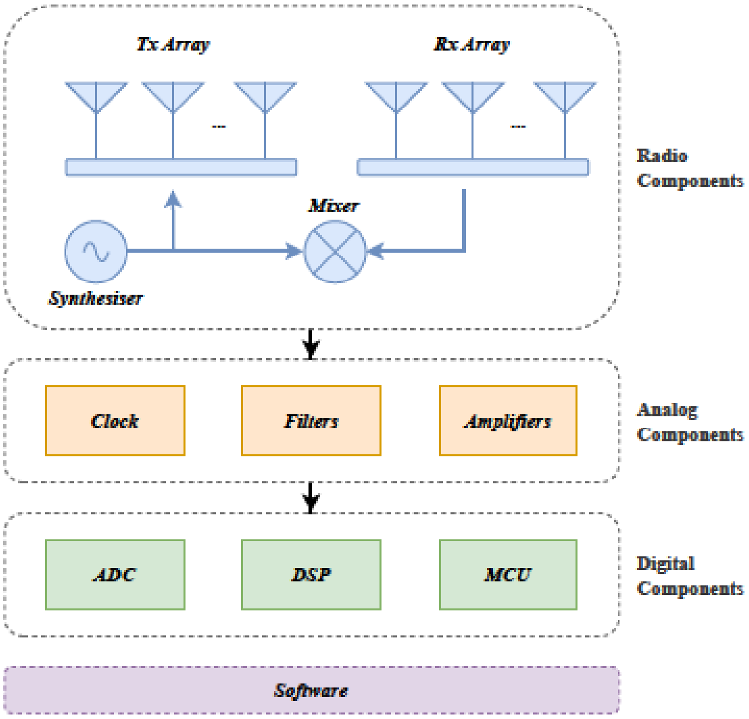


Figure 2.4: Single board mmWave radar sensor architecture.

(DSP), and finally a Microcontroller Unit (MCU). Lastly, the software components are responsible for managing the overall operation of the radar sensor, this usually includes the radar sensor firmware and any software that interfaces with the radar sensor.

There are a number of considerations to be made when determining the antenna configuration to employ for a mmWave radar multi-object tracking system. Specifically, an acknowledgement should be made regarding the components that contribute to the instability and non-ideal nature of the transmitted signal [35]. A Multiple Input Multiple Output (MIMO) antenna array is the most commonly utilised antenna configuration in radar systems. This is primarily due to its spatial diversity characteristics, ultimately resulting in a more superior detection performance, compared to traditional directional or phased-array antenna configurations [36], [37]. A study conducted in [37] demonstrates statistically the performance advantages of MIMO systems in comparison to alternate antenna models, highlighting the ability to exploit the spatial diversity of a MIMO system to ultimately overcome target fading in radar applications.

2.3.2 Position and Velocity Estimation

Once the appropriate radar architecture has been decided, a strategy for calculating the estimated position and velocity of reflected points should be determined. It should be acknowledged that the position of a reflected point is comprised of the range and azimuth of the reflected point, with respect to the radar.

Frequency Modulated Continuous Wave (FMCW) radar systems are commonly used in mmWave multi-object tracking systems due to their ability in providing high resolution range and velocity estimates. The fundamental principle behind FMCW is the transmission of a frequency-modulated signal, referred to as a chirp, which ultimately sweeps linearly over a range of frequencies during a specified chirp time window. A FMCW chirp signal can be mathematically represented as:

$$s(t) = A\cos\left(2\pi\left(f_0 t + \frac{S}{2}t^2\right)\right), \tag{2.6}$$

where A is the amplitude of the signal, f_0 is the initial frequency of the chirp, S is the slope of the chirp (frequency gradient), and t refers to time.

The chirp signal starts at frequency f_0 and increases linearly to $f_0 + B$ over the total chirp duration T, where B refers to the bandwidth of the chirp, as illustrated in Figure 2.5. The slope S of the chirp, which defines the rate of frequency change, is given by:

$$S = \frac{B}{T}. \tag{2.7}$$

When the transmitted chirp signal reflects off an object, the received signal is a time-delayed version of the transmitted signal. The time delay τ, which is proportional to the distance R of the object from the radar, is expressed as:

$$\tau = \frac{2R}{c}, \tag{2.8}$$

where c is the speed of light.

The received signal, representing the time-delayed version of the transmitted chirp after reflection from an object, is mathematically expressed as:

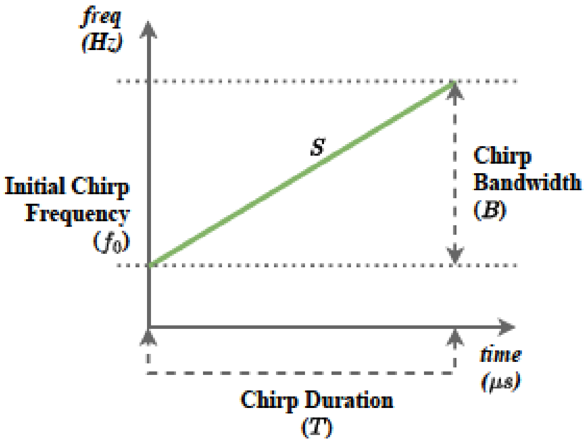


Figure 2.5: Illustration of a FMCW chirp signal showing the linear frequency sweep over time.

$$r(t) = A\cos\left(2\pi\left(f_0(t-\tau) + \frac{S}{2}(t-\tau)^2\right)\right), \tag{2.9}$$

The received signal is then mixed with the transmitted signal to produce the IF. The IF signal, obtained by multiplying the transmitted signal $s(t)$ with the received signal $r(t)$, is given by:

$$IF(t) = s(t)r(t), \tag{2.10}$$

which represents the fundamental mixing operation in FMCW radar processing.

The IF signal consists of the beat frequency, which is ultimately the difference between the transmitted and received frequencies. After mathematical simplification of the mixing operation, the IF signal can be expressed as:

$$IF(t) = A^2\cos\left(2\pi\left(\frac{2SR}{c}t - \frac{2f_0R}{c}\right)\right), \tag{2.11}$$

where the first term represents the beat frequency proportional to range, and the second term represents the constant phase offset.

In an environment where multiple objects are influencing the IF signal, a Fast Fourier Transform (FFT) can be performed on the IF signal to derive a frequency domain expression of the signal. This transformation allows the identification of distinct frequency peaks, each corresponding to a specific detected object. The distance of each detected object can then be calculated based on the frequency present in the IF signal. This relationship is given by:

$$R_x = \frac{cf_{IF}^{[x]}}{2S}, \tag{2.12}$$

where R_x is the distance of the detected object x, $f_{IF}^{[x]}$ is the frequency of the detected object x in the IF signal, c is the speed of light, and S is the slope of the frequency modulation.

Through analysing the frequency peaks in the FFT of the IF signal, otherwise known as the range-FFT, it is possible to determine the range of multiple objects simultaneously. This method is particularly effective in environments with multiple targets, as it allows for the separation and identification of each object's distance based on their unique frequency signature.

The velocity of a detected object can ultimately be obtained by analysing the phase difference between consecutive chirps corresponding to the same object. In the situation where multiple objects are present at the same distance from the radar, the phase difference of the FFT of the IF signal will have multiple objects encoded within it. As a result, a second FFT should be performed, labelled as the Doppler-FFT, which will ultimately reveal peaks of phase differences corresponding to the number of detected objects. The velocity of a given object V_x, derived from the Doppler-FFT analysis, is given by:

$$V_x = \frac{\lambda \omega_x}{4\pi T}, \tag{2.13}$$

where ω_x is the phase difference of the detected object in the IF signal, λ is the wavelength of the transmitted signal, and T is the chirp duration.

The last component of interest, required for multi-object tracking, that can be derived from the reflected signal is the horizontal angle, relative to the radar, of the object that caused the signal reflection. This is termed as the Angle of Arrival (AoA). For two Rx antennas, the AoA of a reflected signal θ_x is expressed as:

$$\theta_x = \sin^{-1}\left(\frac{\lambda \omega_x}{2\pi d}\right), \tag{2.14}$$

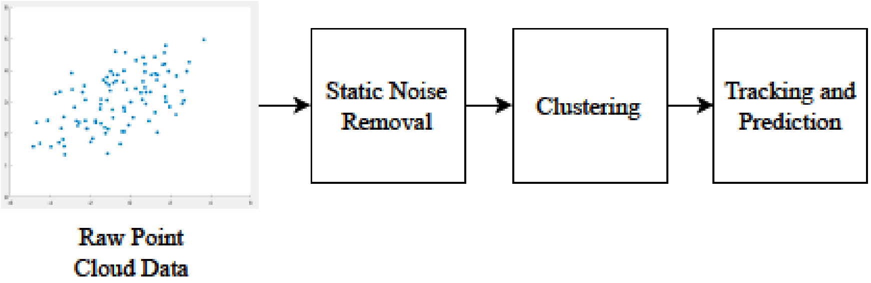


Figure 2.6: Generalised stages of association and tracking in an mmWave tracking architecture system.

where d is the distance between the two Rx antennas. In an architecture where multiple Rx antenna pairs are presented, the final AoA can be derived by determining the average AoA result from all Rx antenna pairs.

The ultimate outcome of this stage in a mmWave tracking system is to obtain the necessary information to construct a 2-dimensional plot that illustrates the reflection points in the environment. Estimating the range, angle, and velocity of each reflection point is sufficient to construct a plot of this nature. The most common way to illustrate this information is to plot it in a point cloud graph. The position of each point is determined by the range and AoA, while the colour or size of the points can be used to represent the velocity of the detected objects.

Combining the range, velocity, and AoA information, the mmWave radar system can effectively display multiple objects in the environment, providing a comprehensive visualisation of their positions and movements.

2.3.3 Association and Tracking

The association and tracking component of an mmWave tracking system should fundamentally consume the information that illustrates reflection points, deduced in Section 2.3.2 of this chapter. Using this information, usually in point cloud format, the process illustrated in Figure 2.6 highlights the typical stages involved in achieving a set of continuously tracked objects from the obtained point cloud data.

The first processing stage illustrated in Figure 2.6, static noise removal, refers to a process whereby any points in the point cloud data that are present in both frame N_x and N_{x-1} are deemed as static noise and removed from frame N_x. This noise removal technique is typical in current mmWave multi-object tracking systems. One key assumption that is made in this noise removal attempt is that targets of interest must always be moving to be tracked. Therefore, any targets that are mostly stationary, such as a person sitting at an office desk, cannot reliably maintain a track under this assumption. This chapter explores advanced strategies in Section 2.4 that attempt to overcome this assumption when tracking multiple-objects.

Proceeding to the second stage in Figure 2.6, although the static noise has been removed it cannot be said that the data points present are noise free. Due to the multi-path theory, it is likely that there will be a number of data points present that are ghosts of the actual reflected objects [38]. As a result, an appropriate correlation and clustering algorithm is usually employed to alleviate this challenge and gate related reflection points. The most successful clustering algorithm that is used in point cloud data is the Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, originally presented in [39].

The DBSCAN algorithm ultimately groups together points that are closely packed together, while flagging points that are alone in low-density regions as outliers in the dataset. The algorithm essentially requires two parameters: the radius ϵ that defines the region around a point, and the minimum number of points $MinPts$ required to form what is classified as a dense region [40]. In the context of mmWave multi-object tracking, the radius ϵ can be considered as the maximum distance between two points for them to be considered as part of the same cluster. The minimum number of points $MinPts$ can be considered as the minimum number of points required to identify as an object. The DBSCAN algorithm can ultimately be expressed in a generalised form as seen in Algorithm 3.

In DBSCAN, a point P can be considered a core point if its neighbourhood has at least $MinPts$ points. A point Q is considered directly density-reachable from P if Q is within the ϵ-neighborhood of P. Additionally, point Q is density-reachable from P as long as there is a chain of points $P_1, P_2, \ldots, P_n$ where $P_1 = P$ and $P_n = Q$, and each P_{i+1} is also directly density-reachable from P_i. Lastly, point Q is considered density-connected to P providing there is another point O, such that both P and Q are density-reachable from O.

Algorithm 3 DBSCAN

```
Input: Dataset $D$, radius $\epsilon$, minimum points $MinPts$
Output: Set of clusters $C$
Initialise $C \leftarrow \emptyset$
Mark all points in $D$ as unvisited
for each point $P$ in $D$ do
    if $P$ is unvisited then
        Mark $P$ as visited
        $N \leftarrow$ neighbourhood of $P$ using $\epsilon$
        if $|N| \geq MinPts$ then
            $C_i \leftarrow$ new cluster containing $P$
            ExpandCluster($C_i$, $P$, $N$, $\epsilon$, $MinPts$)
            Add $C_i$ to $C$
        else
            Mark $P$ as noise
        end if
    end if
end for
Procedure ExpandCluster($C_i$, $P$, $N$, $\epsilon$, $MinPts$)
for each point $P'$ in $N$ do
    if $P'$ is unvisited then
        Mark $P'$ as visited
        $N' \leftarrow$ neighbourhood of $P'$ using $\epsilon$
        if $|N'| \geq MinPts$ then
            $N \leftarrow N \cup N'$
        end if
    end if
    if $P'$ is not yet part of any cluster then
        Add $P'$ to $C_i$
    end if
end for
```

mmWave radar tracking systems predominately either use the DBSCAN algorithm for clustering and association of data points or implement an alternate clustering algorithm that is typically a variation of the DBSCAN algorithm. The presented variations of DBSCAN usually outperform the original algorithm in the datasets specific to the implementation [41]–[45]. Once the point cloud data points have been correlated and clustered together to form a set of groups, a common strategy to decide the position of a holistic object is to logically take the centroid of the respective cluster.

After guaranteeing reliable point cloud associations and clustering has been made to collate the points associated with the various objects in scene, the next step is to persist a track for each of these objects across a continuous set of frames. In the vast majority of mmWave multi-object tracking systems, the tracking aspect in its simplest form is primarily achieved through the use of a Kalman filter, as mentioned earlier in Section 2.1.2. Kalman filtering is a widely adopted approach to efficiently provide tracking and estimations [46]. Many variations of Kalman filters have been presented in the literature to ultimately optimise the performance and outcome of tracking an object via mmWave radar. The research conducted by [47] demonstrates an example where Kalman filtering was applied to successfully track multiple objects with respect to an mmWave radar. For each object detected by the radar, an individual Kalman filter is applied for tracking and estimation of the specific object. Each Kalman filter is then run independently [47]. The authors of [47] highlight that the success of implementing a Kalman filter to track and estimate the position of an object is highly dependent on the clustering and data association techniques that have been employed for object detection.

2.3.4 Sensing and Identification

The final component of a mmWave tracking system is any sensing and identification strategies that might be employed alongside the tracking architecture. The desired outcome of this component of the system is to ultimately perform a particular sensing or identification task and associate the outcomes with the tracked objects. Currently, there is no typical way this component of a mmWave tracking system is jointly achieved.

Specific sensing methodologies can include various techniques, such as micro-Doppler analysis, which utilises the short wave-length of mmWave to extrapolate fine-grained motion characteristics of objects. For example, micro-Doppler signatures could be used to detect human vital signs such as heartbeat and breathing patterns, which can be particularly useful in healthcare applications [48]. Furthermore, these signatures can also help in distinguishing between different types of movements, such as walking, running, or even subtle gestures, thereby enhancing the tracking system's ability to understand the context of the tracked objects.

Identification strategies, on the other hand, focus on uniquely identifying objects within the radar's field of view. This can be achieved through methods such as gait recognition, where the unique walking patterns of individuals are used to identify them [49]. Another approach could be through the use of Radio Frequency Identification (RFID) tags or a Reconfigurable Intelligent Surface (RIS) that can encode a unique signature for each object. Shape profiling is another technique where the physical dimensions and contours of an object are used to create a unique identifier [50]–[52].

Sensing and identification components of mmWave tracking can be loosely coupled with the ability to discontinuously track a particular object. Discontinuous tracking in the context of this thesis refers to the ability to re-establish or correlate a previous track of a specific object to a current track. Specific examples of this are explored in Section 2.4 of this chapter.

2.4 Advanced Technologies and Methodologies

In the previous section of this chapter, a typical mmWave radar multi-object tracking system and its components were explored and discussed. This section of the chapter aims to describe the state-of-the-art advancements in mmWave multi-object tracking and how it contributes to the generalised multi-object mmWave tracking architecture explored in Section 2.3. Figure 2.7 highlights the areas that are being explored in this section of the chapter in contrast to the typical system architecture presented in Figure 2.3. The system architecture stages radar data collection, position and velocity estimation, and gating are all mature with regard to multi-object

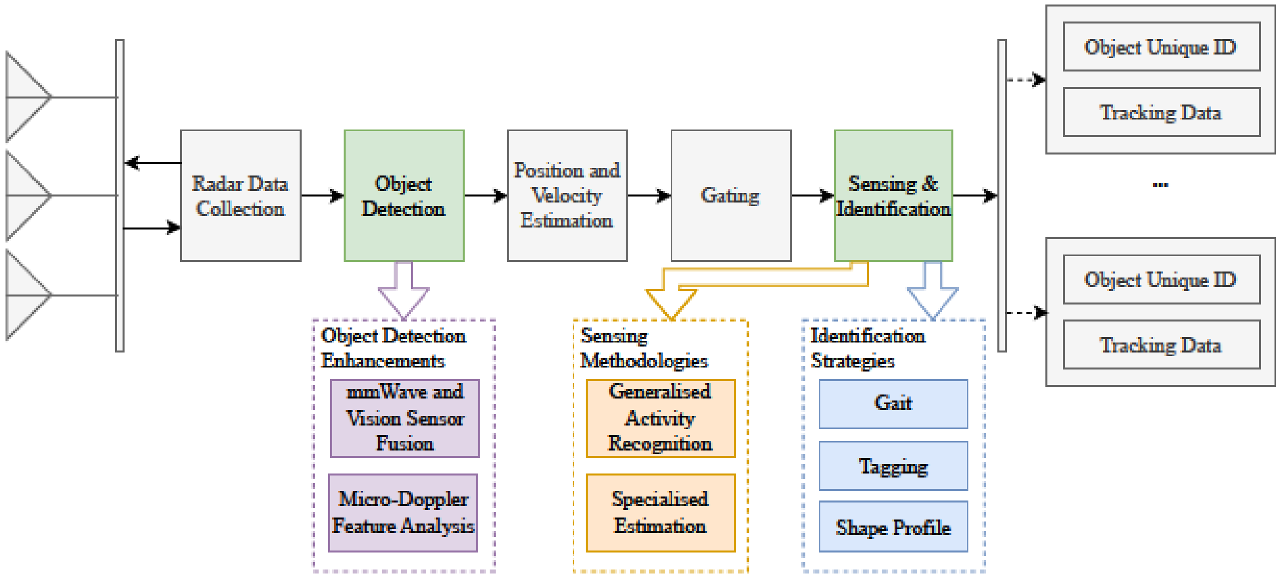


Figure 2.7: Areas explored and discussed in Section 2.4 in contrast to the typical multi-object mmWave tracking architecture block diagram presented in Figure 2.3.

tracking. The areas which require most attention for developing advanced methodologies is object detection, and joint tracking, sensing and identification. These areas specifically are receiving the most focus primarily due to the limitations that are faced in the current typical multi-object tracking architectures.

For each of the below subsections, the methodologies presented will be compared and contrasted with respect to the below criteria. The relevant advantages and disadvantages for the methodologies discussed will be outlined for each criterion. The criteria that will be used to assess the methodologies is:

- **Adaptability:** The ability to apply the methodology in a generalised form so that it can contribute to advancing the system architecture presented in Figure 2.3.
- **Performance:** The overall performance of the methodology with respect to its suitability for real-time applications.
- **Accuracy:** A consideration regarding the accuracy metric of the techniques presented in the specific methodology.
- **Specificity:** The sensitivity of the methodology in regard to the particular event/action being measured or characterised. This criterion provides an op-

portunity to consider any event overlap that the methodology might have, such as false positives.

2.4.1 Object Detection Enhancements

One of the fundamental flaws in a typical mmWave tracking system is the reliance on static noise filtering. This reliance ultimately spawns challenges related to the reliable tracking of a static object. As a result, a large focus on methodologies and strategies to alleviate these challenges can be seen in the literature. The two overarching themes that encompass the research direction for addressing these challenges are sensor fusion and micro-Doppler feature analysis.

Sensor fusion, in the context of this thesis, refers to the combination of data from additional sensors in addition to a mmWave sensor. A common approach to this in the literature is to fuse camera data with the data obtained from the mmWave sensor to achieve a more coherent and comprehensive object detection algorithm. One of the primary challenges with fusing camera and mmWave radar detections is that they are a heterogeneous pair of sensors [53]. The plane in which the radar detections are aligned with is different to that of the camera detection. Therefore, this can make associating the detections between the two sensors quite difficult [53]. Research presented by [53] demonstrates a novel approach to solving the association challenge. In the methodology presented in [53], the authors define the concept of error bounds to assist with the data association and gating within a fusion Extended Kalman Filter (EKF). The concept of error bounds provide a criterion to define the behaviour of the individual sensors before and after the sensor fusion [53].

In the fusion-EKF presented in [53], the radar point cloud clusters are formed using an approach similar to the typical architecture discussed in Section 2.3 of this chapter, with DBSCAN. Similarly, the bounding boxes on the image plane are initially formed in isolation to the radar and then sent to the fusion-EKF to be associated and tracked with the radar clusters. The plane of the camera data points is transformed from an image plane to a world plane using a homography estimation method [53]. A warped bird eye view of the camera data points can then be estimated using the world coordinates. The estimated warped birds eye view can then be compared and associated with the radar point cloud data points [53]. In the fusion-EKF presented by [53], the error bounds are updated using data points from both of the sensors (as opposed to independently) and the warped bird eye view of the image plane is calculated for each sample point. As a result, the authors of [53] demonstrate that although this yields a higher association accuracy, a time synchronisation challenge is faced between the sensors. This challenge is resolved in the research by ensuring timeline alignment between the sensors and a synchronisation strategy is employed by comparing certain regions of the fusion-EKF output with the error bounds [53]. The experimental results presented by [53] appear to demonstrate a higher reliability in real-time target detection and persisted tracks, compared to a radar alone.

Exploiting micro-Doppler in mmWave radar systems is actively being sought as another angle to devise methodologies that resolve the challenge of static object detection and localisation. Specifically in the context of human detection, biometric information, such as heartbeat and breathing are being explored as potential features that are measurable through micro-Doppler. A study performed by [54] demonstrates an algorithm designed to localise multiple static humans using their individual breathing pattern. The research performed by [54] highlights that the time of flight of a signal is minimally impacted by the small movements of a breathing chest cavity. As a result, the sub-millimetre movements are lost when performing static background removal between two consecutive frames, 12.5 milliseconds apart in the case of the experiment performed by [54]. To counter this loss of information, the authors in [54] suggest subtracting the static background from a frame that is a few seconds apart, 2.5 seconds in the case of the research performed by [54]. In doing this, the sub-millimetre movements are ultimately exaggerated in comparison to a truly static object and therefore are left intact when performing a removal of static data points.

The authors of [54] make note that removing static background points from a frame that is a few seconds apart does not work for a non-static object, such as a person walking. This is due to the principle that the movements appear exaggerated when compared to a frame a few seconds apart, so [54] notes that walking appears 'smeared' in this regard. Based on this differing outcome with static and dynamic objects, the algorithm presented in [54] employs independent different background removal strategies; one for static objects using a long window and one for dynamic objects using a short window. The experimental results presented in [54] demonstrate a high accuracy of 95%. It should be noted that the experiments performed by [54] do not appear to quantify the success of both moving individuals and static individuals simultaneously within the scene. The radar architecture used in the research presented by [54] is slightly different to the mmWave tracking system that has been discussed in this chapter. However, the research performed by [54] illustrates the potential to use vital signs as a means of detecting a static object. It would be of interest to assess the range potential of implementing a static localisation algorithm of this nature using a mmWave tracking system architecture.

The literature explored in this chapter regarding vision sensor fusion and biometric micro-Doppler feature analysis presents viable approaches to enhance traditional object detection methods. These approaches enable the tracking of objects that transition between dynamic and static movement states. Table 2.1 outlines the advantages and disadvantages of the two methodologies with respect to the comparison criteria. Although individually both methodologies prove viable, it would be interesting to consider a combination of both methodologies to compliment each other. Specifically, incorporating a micro-Doppler feature analysis component to the vision system could in turn remove the need for utilising the universal background subtraction algorithm [55] for identifying moving objects in an image. This could potentially be considered as a three component sensor fusion approach, where camera data points, static radar data points, and dynamic radar points are fused.

2.4.2 Sensing Methodologies

Sensing is not typically considered a usual aspect that is present in an object tracking system. However, it is a stream of research that has been investigated independently and has the potential when integrated with a tracking system to enhance the tracking systems sensitivity and reliability. An enhancement to the tracking system through sensing could ultimately spawn through the additional extracted features that the sensing solution provides, granting more data points that can be incorporated into the tracking estimation and prediction. The advanced sensing methodologies that are explored in this section can be classified as either general activity recognition or

Table 2.1: A comparison of methodologies explored for the enhancement of object detection in an mmWave tracking architecture.

Criterion	**mmWave and Vision Sensor Fusion**	**Breathing Micro-Doppler Feature Analysis**
Adaptability	✓ Low architecture assumptions. ✓ Unified sensor point cloud data. × Unified plane projection overhead.	✓ Decoupled from architecture dependencies. × Specialised noise treatment.
Performance	✓ Suitability demonstrated in the literature. × Potential time synchronisation drift.	✓ No impact to typical multi-object detection. × Immature understanding on technique overhead.
Accuracy	✓ Azimuth angle accuracy improved. ✓ Multi-object track persistence improved. × Immature system understanding regarding the compromise of a single sensor (i.e. dark room).	✓ High for multiple dynamic objects. ✓ Uncompromised fixed multi-object tracking. × Immature understanding regarding accuracy and range relationship.
Specificity	✓ All moving objects have a presence in radar and vision that can be correlated. × Fixed objects of interest are not typically distinguishable.	✓ Technique not constrained to breathing. × Immature understanding of simultaneous static and fixed multi-object tracking.

specialised estimation methodologies.

General activity recognition can be considered as a class of sensing methodologies that have an underlying objective of classifying a broad set of movements or activities that a given object in the field of view might exhibit. One stream of research that dominates this class of sensing methodologies is Human Activity Recognition (HAR). Traditionally, a radar based HAR system relied on machine learning techniques such as random forest classifiers [56], dynamic time warping [57], and Support Vector Machines (SVMs) [58]. In comparison to a deep learning based approach, these techniques are usually computationally less taxing due to their lower complexity. However, relying solely on conventional machine learning techniques for HAR contrastingly presents several limitations. A survey conducted by the authors of [59] provides a thorough critical analysis over the evolution of radar-based HAR. In [59], a conventional machine learning approach to HAR is considered to make optimisation and generalisation of the HAR solution difficult. The authors of [59] highlight three fundamental limitations of machine learning techniques with respect to a HAR system. The first acknowledges the approach in which feature extraction takes place, specifically a manual procedure based on heuristics and domain knowledge which is ultimately subject to the human's experience [59]. The second limitation identified relates to the fact that manually selected features tend to also be accompanied by specific statistical algorithms that are dependent on the trained dataset. As a result, when applying the trained model to a new dataset the performance is typically not as good as the dataset that was used to train the model. Lastly, the authors of [59] highlighted that the conventional machine learning approaches used in a radar based HAR system primarily learn on discrete static data. This poses a difference between the data that is used to train a model and the data that the model is subject to during real-time testing. The real-time data is principally continuous and dynamic in nature. The survey conducted by [59] explores the potential for deep learning to assist in alleviating these limitations in machine learning radar-based HAR systems.

Although there are some limitations with using conventional machine learning approaches, it should also be acknowledged that there have been successful applications of radar-based HAR using these techniques. The research presented in [60] identifies recent work that attempts to classify three different walking/movement patterns:

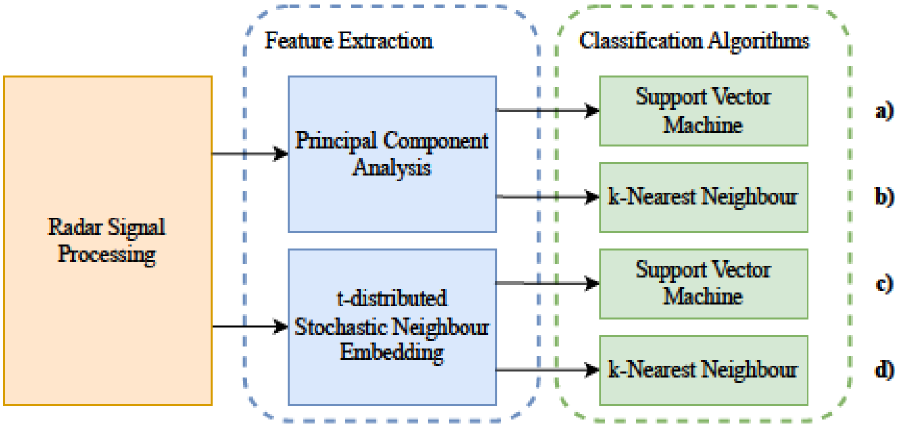


Figure 2.8: Walking classification system designs explored in [60]; **a)** Principal Component Analysis (PCA) combined with SVM classification; **b)** PCA combined with k-NN classification; **c)** t-Distributed Stochastic Neighbour Embedding (t-SNE) combined with SVM classification; **d)** t-SNE combined with k-NN classification.

- Slow walk;
- Fast walk;
- Slow walk with hands in pockets.

The authors of [60] attempt to classify these walking patterns comparing the performance between an approach using K-Nearest Neighbour (k-NN) and SVM. The four system designs explored in the work presented by [60] can be seen illustrated in Figure 2.8. In [60], both the range-Doppler and Doppler-time data is incorporated into feature extraction. The impact each of the walking patterns have in the range-Doppler and Doppler-time maps are illustrated in the form of a heatmap [60]. In this form, it can be seen that the change in walking speed (the difference between slow and fast walking) results in a dramatic change in the range-Doppler and Doppler-time maps. Whereas, maintaining a consistent walking speed and with hands in the pocket has less of a notable difference.

In regard to extracting the features, the authors of [60] explore and compare two potential approaches, using either PCA or t-SNE. Both of which are non-supervised transform algorithms. The two feature extraction methods are compared against each other whilst equally being applied with the two aforementioned classification methods. The permutations of feature extraction methods with classification algorithms explored are shown in Figure 2.8. The results obtained from [60], for each of the explored system designs in Figure 2.8, demonstrate the capability of classifying fast and slow walking with high accuracy. Using the feature extraction methods and classification algorithms explored in [60], the authors note a 72% accuracy in classifying slow walking with hand in the pocket.

Another piece of leading research in radar-based HAR is RadHAR presented in [61]. In [61], the authors explore a range of classification approaches, including both conventional machine learning algorithms and deep learning based algorithms. The primary objective of the RadHAR system is to classify five human movement activities; walking, jumping, jumping jacks, squats, and boxing.

Unlike the research presented in [60], in [61] the data that is used for classification originates from point cloud. The point cloud data is first voxelised to ensure a uniform frame size, despite the number of points, before feeding to the classification algorithm. Using the voxelised point cloud data, a SVM, Multi-Layer Perceptron (MLP), LSTM and CNN combined with LSTM were trained and compared against each other.

The results of the research conducted in [61] demonstrate that the classification algorithm with the highest accuracy, 90.47%, is that of a combined time-distributed CNN and bidirectional LSTM. The authors of [61] hypothesise that the high accuracy of this approach can be attributed towards the fact that the time-distributed CNN learns the spatial features of the point cloud data, whilst the bidirectional LSTM learns the time dependent component of the activities being performed.

Specialised estimation, as opposed to general activity recognition, is a class of sensing that has a primary focus on a single objective that can be measured. Measurement of this nature, of course, should be considered as an estimation. This class of sensing has overlap with features that can be used as a criteria for identifying a specific object. More details on features with the potential to be used as an identification strategy are addressed in Section 2.4.3 of this chapter. The primary driver behind research in radar-based specialised estimation methodologies originates from a human health perspective. The ability to determine human vital signs passively is an area in which mmWave radar is being explored as a viable solution. A study performed in

[62] demonstrates a solution named ‘mBeats’ which implements a moving mmWave radar system that is capable of measuring the heartbeat of an individual. The proposed ‘mBeats’ system implements a three module architecture. The first module is a user tracking module, which the authors of [62] state that the system utilises a standard point cloud based tracking system, as illustrated in Section 2.3 of this chapter. The purpose of this module is to ultimately find the target in the room. It should be noted that in [62] an assumption is made that there will only be one target in the field of view. The second module proposed in [62] is termed as the ‘mmWave Servoing’ module. The purpose of this module is to optimise the angle in which the target is situated from the mmWave radar to give the best heartbeat measurement. To achieve this, the authors of [62] specify the ultimate goal of this module as obtaining peak signal reflections for the target’s lower limbs, since the mmWave radar is situated on a robot at ground level. Using the Peak To Average value as a determinant for the reflected signal strength, the authors define an observation variable which is incorporated by a feedback Proportional-Derivative controller to orientate the radar in the direction that yields the highest signal strength.

The last module is the heart rate estimation module, responsible for ultimately determining the target’s heart rate from a set of different poses. The poses consist of various sitting and lying down positions. The authors of [62] acknowledge that heartbeats lie in the frequency band of 0.8Hz - 4Hz, and therefore implement a biquad cascade Infinite Impulse Response (IIR) filter to eliminate unwanted frequencies and extract the heartbeat waveform. A CNN is selected in [62] as the predictor due to the heartbeat detection problem being considered as a regression problem. The authors state that a key challenge with using a CNN for this problem is estimating the uncertainty of the result. Uncertainty in this problem is ultimately caused by measurement inaccuracies, sensor biases and noise, environment changes, multipath, and inadequate reflections [62]. To overcome this, the authors of [62] cast the problem into a Bayesian model, defining the likelihood between the prediction and ground truth ($\mathbf{y}$) as a probability following a Gaussian distribution. This ultimately results in a loss function that quantifies the uncertainty-aware prediction error, expressed as:

$$loss(\mathrm{x}) = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|_2}{2\sigma^2} + \frac{1}{2}\log\sigma^2, \tag{2.15}$$

where the CNN predicts a mean $\hat{\mathbf{y}}$ and variance σ^2. Using this approach the authors of [62] compare the outcome of their model with three other common signal processing approaches (FFT [63], peak count [64], and auto-correlation [65]) with accuracy as the metric that is compared.

In the results presented in [62], it can be seen that the other approaches fail to maintain an accuracy above 90% in all poses, whereas the CNN presented in [62] does maintain a high accuracy for the selected poses. The authors acknowledge that in the current system the target must maintain static whilst performing the heartbeat measurement, and that future work will be focused on measuring a moving object. It would also be interesting to assess the viability and challenges of this approach in a multi-person scene.

The underlying theme of the sensing methodologies explored in this chapter is that independently they are successful in the goal they aim to achieve. However, there is a lack of acknowledgement in the literature regarding the suitability of these methodologies in a combined holistic tracking and sensing architecture. It would not only be interesting to assess their suitability in such a system, but also how they may contribute to enhance the sophistication and reliability of such a tracking system. Table 2.2 outlines the advantages and disadvantages of the explored sensing methodologies, with respect to the comparison criteria. It can be seen in this table that both methodologies explored fail to address the challenges of operating in a multi-object environment. In order to achieve a tracking system that completes a target profile with sensing characteristics, the challenge of sensing multiple objects and associating the acquired information to a detected target must be solved.

2.4.3 Identification Strategies

The development of identification methodologies is a natural direction for the evolution of mmWave tracking systems. It can be considered a more unique type of specialised estimation sensing but with the key focus on being able to reliably and uniquely correlate the sensed information to a tracked object. There are a number of challenges that need to be considered and overcome in identification approaches, such as the feasible range, separation of multiple objects, and generalisation of the

Table 2.2: A comparison of sensing methodologies explored for the enhancement of tracking reliability in an mmWave tracking architecture.

Criterion	**Generalised Activity Recognition**	**Specialised Estimation**
Adaptability	✓ Decoupled architecture impact. × Uncertain tracking enhancement reliability.	✓ Trusted point cloud processing techniques. × Uncertain feedback enhancement reliability.
Performance	✓ Algorithm real-time performance proven. × Uncertain system suitability.	✓ Real-time suitability has been proven viable. × Optimisation overhead to accommodate.
Accuracy	✓ High pre-defined activity accuracy. × Accuracy dependent on training environment.	✓ Accuracy high due to the narrow focus. × Highly coupled to the training data.
Specificity	✓ Pre-defined actions reliably classified. × Uncertainty of multi-object suitability. × Simultaneous classification challenging.	✓ Optimised for estimating a single action. × One target is considered for estimation. × Underdeveloped literature in mmWave field.

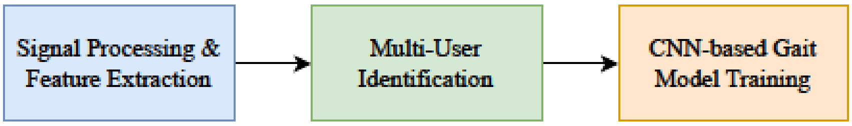

Signal Processing & Feature Extraction
Multi-User Identification
CNN-based Gait Model Training

Figure 2.9: System breakdown of the gait identification methodology presented in [66].

approach. This section aims to explore the leading identification methodologies of radar-based tracking systems.

Gait identification approaches rely on the different gait characteristics between individuals. Gait based identification strategies are the most common passive based approach to identifying people in a radar or Wi-Fi based tracking system. They fundamentally leverage that each person typically has a unique pattern in the way they walk, this pattern is most often identified through a deep learning based technique. Gait recognition can pose its own challenges, such as inconsistencies and unpredictable upper limb movements that influence the lower limb signal reflections. These interferences ultimately reduce the reliability of obtaining a consistent lower limb gait pattern for a given individual. A recent study performed in [66] attempts to overcome the challenges associated with upper limb movement interference by narrowing the vertical field of view and focusing attention on the finer grain movements of the lower limbs. The research presented in [66] proposes a system that consists of three phases, illustrated in Figure 2.9.

In the first phase, the authors of [66] construct a range-Doppler map following the traditional methodology described in Section 2.3 of this chapter. The stationary interference in the range-Doppler map is then removed following a technique similar to the described approach in Section 2.3.3 of this chapter. The stationary reflections are subtracted from each frame of the range-Doppler frequency responses. The authors of [66] observe that a cumulative deviation of the range-Doppler data occurs due to the dynamic background noises, which are not eliminated when subtracting the static interference. To overcome this, a threshold-based high-pass filter is implemented with a threshold τ of $10dBFS$. The filtering operation is mathematically defined as:

$$R_{(i,j,k)} = \begin{cases} R_{(i,j,k)}, & R_{(i,j,k)} \geq \tau, \\ 0, & R_{(i,j,k)} < \tau, \end{cases} \tag{2.16}$$

where $R_{(i,j,k)}$ is the range-Doppler domain frequency response at the k_{th} frame with range i and velocity j.

The authors of [66] identify that the dominant velocity $\hat{V}_i$ can be used to describe the targets lower limb velocity in each frame. The calculation of the dominant velocity is given by:

$$\hat{V}_i = \frac{\sum_{j=1}^{N_D} \left(\hat{R}_{(i,j,k)} V_j \right)}{N_D}, i \in [1, N_R], j \in [1, N_D], \tag{2.17}$$

where $\hat{R}_{(i,j,k)}$ is the normalised frequency response, V_j is the velocity corresponding to the frequency response $R_{(i,j,k)}$, N_R and N_D represent the number of range-FFT and Doppler-FFT points respectively.

The authors of [66] illustrate the composition of these gait characteristics as a heatmap corresponding to the actual gait captured with a camera. Using these extracted gait features, the author of [66] identifies that multiple targets can be differentiated firstly by range and secondly (if the range is the same) by leveraging distinct spatial positions. This is ultimately done by projecting the point $R_{(i,j,k)}$ in the k_th frame to a point $\hat{R}_{(i,j,k)}$ in the two-dimensional spatial Cartesian coordinate system. To differentiate the data points in the spatial Cartesian coordinate system, [66] implements a K-means clustering algorithm. Each individual gait feature can be generated as a range-Doppler map by negating the frequency responses that were not correlated in the K-means clustering [66]. After differentiating the gait features, the authors of [66] then identify a challenge regarding the segmentation of the actual step. This is ultimately overcome by using an unsupervised learning technique to detect the silhouette of the steps [66].

Finally, a CNN-based classifier in the image recognition domain is used to identify the patterns associated with the gait feature maps. The classifier is assessed with multiple users and varying steps to determine the overall accuracy of the system. Overall, the system demonstrates a high accuracy that marginally decreases in accuracy as the number of users increases but is ultimately corrected as the number of steps increases.

Another overarching class of identification strategies being explored are tagging based approaches. This is not a passive approach, unlike the others mentioned in this chapter, and involves incorporating a tag on the object so that it can be uniquely identified. There are two directions in which the literature focuses on in regard to identification of this nature. The first is RFID. In a chipless based RFID system, data must be encoded in the signal either by altering the time-domain, frequency-domain, spatial-domain or a combination of two or more of the domains. An example of RFID implemented as an identification strategy in mmWave can be seen in the FerroTag research presented in [67]. The FerroTag system presented in [67] is a paper-based RFID system. Although the usage of the FerroTag research is intended for inventory management, it could potentially be adopted as a tagging strategy for a tracking based system. FerroTag is ultimately based on ferrofluidic ink, which is a collodial liquid that fundamentally contains magnetic nanoparticles. The ferrofluidic ink can be printed onto surfaces which in turn will result in embedded frequency characteristics in the response of a signal. The shape, arrangement, and size of the printed ferrofluidic ink will ultimately influence the frequency tones that are applied to the response signal. In order to identify and differentiate the different signal characteristics caused by the chipless RFID surface, the solution presented by [67] utilises a random forest as a classifier to identify the corresponding tags present in the field of view. The second approach to tagging as a means of identification is through RIS. To the best of our knowledge no system has been presented in the literature that demonstrates a practical RIS solution for identification purposes in a mmWave tracking system. Research regarding RIS with respect to mmWave is predominantly in the communication domain. The challenges and opportunity to design a RIS based identification system for a mmWave tracking system are yet to be detailed.

Shape profiling has been implemented in previous mmWave research to identify an object by the properties of the object's shape. For example, if the object being tracked is a human, the height and curvature of the human body can influence the way in which the mmWave signal is reflected [68]. The authors of [68] demonstrate how a human being tracked and represented in point cloud form can be identified based on the shape profile of their body. Using a fixed-size tracking window, the related points to the particular human are voxelised to form an occupancy grid [68]. This is then sequenced through a LSTM network to classify the particular human [68]. This particular identification method is abstracted from the tracking aspect of the process, therefore making it suitable regardless if there are multiple objects being tracked.

The research presented in [2] differs to that presented in [68] in the regard that the tracking data is not used during the identification stage. Instead, the authors in [2] propose a strategy where once the human has been tracked, the radar steers its transmit and receive beams towards the tracked human. By doing so, the granularity of the feature set available from the human body is increased. In other words, more specific profiling can be performed on the individual. The research presented in [2] demonstrates the ability to characterise the human body by its outline, surface boundary, and vital signs. Having this granular feature set, and tailored profiling, provides a stronger ground to positively identify an individual. However, this particular method does come at the cost of beam steering for identification purposes. Additionally, the existing research presented in [2] does not make any remarks regarding the suitability for this method in real-time applications.

The various identification strategies explored in this section of the chapter each have their own complexities involved in fundamentally incorporating into a tracking system. Table 2.3 aims to assist in comparing the various identification methodologies, to ultimately understand their suitability and limitations around implementing them in a tracking system.

2.5 Review Summary

This literature review has provided a comprehensive analysis of the current state-of-the-art in mmWave radar multi-object tracking and sensing. We began by outlining the historical context and foundational theories of multi-object tracking, highlighting the significant contributions of Kalman filters, particle filters, and data association techniques such as the JPDA algorithm. These early developments laid the groundwork for the sophisticated tracking systems in use today.

Table 2.3: A comparison of identification methodologies explored for the enhancement of tracking objects discontinuously in an mmWave tracking architecture.

Criterion	**Gait**	**Tagging**	**Shape Profile**
Adaptability	✓ Low architecture impact. × Ability to correlate to multiple tracks unknown. × Specific hardware positioning.	✓ Loosely coupled to tracking architecture. × No common data plane. × Additional hardware. × Challenging multi-object correlation.	✓ Potential to extend on point cloud. × Sampling concerns with simultaneous beam steering and tracking.
Performance	✓ Proven real-time viability. × Computational overhead.	✓ Very minimal impact. ✓ Pre-encoded data absorbs impact. × Uncertainty in multi-object setting.	✓ Minimal overhead. × Suitability unproven.
Accuracy	✓ High multi-object accuracy. × Scalability challenges.	✓ Very accurate. × Immature understanding on range.	✓ No impact due to multi-object. × Environmental dependencies.
Specificity	✓ Focussed movement considerations. × Challenges with wider field of view.	✓ Low risk of false positives. × Undefined challenges with multi-object.	✓ Multi-objects independently profiled. × Immature understanding on environmental impacts.

Recent advances in multi-object tracking, particularly in the field of computer vision, have been driven by the adoption of deep learning techniques. CNNs and RNNs have improved object detection, feature extraction, and tracking persistence, demonstrating reliability in complex scenarios. The integration of sensor fusion techniques, combining data from multiple sensors like cameras, LiDAR, and mmWave radars, has further enhanced tracking performance and system reliability in diverse environmental conditions.

The chapter also explored the specific challenges and methodologies associated with mmWave radar multi-object tracking. Continuous and discontinuous tracking were defined, with discontinuous tracking being particularly relevant for applications requiring the re-identification of targets that temporarily leave the radar's field of view. The potential applications of sophisticated mmWave tracking systems were discussed, including security, healthcare, and public space analytics.

A typical mmWave tracking system architecture was presented, detailing the components involved from radar data collection to position and velocity estimation, association, tracking, sensing, and identification. Advanced methodologies were explored to address the limitations of traditional tracking systems, such as sensor fusion and micro-Doppler feature analysis for object detection, and specialised estimation techniques for sensing.

Finally, the chapter reviewed various identification strategies, including gait recognition, tagging, and shape profiling, each with its own set of challenges and advantages. The comparison of these methodologies highlighted the complexities involved in integrating identification strategies into a tracking system and the potential for enhancing tracking reliability and sensitivity.

In conclusion, the advancements in mmWave radar multi-object tracking and sensing systems have shown significant potential for various applications. However, challenges remain in achieving robust and reliable tracking, particularly in complex and dynamic environments.

Chapter 3

Multi-Object Regional Trajectory Analysis

This chapter presents a detailed exploration of the proposed framework for enhancing multi-object tracking through environmental characterisation using mmWave radar. The aim of this chapter is to establish the foundational concepts and methodologies that underpin the research presented. The chapter is structured as follows: Section 3.1 introduces the motivation and objectives of the research, highlighting the challenges and the need for environmental characterisation. Section 3.2 describes the architecture of the RDTP multi-object tracking system, outlining the key stages involved. Section 3.3 details the methodology and implementation of each stage, including trajectory collection, pre-processing, RDTP analysis, and environmental association. Section 3.4 presents the experimental results and analysis, demonstrating the effectiveness of the proposed framework. Finally, Section 3.5 provides concluding remarks and discusses future directions for research in this area.

3.1 Introduction

Understanding the characteristics of an environment can significantly enhance the ability to perform more accurate multi-object tracking. In dynamic and complex environments, traditional multi-object tracking systems often struggle with occlusions and disturbances caused by non-penetrable objects such as walls, columns, or furniture. These challenges lead to frequent loss of tracks, re-identification errors, and overall degradation in tracking performance. To address these issues, a foundational framework must be established to extract and utilise environmental characteristics from the observed area.

In this research, we propose an approach to extract entry and exit points for an environment using multi-object tracking trajectories. The observed environment is divided into a grid, where regions are defined. The trajectories obtained from the multi-object tracking data are organised into their respective regions. For each region, an activity heatmap is formed and classified using a CNN to determine if the region consists of either an entry or exit point. The classified entry and exit points are then projected onto the multi-object tracking plane to illustrate the entry and exit points of the observed environment. This approach provides a foundation for future work to enhance multi-object tracking capability in real-time through a greater understanding of the observed environment.

A movement pattern in the context of this chapter can ultimately be explained as a commonality of transitions between different start and end states. Regional dominant movement patterns can be defined as the most frequently occurring transitional pattern for a given start and end state. The study of regional dominant movement patterns in trajectory data has been well explored. A study performed by [69] demonstrates the potential to determine regional dominant movement patterns in trajectory data using a pre-defined taxonomy of trajectory patterns. The authors in [69] define different types of movement patterns, which are then used as criteria for a trained CNN to classify trajectory clusters. Another study performed in [70] demonstrates a novel approach for mining trajectory patterns, as opposed to utilising a taxonomy of pre-defined patterns. Mining trajectory patterns is a necessary approach to take in situations where the types of trajectory pattern that might occur is not known. In situations where particular trajectory patterns are known or searched for, maintaining a taxonomy of those patterns can yield more performant results.

Applying the concept of regional dominant movement patterns to trajectory data, termed as RDTPs, will allow for the identification of recurring movements within the field of view in which the trajectory data was collected. These movement patterns fundamentally expose characteristics of the environment that is being sampled in the field of view. By understanding these patterns, it becomes possible to predict object trajectories more accurately and adapt multi-object tracking systems to varying conditions.

The proposed framework involves several stages, starting with the collection of multi-object tracking trajectory data using a mmWave radar sensor. The collected data is then pre-processed and normalised to form regional activity heatmaps. These heatmaps are analysed using a CNN to classify entry and exit points. The classified points are then projected onto the multi-object tracking plane, providing a visual representation of the environmental characteristics. This systematic approach aims to improve the robustness and accuracy of multi-object tracking systems by leveraging the extracted environmental characteristics.

This research presents a novel approach to enhance multi-object tracking performance by systematically deriving environmental characteristics from trajectory data. The proposed framework not only addresses the challenges posed by occlusions and disturbances but also provides a foundation for future advancements in multi-object tracking technologies. By integrating environmental understanding into multi-object tracking systems, we can achieve more reliable tracking, reduced errors, and better handling of complex and dynamic environments.

3.2 RDTP Multi-Object Tracking Architecture

The architecture adopted for the RDTP multi-object tracking environmental characterisation explored in this chapter can be broken into 5 stages. The sequencing of the 5 stages can be seen in Figure 3.1. The methodology and implementation of these stages will be explored in Section 3.3 of this chapter. The remainder of this section of the chapter will describe the purpose of each of the stages depicted in Figure 3.1 in relation to the problem statement.

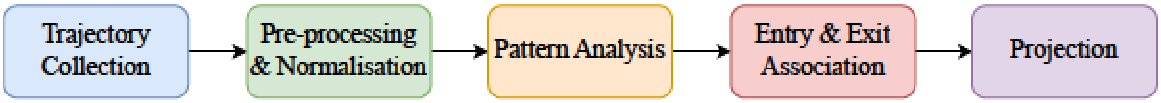


Figure 3.1: Block diagram of overall stages involved in the RDTP multi-object tracking architecture.

3.2.1 Problem Statement

Multi-object tracking in dynamic environments is inherently challenging due to the presence of occlusions and disturbances caused by non-penetrable objects such as walls, columns, or furniture [71]. Traditional systems often lack the capability to adapt to these complexities, necessitating a framework that can systematically derive and utilise environmental characteristics to enhance tracking system accuracy.

In more complex environments, the ability to understand and characterise the spatial layout and the dynamic interactions within the environment is crucial for enhancing the robustness and accuracy of multi-object tracking systems. Without a foundational framework to extract and utilise environmental characteristics, multi-object tracking systems are limited in their capacity to adapt to varying conditions and to predict object trajectories accurately.

The primary problem addressed in this research is the development of a systematic approach to derive environmental characteristics from multi-object tracking trajectory data. This involves identifying key features of the environment, such as entry and exit points, and understanding the movement patterns within the observed area. By establishing a method to classify and project these environmental characteristics onto the multi-object tracking plane, the proposed approach aims to provide a foundational basis for improving multi-object tracking performance in real-time applications.

3.2.2 Proposed Framework

The block diagram in Figure 3.1 illustrates the proposed architecture for deriving environmental characteristics from multi-object tracking trajectory data. The architecture is divided into five key stages: Trajectory Collection, Pre-processing and Normalisation, Trajectory Analysis, Entry and Exit Association, and Projection.

The *Trajectory Collection* stage involves using a mmWave radar to collect raw data, which is then processed to track multiple objects. This stage generates a vector of tracked objects in the form of trajectories, providing the foundational data required for subsequent analysis. The process includes transmitting chirp signals, receiving reflected signals, and calculating the range, velocity, and AoA of detected objects. The collected data is then organised into a point cloud, with static objects removed to focus on moving objects. Clustering and association algorithms are applied to track the movement of objects over time, resulting in a comprehensive trajectory dataset.

In the *Pre-processing and Normalisation* stage, the collected trajectory data is then organised into regional trajectories. The observed environment is divided into a grid, and tracked objects are assigned to their respective regions. This stage involves cleaning the data to remove noise and outliers, interpolating missing data, and normalising the coordinates. The regional trajectories are then transformed into activity heatmaps, which then serve as input for the subsequent analysis. The heatmaps are generated by mapping the normalised coordinates of tracked objects to the cells within each region, with Gaussian smoothing applied to enhance quality.

The *Trajectory Analysis* stage leverages a CNN to classify the regional trajectories into predefined patterns. The CNN is trained to recognise entry and exit patterns based on a taxonomy of movement patterns. The network consists of convolutional and pooling layers to extract features from the activity heatmaps, followed by a softmax layer to classify the patterns. The classifier outputs a probability distribution over the classes, identifying regions as entry or exit points based on the highest probability. Data augmentation and dropout layers are used during training to improve network reliability and prevent overfitting.

The *Entry and Exit Association* stage involves correlating and grouping regions identified as entry and exit points. This stage maps the classified regions back to their original spatial coordinates and defines rectangular bounds around the entry and exit points. The bounding process identifies the minimum and maximum coordinates of tracked objects within each region, creating a clear representation of the environmental features. This stage helps in understanding the spatial layout and identifying non-penetrable objects that can cause occlusions.

Finally, the *Projection* stage visualises the identified environmental characteristics on the multi-object tracking plane. The rectangular bounds of entry and exit points are overlaid onto the visual representation of the environment, providing a comprehensive view of the dynamic interactions within the environment. This visualisation aids in improving the overall performance of the multi-object tracking system by integrating environmental understanding into the tracking process.

The proposed framework systematically processes multi-object tracking trajectory data to derive environmental characteristics, enhancing the capability of multi-object tracking systems to handle complex and dynamic environments. By integrating environmental understanding into the tracking process, the framework aims to achieve more reliable tracking, reduced errors, and better handling of occlusions.

3.3 Methodology and Implementation

To implement the RDTP multi-object tracking architecture, a TI IWR6843 mmWave sensor was used, as seen in Figure 3.2. The mmWave sensor was installed on a TI MMWAVEICBOOST evaluation board, in which raw ADC data from the sensor was streamed via a TI DCA1000EVM. The sensor was mounted on a tripod and positioned approximately 1.5m above ground. The sensor was also tilted downward at a 10° angle to achieve the best field of view. This position is in accordance with the recommendation provided by TI. Figure 3.3 illustrates the positioning of the sensor in relation to the external environment. The remainder of this section describes the methodology adopted for each of the RDTP multi-object tracking stages, seen in Figure 3.1.

3.3.1 Multi-Object Tracking Trajectory Collection

The process of collecting multi-object tracking trajectories begins with the transmission of a 'chirp' signal from the mmWave radar. This chirp signal is reflected off objects in the environment and received back by the radar. The difference in frequency between the transmitted and received signals is known as the IF signal,

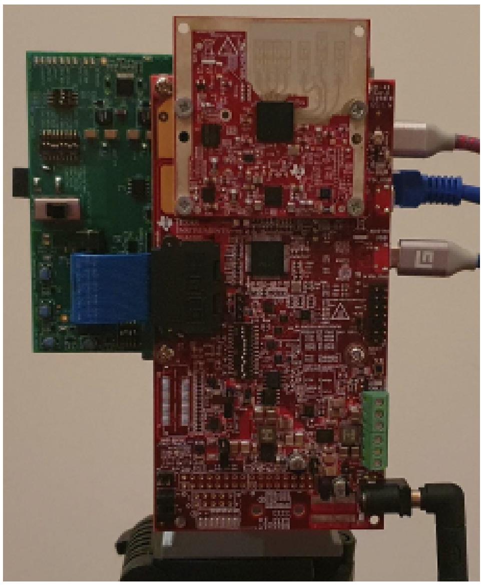


Figure 3.2: TI mmWave IWR6843ISK with MMWAVEICBOOST and DCA1000EVM.

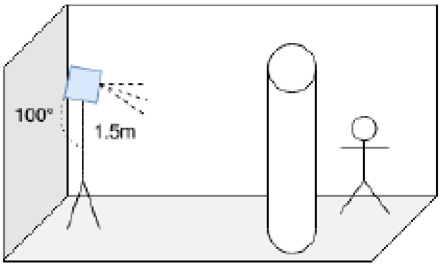


Figure 3.3: Positioning of the mmWave sensor for RDTP.

as discussed in Section 2.3.2 of this thesis. This IF signal is crucial for determining the range, velocity, and AoA of the detected objects.

Firstly, the range R_x from the radar to the detected object x can be computed using the IF signal frequency f_{IF} and the frequency slope S of the transmitted chirp. Recall, this relationship was presented in Section 2.3.2, Equation 2.12. This equation is derived from the fact that the frequency difference f_{IF} is proportional to the time delay of the reflected signal, which in turn is proportional to the distance R_x.

Next, the velocity V_x of the detected objects is calculated by analysing the phase shift in the IF signal between two consecutive chirps. The relationship between the velocity and the phase difference is expressed as:

$$V_x = \frac{\lambda w}{4\pi T_c}, \tag{3.1}$$

where λ is the wavelength of the transmitted signal, w is the phase difference between the received signals of the two chirps, and T_c is the time interval between consecutive chirps. This equation is based on the Doppler effect, where the phase shift w is directly related to the velocity of the moving object.

The AoA θ_x of the detected object is estimated by averaging the phase differences across multiple transmitter-receiver pairs. Recall, for a single pair, the phase difference θ_x can be calculated using Equation 2.14, from Section 2.3.2. This equation uses the principle of phase difference to determine the angle at which the signal arrives at the receiver.

Using the range R_x, velocity V_x, and AoA θ_x calculated using Equations 2.12, 3.1, and 2.14, respectively, a point cloud graph is constructed. This graph represents the positions of detected objects in the environment. Static objects, which do not change position between frames, are identified and removed from the point cloud to focus on moving objects.

The next step involves clustering the point cloud data using the DBSCAN algorithm. DBSCAN identifies clusters of points that represent individual objects. The computational complexity of DBSCAN is $\mathcal{O}(N_{points} \log N_{points})$ for N_{points} point cloud points. These clusters are then associated across consecutive frames using the Hungarian Algorithm, which matches clusters from the current frame F_n to those from the previous frame F_{n-1}. The Hungarian algorithm has a computational complexity of $\mathcal{O}(N^3_{objects})$ where $N_{objects}$ is the number of clusters to be matched. In typical multi-object tracking scenarios, $N_{objects}$ represents the number of detected objects and is generally small, making the algorithm computationally tractable. This association helps in tracking the movement of objects over time.

To improve the accuracy of the tracking, a Kalman filter is applied. The Kalman filter predicts the future position of each tracked object and corrects the prediction based on the actual measurements, as discussed in Chapter 2. The computational complexity of the Kalman filter is $\mathcal{O}(d^3)$ per tracked object, where d is the state vector dimension. For $N_{objects}$ tracked objects, the total complexity becomes $\mathcal{O}(N_{objects}d^3)$ per frame. The dominant operations are matrix inversions and multiplications in the Kalman gain computation and covariance updates. This results in a more stable and accurate tracking of objects.

A single tracked object i is represented as a tuple structure, defined by the following expression:

$$TO_i = (p, v), \tag{3.2}$$

where p is a two-dimensional vector $[x, y]$ representing the current coordinates of the tracked object, and v is the current velocity vector of the tracked object.

For a given frame j, the set of tracked objects present in that frame is represented as:

$$FTO_j = \{TO_1, TO_2, \ldots, TO_{N_{objects}}\}, \tag{3.3}$$

where each TO is a tuple as defined in Equation 3.2, and $N_{objects}$ is the total number of tracked objects in frame j.

Over multiple frames, the sets of tracked objects are persisted, forming a comprehensive trajectory dataset expressed mathematically as:

$$TTO = \{FTO_1, FTO_2, \ldots, FTO_{N_{frames}}\}, \tag{3.4}$$

where each FTO represents the set of tracked objects in a single frame, and N_{frames} is the total number of frames. This dataset TTO provides a comprehensive record of the movement of objects over time, which is essential for further analysis and environmental characterisation.

3.3.2 Pre-processing and Normalisation

To prepare for *RDTP Analysis*, a grid $G = (i \times j)$ must be constructed for a field of view of size $(l \times w)$. The terms l and w are expressed in meters, whilst i and j are the number of cells that l and w should be split into. Regions are defined as a collection of adjacent cells from G that form $R_x = (m \times n)$, where x is the region number. Region dimension m and n must both be factors of i and j respectively. Regions must not overlap with each other and must fit into G perfectly.

For each region R_x, an activity heatmap of R_x is required to perform *RDTP Analysis*. In order to construct a heatmap for each R_x, the vector represented in Equation 3.4 will need to be transformed so that the tracked objects TO are grouped into their respective region. A given TO is correlated with an R_x through the coordinates it occupies, present in Equation 3.2. The related FTO that the tracked object took place at must not be lost when transforming the vector, this information will be required when performing *Environmental Association and Bounding*. The set of regional tracked objects is expressed as a collection of tuples, defined as:

$$RTO_x = \{(F, TO)_1, (F, TO)_2, ..., (F, TO)_n\}, \tag{3.5}$$

where F is the frame the coupled TO occurred at, n is the total number of tracked objects, and x is the region index.

The activity heatmap image can then be constructed for each RTO_x expressed by Equation 3.5. Each TO equates to activity in a region, activity is positioned at the coordinates of the respective TO. The activity heatmap is illustrated as a fixed-sized image, stored in a set, mathematically expressed as:

$$RHM = \{HM_1, HM_2, ..., HM_n\}, \tag{3.6}$$

where HM is the activity heatmap constructed for the respective RTO_x expressed in Equation 3.5 and n is the total number of regions in G. The tracked regional trajectory objects RTO_x are ultimately normalised through their illustration as an activity heatmap.

To ensure the accuracy and consistency of the activity heatmaps, several pre-processing steps are necessary. First, the raw trajectory data must be cleaned to remove any noise or outliers that could distort the heatmap. This involves filtering out any tracked objects that exhibit erratic or implausible movements, which are likely to be artefacts of the tracking process rather than genuine object movements.

Next, the trajectory data is interpolated to fill in any gaps where objects were temporarily lost due to occlusions or other tracking failures. This interpolation is performed using a linear or spline method, depending on the nature of the missing data. By filling in these gaps, the resulting heatmaps will more accurately reflect the continuous movement of objects within the environment.

Once the data is cleaned and interpolated, it is then normalised to ensure that all trajectories are represented on the same scale. This involves scaling the coordinates of the tracked objects to fit within the dimensions of the grid G. The normalisation process ensures that the activity heatmaps are consistent in resolution, which is crucial for accurate analysis by the CNN.

After normalisation, the trajectory data is divided into individual regions based on the grid G. Each region R_x is processed separately to generate its corresponding activity heatmap. This involves mapping the normalised coordinates of the tracked objects to the cells within the region and incrementing the activity count for each cell. The resulting heatmap is a two-dimensional array where each cell value represents the level of activity in that cell.

To enhance the quality of the heatmaps, a Gaussian smoothing filter is applied. This filter helps to reduce noise and create a more continuous representation of the activity within each region.

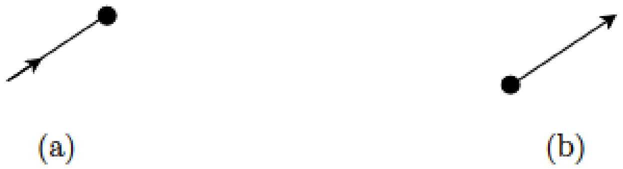
Finally, the pre-processed and normalised activity heatmaps are stored in a set RHM as described in Equation 3.6. These heatmaps serve as the input for the *RDTP Analysis* stage, where they will be analysed by the CNN to classify entry and exit points. The detailed pre-processing steps ensure that the activity heatmaps are accurate and consistent, providing a solid foundation for the subsequent analysis and environmental characterisation.

3.3.3 RDTP Analysis

The purpose of the *RDTP Analysis* is to extract entry and exit points from the regional trajectories formed by the tracked objects. In order to do so, an idealistic taxonomy of regional patterns that expose entry and exits should be pre-defined.

Figure 3.4 defines the taxonomy utilised to classify entry and exit trajectories. The trajectory used to classify an entry movement pattern is illustrated in Figure 3.4 (a), whilst Figure 3.4 (b) demonstrates the trajectory used to classify an exit movement pattern.

A single classifier is used to determine the probability that a given activity heatmap,



(a) (b)

Figure 3.4: Trajectory taxonomy.

expressed in the set illustrated in Equation 3.6, is one of the movement patterns expressed in Figure 3.4. The classifier is ultimately based on the handwritten digit recognition network presented in [72].

The input shape of the network is a 32×32 pixel activity heatmap. A convolutional and maximum pooling layer is added to extract features from the activity heatmap. The convolutional layer consists of 16 filters of size 5×5. Another convolutional and maximum pooling layer is added to the network with 32 filters of size 3×3. Finally, a softmax layer of size 3×1 is added. The 3 softmax classes (C_1, C_2 and C_3) are the two movement patterns defined in Figure 3.4, respectively C_1 and C_2, and the probability that neither C_1 nor C_2 are correct, respectively C_3. The loss function utilised to correct weights is the Mean Squared Error (MSE) loss.

For input heatmaps of size $H \times W$ with dimensions 32×32 pixels, the computational complexity of the CNN during inference is $\mathcal{O}(F_1 K_1^2 HW + F_2 K_2^2 H'W' + F_1 F_2 + F_2 \times 3)$, where F_1 and F_2 are the number of filters in the first and second convolutional layers with values of 16 and 32 respectively, K_1 and K_2 are the kernel sizes of 5×5 and 3×3 respectively, and $H'W'$ represents the reduced spatial dimensions after pooling. For the fixed input size of 32×32 pixels, this evaluates to approximately 4×10^5 operations per classification.

The training process for the classifier involves feeding the network with labelled activity heatmaps, where each heatmap is associated with one of the three classes (C_1, C_2, or C_3). The network identifies the distinguishing features of entry and exit patterns through backpropagation and gradient descent. During training, the weights of the convolutional layers are adjusted to minimise the loss function, thereby improving the accuracy of the classifier.

To ensure robustness and avoid overfitting, the training dataset is augmented with various transformations such as rotations, translations, and scaling. This augmentation helps the network generalise better to different scenarios and variations in the trajectory data. Additionally, dropout layers are incorporated into the network to further prevent overfitting by randomly deactivating a fraction of the neurons during training.

Once the classifier is trained, it is used to analyse the activity heatmaps generated from the regional trajectories. For each heatmap, the classifier outputs a probability distribution over the three classes. The class with the highest probability is selected as the predicted label for the heatmap. If the predicted label is C_1 or C_2, the corresponding region is classified as an entry or exit point, respectively. If the predicted label is C_3, the region is considered neither an entry nor exit point.

The classified entry and exit points are then used to enhance the understanding of the environment. By identifying the regions where objects frequently enter or exit, the system can infer the locations of doors, windows, or other significant features in the environment. This information is crucial for improving the performance of multi-object tracking systems, as it allows for better handling of occlusions and more accurate prediction of object trajectories.

3.3.4 Environmental Association and Bounding

The classified regional activity heatmaps are organised into those that are deemed either an entry or exit trajectory pattern and those that are not. The regional activity heatmaps that are not an entry or exit trajectory pattern are negated going forward. The remaining regional activity maps are correlated back to the individual tracked objects that constitute the region, illustrated in Equation 3.5. In this state, the regions that have been classified as entry and exit points of the field of view are projected onto the multi-object tracking plane and rectangularly bounded.

To achieve this, each classified region R_x is mapped back to its corresponding coordinates in the original field of view. This involves translating the grid coordinates of the region to the actual spatial coordinates in the environment. The bounding process is performed by identifying the minimum and maximum coordinates of the tracked objects within each region. These coordinates define the rectangular bounds that encapsulate the entry or exit points.

The bounding box for a region R_x can be defined by determining the minimum and maximum x-coordinates ($x_{\min}$ and $x_{\max}$) and y-coordinates ($y_{\min}$ and $y_{\max}$) of tracked objects within that region, expressed as:

$$x_{\min} = \min(p_x), \quad x_{\max} = \max(p_x), \tag{3.7}$$

$$y_{\min} = \min(p_y), \quad y_{\max} = \max(p_y), \tag{3.8}$$

where p_x and p_y are the x and y coordinates of the tracked objects within the region R_x. The bounding box is then represented as a rectangle with these minimum and maximum coordinates.

Once the bounding boxes are determined, they are projected onto the multi-object tracking plane. This projection involves overlaying the rectangular bounds onto the visual representation of the environment, providing a clear indication of the entry and exit points. The projection helps in visualising the spatial layout of the environment and understanding the movement patterns of objects.

The final step involves validating the projected entry and exit points against the actual layout of the environment. This validation is performed by comparing the projected points with known entry and exit locations, such as doors and windows. Any discrepancies are analysed and corrected to ensure the accuracy of the environmental characterisation.

3.4 Experimental Results and Analysis

The methodology and implementation discussed in this chapter were trained and tested using real data. A dataset of 1100 entry and exit events was collected using the multi-object tracking trajectory collection process described in the previous section of this chapter. These events were collected across 5 different environments.

The dataset was then pre-processed and transformed into a collection of activity heatmaps, as described in the previous section of this chapter. The activity heatmaps were then manually tagged for identification of entry and exit events. The training set of data occupied 80% of the tagged dataset. The remaining 20% was

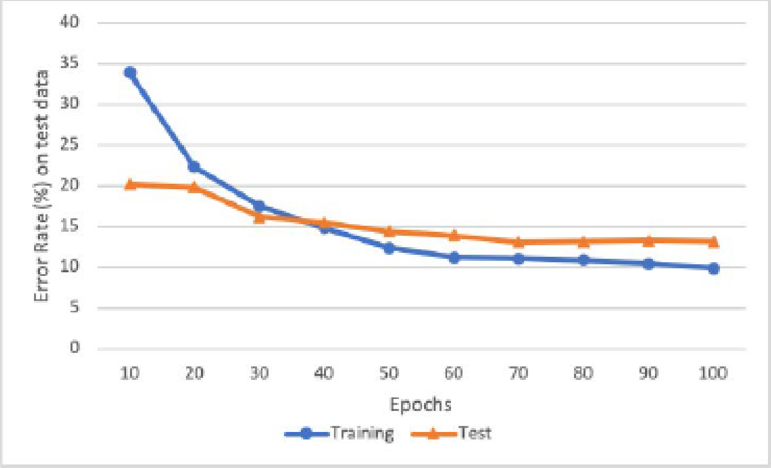


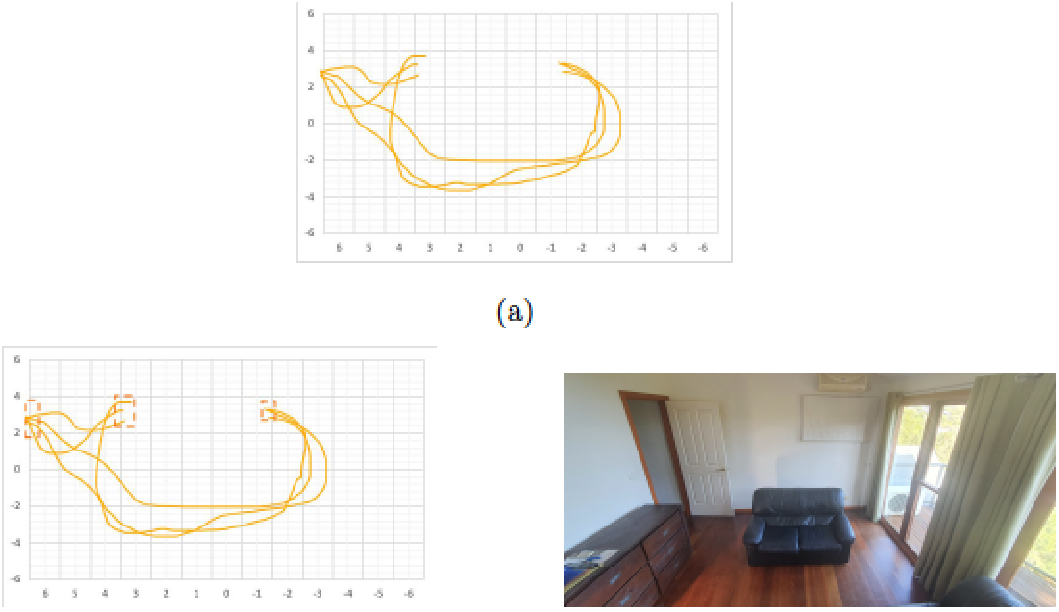
Figure 3.5: Training and test error rate percentage across epochs.

reserved for testing. After performing 65 epochs, taking 102.39 seconds to complete training, the average accuracy of the network was 87.18%. Figure 3.5 demonstrates the change in error over the number of iterations performed. A value of 65 epochs was deemed appropriate due to the notable convergence evident in Figure 3.5.

The classified entry and exit points are projected onto the multi-object tracking plane, demonstrated in Figure 3.6 (a) and (b). The projected entry and exit points in Figure 3.6 (b) can be compared to the image of the observed environment illustrated in Figure 3.6 (c). During multi-object tracking in this environment, at most two individuals were present in the field of view. The individuals independently walked around in the field of view, as well as leaving and re-entering through the door on the left side of the room and ducking behind the couch in the centre of the room. It is evident that the trained network successfully classified these 4 entry and exit events through the projections presented in Figure 3.6 (b).

To further analyse the performance of the proposed framework, additional metrics such as precision, recall, and F1-score were calculated. These metrics provide a more comprehensive evaluation of the classifier's performance. Precision measures the proportion of true positive classifications among all positive classifications, while recall measures the proportion of true positive classifications among all actual positive instances. The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both aspects.

The overall precision, recall, and F1-score for both the entry and exit classification



(a)

(b)

(c)

Figure 3.6: RDTP multi-object tracking plot with projections and scene.

tasks can be seen in Figure 3.7. These metrics indicate that the classifier performs well in identifying entry and exit points, with a high level of accuracy and balanced precision and recall. The average F1-score of 87.1% further confirms the ability of the classifier to handle variability in the trajectory data.

Additionally, the impact of different environmental conditions on the classifier's performance was analysed. The environments varied in terms of layout complexity, the presence of occlusions, and the number of moving objects. The classifier's performance was consistent across the different environments, with only minor variations in accuracy. This consistency demonstrates the generalisability of the proposed framework and its applicability to various real-world scenarios.

3.5 Conclusion

In conclusion, this chapter presented a novel framework for enhancing mmWave multi-object tracking systems by systematically deriving environmental characteristics from trajectory data. The proposed approach addresses the challenges posed by occlusions and disturbances in dynamic environments, which often degrade the performance of traditional multi-object tracking systems. By leveraging a mmWave

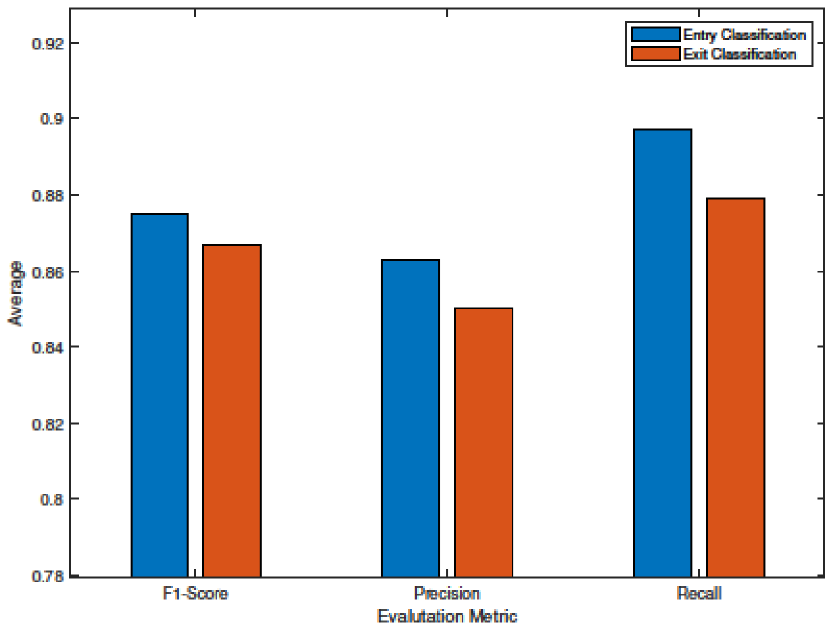


Figure 3.7: Precision, recall, and F1-score for entry and exit classification.

radar sensor for trajectory collection, pre-processing and normalising the data into regional activity heatmaps, and utilising a CNN for RDTP analysis, the framework effectively identifies entry and exit points within the observed environment.

The methodology and implementation were validated through extensive experiments, demonstrating high accuracy and robustness in classifying entry and exit points. The experimental results showed an average accuracy of 87.18% after 65 epochs of training, with precision, recall, and F1-score metrics further confirming the classifier's performance.

By integrating environmental understanding into the multi-object tracking process, the proposed framework provides a foundation for future advancements in multi-object tracking technologies. The enhanced capability to handle complex and dynamic environments leads to more reliable tracking, reduced errors, and better handling of occlusions. This research contributes to the advancement of multi-object tracking systems, by providing a means for improved performance in various applications, including surveillance, autonomous navigation, and human-robot interaction.

Chapter 4

Combined mmWave Tracking and Classification Framework Using Camera for Labelling and Supervised Learning

This chapter presents a novel framework for combining mmWave radar and camera data to enhance tracking and classification capabilities. The proposed approach leverages the strengths of both sensor modalities to address the challenges associated with labelling and training deep learning models against radar data. By fusing radar and camera data, we aim to create a system that can accurately classify and track objects in various environments. The chapter is structured as follows: Section 4.1 introduces the problem space and the motivation behind using sensor fusion for labelling and training radar data. Section 4.2 reviews existing sensor fusion architectures and methodologies, highlighting the challenges and solutions presented in related literature. Section 4.3 details the proposed methodology for radar training with camera labelling and supervision, outlining the steps involved in data collection, correlation, and training. Section 4.4 describes the system design and implementation of the proposed framework, providing a practical example of its application. Section 4.5 presents the experimental results and analysis, demonstrating the effectiveness of the framework. Finally, Section 4.6 concludes the chapter with a summary of the contribution and the potential avenues that can be explored to further progress the research.

4.1 Introduction

The process of training mmWave sensors to solve classification problems is rapidly becoming more popular and proving to be a promising direction in radar sensing research. One of the most promising techniques that is being pursued in this field of research is a deep learning based approach. However, successfully using a deep learning based approach typically requires an abundant set of training data to adequately teach a model the relevant features that can be relied on for classification and/or prediction. Constructing a large and meaningful dataset requires a domain expert to spend the time appropriately labelling the raw data collected from the sensor. This process can be quite difficult, specifically when dealing with mmWave raw data that is notoriously not intuitively easy to correctly label.

To solve this challenge, one direction is through information fusion, more specifically the fusion of mmWave radar and camera. As a result, it is important to understand the processes involved in general information fusion, with respect to mmWave radar and camera. Information fusion with mmWave radar and camera refers to the combination of the two independent streams of data, so that they are presented and interpreted from a unified perspective [73]. There are a number of different variables that are involved in achieving this fused state of information. In an attempt to break down the varying components involved in information fusion [74], the following high-level characteristics should be considered:

- System architecture;
- Fusion depth;
- Fusion process;
- Fusion algorithm.

The system architecture of mmWave radar and camera fusion focuses on the high-level structure that the fusion process operates on. In a review article presented by

Table 4.1: Types of mmWave radar and camera fusion system architectures.

Architecture Type	Description
Centralised	This refers to an architecture where the individual raw data of both the camera and mmWave radar is obtained independently and converged in a central processor for processing.
Distributed	This refers to an approach where each the radar and camera process their own data independently and sends the post-processed data to a central fusion unit to then before fusion on post-processed data.
Hybrid	The hybrid fusion approach refers to an architecture where some sensors follow the centralised approach, as defined above, and others follow the distributed approach, also as defined above. Measurements from all sensors are combined into a hybrid measurement which in turn is used to update the final data.

[74], the authors have identified three major fusion structures that are commonly abstracted in related literature. These three types of fusion system architectures are depicted in Table 4.1, and their respective benefits and limitations are shown in Table 4.2.

The three types of fusion architectures presented in Table 4.1 ultimately describe the major architecture types found in existing research. The rationale responsible for deciding which architecture type to implement over the others fundamentally stems from the run-time requirements a given solution must meet.

The next characteristics that can be used to describe mmWave radar and camera information fusion is the depth of the fusion that is performed. The authors of [75] and [76] term this characteristic as the level of fusion. This simply refers to the point in which the mmWave data is fused with the camera data, starting from the primitive point in which raw data is collected and stemming until a point where fusion might take place only once several layers of processing has already taken place

Table 4.2: Benefits and limitations of fusion architectures.

Architecture Type	Benefits & Limitations
Centralised	**Benefits:** Low information loss, original data preserved, simple structure, high processing rate. **Limitations:** Independent sensor units, large communication bandwidth required, high computing power needed by centralised unit, single point of failure.
Distributed	**Benefits:** Reducing transmission time, reduced pressure on the fusion centre, higher reliability resistance, low communication bandwidth. **Limitations:** Data collection units also require the capability of processing the data, central processor is operating on post-processed data resulting in reduced flexibility.
Hybrid	**Benefits:** Advantages of both centralised and distributed is retained, flexibility in satisfying varying requirements. **Limitations:** Complex data structures, increased computational and communication load, high design requirements.

Table 4.3: Types of mmWave radar and camera fusion depths.

Fusion Depth	Description
Low level	This class of fusion depth is best considered to be at the data level. It refers to a level of fusion that takes the raw data from each sensor to form a synthetic dataset illustrating a raw fused state, ready to be further processed.
Medium level	This refers to a class of fusion that takes place once several primitive features have been derived for each sensor independently and are fused to form a feature superset.
High level	This fusion level is considered an advanced form of fusion. Fusion at this level is taken place once independent outcomes have been derived for each sensor and the fused result is an expression of the combined sensor specific outcomes.

independently, for both/either radar and/or camera.

The authors of [75] and [76] have abstracted these depths of fusion into three progressive levels. These levels are further described in Table 4.3.

The fusion process is another aspect that can differentiate the fusion that takes place for mmWave radar and camera. The fusion process ultimately refers to the basis in which the actual fusion of the two sensors takes place upon. There are a number of different approaches that can serve as the means to perform fusion. One method explored and demonstrated by the authors of [77] attempts to spatially fuse the mmWave radar and camera. This process refers to the mmWave radar and camera each recording data in their own coordinate system. Following this, each of the sensor's coordinate system should be transformed into a world coordinate system, which closely depicts the three-dimensional coordinate system we perceive the world via. Another fusion process that is closely related to spatial fusion, and perhaps necessary for spatial fusion to take place, is fusion through sensor calibration. There are a number of varying techniques presented for calibrating mmWave and camera sensors, such as the work presented by the authors of [78]–[81]. Lastly, and probably the most simple process in which the basis of fusion can take place is temporally. Finally, regardless of the basis in which the fusion takes place, an appropriate correlation and association algorithm needs to be designed and implemented.

The research discussed in this chapter presents a framework for automated labelling of mmWave radar data using information fusion theory through camera. The research and methodologies we propose in this chapter are novel in three major regards. Firstly, the generalised automated labelling framework we present is one of the first proposed in the context of mmWave, where an attempt has been made to abstract the specific teacher and student objectives from the framework. Secondly, the framework we present is also one of the first of its kind to encompass the complete processing chain for training a standalone mmWave radar classification model using camera as a teacher. Lastly, the example implementation of the framework we present demonstrates a novel adaption for the correlation and fusion of camera and radar data. These primary contributions we present are further explained in the following:

- The radar training with camera labelling framework we present is generalised by definition as it is not one that is specific to a given classification problem in either the radar or camera domain. The agnostic nature of the framework we propose is the first of its kind that we are aware of, in the context of mmWave radar. Existing approaches are either specific to the task of object detection or specific to the classification problem the given authors are attempting to solve.

- The framework we present in Section 4.3 is the first of its kind that includes a suggested approach towards all stages in the processing chain involved in achieving a radar classifier. Existing approaches usually have a focus on presenting a framework that only shows a means for labelling camera data, usually specific to the task at hand, and applying it to either raw or pre-processed radar data. Our framework also satisfies that objective but takes the labelled data further and demonstrates how this labelled radar data can be used in a teacher and student based approach to form a standalone radar classifier.

- To demonstrate the feasibility of the framework proposed, we also demonstrate a practical implementation of our proposed framework. In our example implementation we demonstrate how a pre-trained camera classifier can be used to label raw mmWave data for HAR, in conjunction with performing mmWave multiple object tracking. The correlation technique we devised and utilised is unique and a looser form of calibration that takes place between the camera and radar. This removes the need for tight coupling between raw radar points and points in the vision domain.

4.2 Sensor Fusion Architectures

As increasingly more deep learning based sensing research is being released for mmWave radar, the difficulties associated with the labelling of mmWave data is being acknowledged. As a result, a few different labelling strategies have been presented in recent literature, ultimately demonstrating the feasibility of using another sensor, such as camera, to label datasets collected by radar.

One of the earlier pieces of research that demonstrate a fusion based approach with radar and camera to classify objects is the work presented by the authors of [82]. The authors of [82] deconstruct the problem into a two stage approach. The first stage involves recording the data and performing a typical Kalman filter based approach to identify objects in the field of view of the radar. This involves using the Kalman filter to predict the state of a moving object based on previous measurements, which helps in tracking the object's position and velocity over time. The mathematical foundation of the Kalman filter consists of prediction and update steps that can be expressed through the following set of equations:

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} + Bu_k, \tag{4.1}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q, \tag{4.2}$$

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1}, \tag{4.3}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}), \tag{4.4}$$

$$P_{k|k} = (I - K_kH)P_{k|k-1}, \tag{4.5}$$

where $\hat{x}_{k|k-1}$ is the predicted state, $P_{k|k-1}$ is the predicted covariance, K_k is the Kalman gain, z_k is the measurement, F is the state transition model, B is the control-input model, u_k is the control vector, Q is the process noise covariance, H is the observation model, and R is the measurement noise covariance.

In the second stage, the identified radar points are projected onto the camera's image plane through a coordinate transformation process. This involves using the intrinsic and extrinsic parameters of the camera to map the 3D radar points to 2D image coordinates. The mathematical relationship governing this transformation from 3D radar points to 2D image coordinates is expressed as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{4.6}$$

where (u, v) are the image coordinates, K is the camera intrinsic matrix, R is the rotation matrix, t is the translation vector, and (X, Y, Z) are the 3D world coordinates of the radar points.

Another more recent piece of literature that demonstrates an approach to fusion of mmWave radar and camera is the work presented in [83]. The approach discussed by the authors of [83] is largely similar to the technique presented in [82]. The authors of [83] provide a detailed methodology for the fusion process, which includes spatial and temporal fusion techniques. The spatial fusion involves converting the radar's polar coordinates into the camera's image plane coordinates using a series of transformations, similar to the technique presented by [82]. This is achieved by first converting the radar's polar coordinates into a rectangular coordinate system and then projecting these coordinates onto the image plane using the camera's intrinsic and extrinsic parameters.

In addition to spatial fusion, the authors also address the challenge of temporal fusion. Given that radar and camera sensors may operate at different frequencies, it is crucial to synchronise the data from both sensors. The authors propose a method to achieve this by creating radar, camera, and data fusion processing threads. When the data fusion processing thread is triggered, the system acquires radar data from the buffer queue that is consistent with the image data in time. This ensures that the data from both sensors is synchronised, allowing for accurate fusion and object detection.

The fusion model proposed by [83] presents results that indicate the proposed fusion method achieves a detection rate of up to 91.6%, demonstrating its effectiveness in real-world scenarios. The authors also highlight the advantages of their approach, such as reduced data processing time and improved detection accuracy, by focusing on the Region of Interest (ROI) generated from the radar data. It is also noted that [83] makes use of bounding box regression to refine the detected ROIs. This technique, initially proposed by [84], involves mapping the original window to a regression window that more closely aligns with the ground truth. The objective function for this regression process is designed to minimise the difference between the predicted value and the ground truth, ensuring accurate localisation of objects within the ROI.

Some more recent works presented by the authors of [85] produce a labelled FMCW dataset correlated with a Inertial Measurement Unit (IMU) sensor and corresponding camera frames. The labelling strategy proposed by the authors of [85] ultimately relies on time synchronisation between the three sensors. After temporally aligning the sensors, the authors require spatial calibration between the radar and camera in order to match detected objects.

The technique used by [85] to spatially calibrate the radar and camera involves introducing an object that is both distinctly identifiable in vision and reflectivity in the radar domain. This spatial calibration process described by [85] essentially involves the use of a radar corner reflector, which is a well-known object in radar signal processing due to its strong reflective properties. The reflector is placed at various locations within the field of view to establish point correspondences between the radar and camera frames. This calibration process is crucial for accurately projecting radar detections onto the camera image plane, enabling effective sensor fusion.

The work of [85] leads to an interesting question regarding the techniques available to calibrate the mmWave radar and camera sensors. A review presented by [86] breaks this question down into three overarching components that encompass the sensor calibration in the context of radar and vision presented in modern literature:

- Coordinate calibration - the alignment of individual points in the radar with objects in the field of view of the camera. This initial stage of calibration can be seen implemented in three varying mechanisms in the works presented by [87]–[89].

- Radar point filtering - where noise and undesirable data is acknowledged and filtered from the radar data. The work of [90] presents an approach that demonstrates calibration involving the filtering of undesired data points based on speed and angular velocity.

- Error calibration - refers to the processes implemented to overcome errors in the calibrated data. There are many methods that can be devised to attempt to overcome calibration error. One approach presented by [91] demonstrates a EKF that is used to model the measurement errors present in the independent sensors.

The authors of [92] and [93] propose two similar approaches that demonstrate object detection through the fusion of radar and camera. Both of the techniques demonstrate an Artificial Neural Network (ANN), where the inputs are pre-processed radar data and raw camera data. The primary difference between the two techniques is that the authors of [92] pre-process the radar data to produce range-azimuth images as an input for the ANN. While, the authors of [93] pre-process the radar data to form 2D point cloud data and utilise this as the input into the ANN.

In [92], the authors focus on early fusion of camera and radar sensors to enhance the accuracy and robustness of object detection in advanced driver assistance systems. They propose a deep learning architecture called FusionNet, which combines minimally processed radar signals with corresponding camera frames.

The radar data is fed into the network as a dense 2D range-azimuth image, allowing the use of feature pyramid network structures, popular in image object detection networks. The camera data is transformed into Cartesian space using a Inverse Perspective Mapping (IPM) to align with the radar data.

The FusionNet architecture consists of independent branches for each sensor, followed by fusion layers that concatenate the spatially aligned feature maps from both branches. The network is trained using a unique strategy of partially freezing the network and fine-tuning to ensure meaningful representations from different signal sources.

On the other hand, [93] presents a deep learning-based radar and camera sensor fusion architecture for object detection, called CameraRadarFusionNet (CRF-Net). The proposed approach enhances current 2D object detection networks by fusing camera data and projected sparse radar data in the network layers. The CRF-Net automatically learns the optimal level for sensor data fusion to maximise detection performance. The radar data is pre-processed to form 2D point clouds, which are then projected onto the camera image plane.

Lastly, an approach presented by the authors of [94] demonstrates an auto-labelling framework, achieving a similar goal to what we present in this chapter but through a different means. The approach presented by [94] uses an active learning system based on a CNN. Although the technique presented by [94] demonstrates promising results, it is important to note that the technique requires human input to manually label ambiguous data. The framework we present in this chapter demonstrates an approach that requires no human interaction for labelling of radar frames.

4.3 Radar Training with Camera Labelling and Supervision Methodology

In this section we aim to describe and illustrate a generalised methodology for labelling radar data and training a standalone radar model using camera as the ground truth for the radar model. The purpose of this methodology is to provide a framework for others to follow when attempting to extend camera based models into a radar based model. The methodology described in this section is practically applied and demonstrated in Section 4.4 of this chapter.

4.3.1 Problem Space

Raw radar data is notoriously difficult to interpret intuitively without applying preprocessing techniques to extract the desired information. The radar data typically consists of range, velocity, and angle information, which can be challenging to visualise and understand without appropriate processing. Furthermore, the labelling of raw radar data can be a difficult and tedious task for a domain expert, especially due to the large dataset sizes that are typically involved. This labelling process often requires manual inspection and annotation, which is time-consuming and prone to human error. As a result of this labelling difficulty, training a model that utilises radar data to classify complex events also becomes a difficult task. This problem is typically addressed in existing literature by reducing the dataset size of the radar data or restricting the potential of the classifier being trained to only a small set of classification types. Although this may alleviate the problem, there are negative implications to the potential of the designed radar model. A reduced dataset size can lead to overfitting, where the model performs well on the training data but fails to generalise to new, unseen data. Similarly, restricting the classification types can limit the applicability and robustness of the model. Therefore, there is an evident need to devise a solution to simplifying the labelling approach for radar data that in turn can be utilised to train a classifier without impacting the constraints of the designed model.

Camera classification networks are a well-defined and researched domain. As seen in Section 4.2 of this chapter, there are many existing models available that demonstrate successful classification capabilities for a variety of complex movements. These models leverage the rich spatial information available in camera images to perform tasks such as object detection, pose estimation, and activity recognition. The methodology proposed in this chapter uses camera as a means of addressing this labelling challenge with raw radar data and the inherent training difficulty of standalone models/classifier networks. By using camera data to label radar data, we can leverage the strengths of both sensor modalities. However, attempting to ultimately use vision data to label and act as ground truth for radar data presents two major challenges that need to be considered.

Firstly, vision data is inherently a snapshot of a horizontal and elevation domain at a given point in time; in other words, the perspective of the two-dimensional data is typically considered to be still/static in nature. Radar data, on the other hand, is typically a perspective of a range/distance and relative angle, or an inferred horizontal plane. Additionally, radar data in this domain is also typically collected on moving/dynamic objects. This domain alignment issue between camera and radar data ultimately poses a challenge, specifically the correlation of static objects present in vision data with moving objects present in the radar data. To address this challenge, it is essential to develop techniques that can accurately map the spatial coordinates of objects detected in camera images to the corresponding coordinates in radar data. This may involve the use of calibration techniques to align the coordinate systems of the two sensors and the development of algorithms to track the movement of objects over time.

The second major challenge identified is also a correlation problem in nature that presents itself when operating in an environment where multiple objects are simultaneously present and/or moving in the field of view. This scenario ultimately surfaces the challenge of correctly associating the multiple objects in the vision data with the same objects in the radar data. In environments with multiple objects, it is crucial to develop robust data association techniques that can accurately match objects detected by the camera with their corresponding radar detections. Techniques such as data fusion and sensor fusion can be employed to combine information from both sensors and improve the overall accuracy and reliability of the system.

4.3.2 Proposed Approach

This section depicts the proposed solution methodology to the previously discussed problem space. The methodology proposed in this chapter should be interpreted as a framework that can be applied to a given camera classification model, so that radar can achieve an ideally equal performing standalone classification network.

The proposed approach can be conceptually considered in the following three stages:

1. Data collection;
2. Correlation and labelling;
3. Radar training.

Figure 4.1 illustrates the generalised processing chain that is involved throughout the aforementioned three high-level stages. The data collection stage is an abstraction in the framework that is responsible for collecting data independently of the radar and camera. The data collected from each of the different sensors is then undertaken through the appropriate pre-processing and normalisation methods depending on the particular application this framework is being applied to. The desired output state for the radar data is a sequence of radar data frames across the time domain. At this stage, the camera data should be in a state that is consumable by the particular camera classifier network that is being applied to train the radar with.

After successful data collection and the appropriate transformations, the pre-processed camera data should then be applied to the camera classifier being implemented to train the radar. The expectation of the camera classifier is to perform the respective classifications against the camera data so that a sequence of camera frames with labelled classifications can be obtained. The domain in which these camera frames are obtained can be considered abstract for the definition of this methodology, as this is dependent on the particular application.

An important part of the proposed methodology is the correlation approach to synchronise the camera and radar data. The time associated with the sample taken for the camera data is used as a reference so that the radar data can be extrapolated in order to synchronise with time. Figure 4.2 demonstrates the time bias that is present between the radar and camera samples.

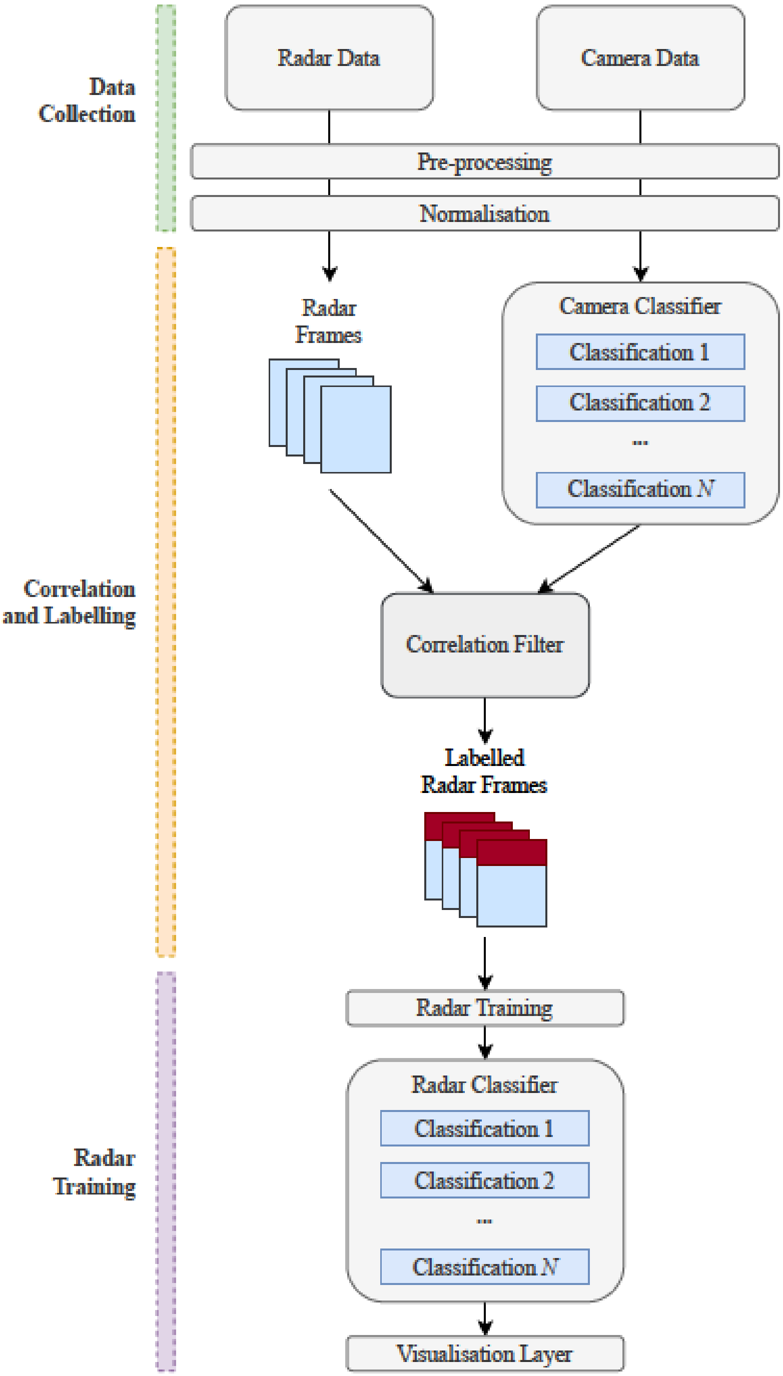


Figure 4.1: Radar training with camera labelling and supervision methodology processing chain.

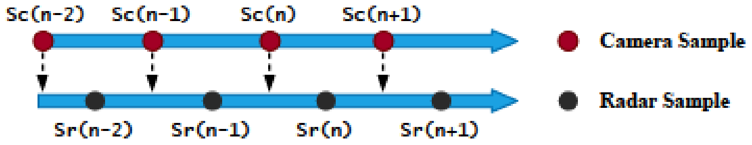


Figure 4.2: Radar and camera time alignment bias.

Assuming two consecutive radar time stamps are expressed as $S_r(n-2)$ and $S_r(n-1)$, the next sample predicted by the radar can be expressed as $S_f(n)$. Using the position and velocity components of the radar data, $S_r(n-1)$ and $S_f(n)$ can be linearly interpolated to estimate the radar sample at the correlating camera data point $S_c(n)$. The linear interpolation for temporal correlation is expressed as:

$$S_c(n) = S_r(n-1) + \frac{S_f(n) - S_r(n-1)}{t_f - t_{r(n-1)}}(t_c - t_{r(n-1)}), \tag{4.7}$$

where t_f is the time of the next radar sample, $t_{r(n-1)}$ is the time of the previous radar sample, and t_c is the time of the camera sample.

Once both radar and camera data have been correlated using the above approach, a labelled set of radar frames can be formed based on the correlation that was achieved. The labelled set of frames $F_l(n)$ can then be subjected to training for classification of the desired feature sets encoded in the radar and camera data.

The classification network applied to the labelled radar data can be an abstracted problem in the context of the framework proposed in this chapter. The proposed approach ultimately abstracts the design challenges associated with fusing and labelling the radar data with camera classifiers. As a result, a generic model can be applied and trained against the labelled radar frames based on the original camera classification network that was selected. The next section of this chapter demonstrates a practical implementation of the generalised methodology illustrated, showcasing its effectiveness in training a radar classifier using camera-labelled data.

4.4 System Design and Implementation

This section demonstrates an implementation of the labelling and supervision framework presented in the previous section. As mentioned in Section 4.3, the framework

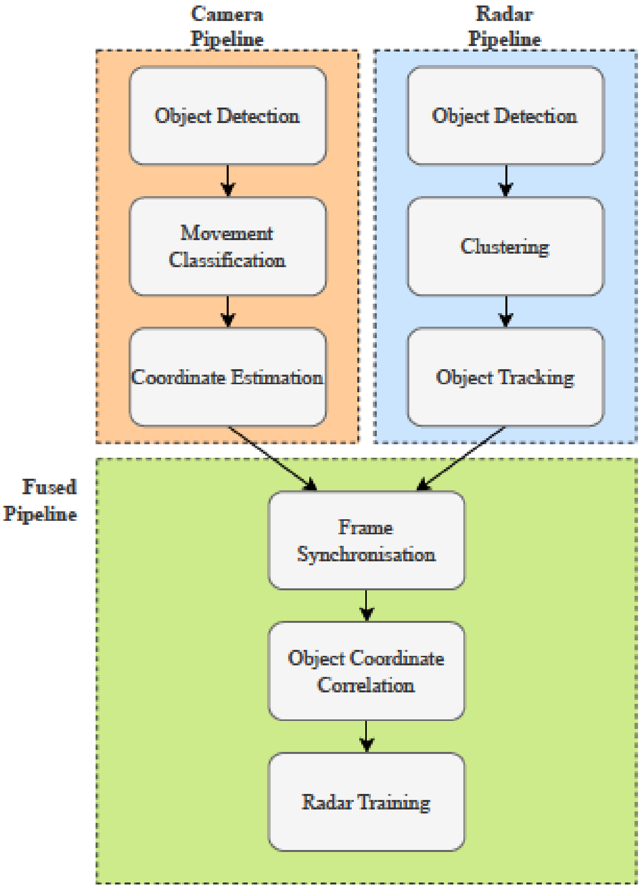


Figure 4.3: Radar tracking system design with a human movement pattern classifier trained with camera labelled frames.

provides a means to training a radar using labelled camera frames. As such, the design and implementation discussed in this section demonstrates the suitability of the framework by applying it to train a mmWave tracking system to classify human movement patterns.

Figure 4.3 illustrates the overall system design that implements the framework discussed in Section 4.3 and illustrated in Figure 4.1. The system design presented contains three high-level processing pipelines:

1. Radar Pipeline;

2. Camera Pipeline;

3. Fused Pipeline.

The remainder of this section will continue to break down the system design with respect to each of the pipelines illustrated in Figure 4.3.

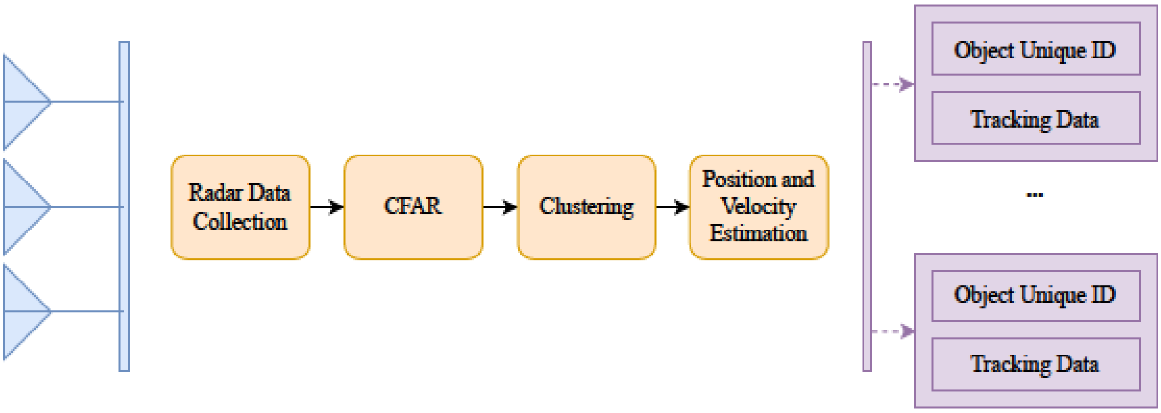


Figure 4.4: Radar processing pipeline design.

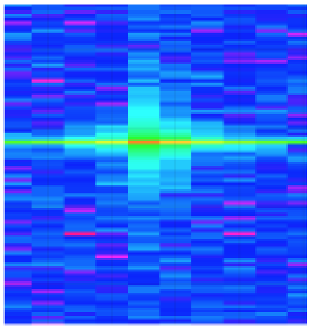


Figure 4.5: Generated radar range-Doppler heatmap.

4.4.1 Radar Pipeline

The radar pipeline concerns itself of the processing required to prepare the radar data for fusion with the camera frames. As seen in Figure 4.3, it is expected that the radar pipeline can achieve object detection and tracking. Figure 4.4 further extends on the high-level aspect of the radar pipeline presented in Figure 4.3.

The radar processing pipeline has been broken down into 4 different submodules. The *Radar Data Collection* module is responsible for collecting the raw ADC data from the radar. The raw radar data is then processed to perform two FFTs, the range-FFT followed by the Doppler-FFT. These transformations are necessary so that the respective range-Doppler heatmaps can be generated for each radar frame. An example range-Doppler heatmap can be seen in Figure 4.5.

The second module of the radar processing pipeline is the *Constant False Alarm Rate (CFAR)* stage, which is ultimately responsible for implementing a CFAR filter for performing object detection on the range-Doppler heatmaps. It is important to note the decision to operate with range-Doppler heatmaps was made primarily for the later radar classification that will be discussed. The CFAR algorithm dynamically adjusts the detection threshold based on the noise level in the surrounding cells of the radar data. Algorithm 4 details the CFAR process implemented for object detection in the range-Doppler heatmaps.

Algorithm 4 CFAR Process for Object Detection in Range-Doppler Heatmaps

```
Input: Range-Doppler heatmap $H$, guard cells $G$, training cells $T$, false alarm rate $P_{fa}$
Output: Detected objects $D$
Initialise an empty list $D$ to store detected objects
Calculate the number of cells $N$ in the training window: $N = 2T + 2G + 1$
Calculate the scaling factor $\alpha$ using the false alarm rate $P_{fa}$:
    $$\alpha = N \left( P_{fa}^{-1/N} - 1 \right) \qquad (4.8)$$
for each cell $(i, j)$ in the heatmap $H$ do
    Extract the training window around the cell $(i, j)$, excluding the guard cells
    Calculate the noise level $Z$ as the average power of the training cells:
        $$Z = \frac{1}{N} \sum_{(m,n) \in T} H(m, n) \qquad (4.9)$$
    Calculate the detection threshold $T_{cfar}$:
        $$T_{cfar} = \alpha Z \qquad (4.10)$$
    if $H(i, j) > T_{cfar}$ then
        Add the cell $(i, j)$ to the list of detected objects $D$
    end if
end for
return $D$
```

Following the object detection in the range-Doppler heatmaps, the data is further processed to be illustrated as a point cloud data so that traditional radar point cloud clustering and tracking can take place using DBSCAN and a Kalman filter. The radar hardware architecture used in this system was a TI IWR6843 mmWave radar

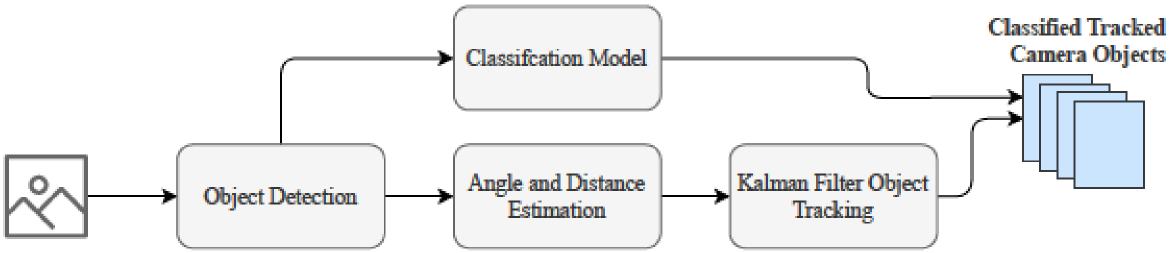

Classifcation Model
Classified Tracked Camera Objects
Object Detection
Angle and Distance Estimation
Kalman Filter Object Tracking

Figure 4.6: Camera processing pipeline design.

with a DCA1000EVM for capturing the raw ADC data of the radar. The complete radar pipeline is further detailed in Algorithm 5.

The radar pipeline consists of multiple processing stages, each contributing to the overall computational complexity. The Range-FFT and Doppler-FFT operations dominate the initial processing stages. For N ADC samples and M chirps, the FFT complexity is $\mathcal{O}(NM(\log N + \log M))$. The CFAR processing has complexity $\mathcal{O}(WH)$ for processing the range-Doppler heatmap of dimensions $W \times H$, where the window operations involve constant-time computations. For point cloud clustering, DBSCAN has a complexity of $\mathcal{O}(n \log n)$ with spatial indexing structures for n detected points. The Kalman filter tracking operations have complexity $\mathcal{O}(k \times d^3)$ per frame for k clusters with state dimension d, due to matrix operations in the prediction and update steps. The total radar pipeline complexity per frame simplifies to $\mathcal{O}(NM \log(NM) + WH + n \log n)$ since the FFT operations dominate when $N, M \gg W, H \gg n$, and the tracking complexity $\mathcal{O}(kd^3)$ is typically negligible for small cluster counts and low-dimensional state spaces.

4.4.2 Camera Pipeline

The camera pipeline is responsible for preparing and labelling the camera frames for fusion with the radar range-Doppler heatmaps. The data that is recorded from the camera must first be processed for object detection and each object coordinately mapped in the field of view. Following this the appropriate movement classifications can be made and associated with objects in the field of view of the camera. Figure 4.6 illustrates a more granular perspective of the stages involved in the camera processing pipeline.

Algorithm 5 Radar Pipeline for Object Detection and Tracking

```
Input: Raw radar ADC data $R_{adc}$, radar hardware parameters
Output: Tracked radar objects $T_{radar}$
Radar Data Collection:
Collect raw ADC data $R_{adc}$ from radar sensors
Range-FFT:
Perform Range-FFT on $R_{adc}$ to obtain range profiles $R_{range}$
Doppler-FFT:
Perform Doppler-FFT on $R_{range}$ to obtain range-Doppler heatmaps $H_{rd}$
CFAR Object Detection:
Define guard cells $G$, training cells $T$, and false alarm rate $P_{fa}$
Calculate the number of cells $N$ in the training window: $N = 2T + 2G + 1$
Calculate the scaling factor $\alpha$ using $P_{fa}$ as per Algorithm 4
for each cell $(i, j)$ in the heatmap $H_{rd}$ do
    Extract the training window around the cell $(i, j)$, excluding the guard cells
    Calculate the noise level $Z$ as the average power of the training cells
    Calculate the detection threshold $T_{cfar}$ using $Z$ and $\alpha$
    if $H_{rd}(i, j) > T_{cfar}$ then
        Add the cell $(i, j)$ to the list of detected objects $D$
    end if
end for
Point Cloud Generation:
Convert detected objects $D$ into point cloud data $P_{cloud}$
DBSCAN Clustering:
Apply DBSCAN clustering on $P_{cloud}$ to form clusters $C_{clusters}$
Kalman Filter Tracking:
for each cluster $c \in C_{clusters}$ do
    Predict the next state of the cluster using a Kalman filter
    Update the state of the cluster with new measurements
    Add the tracked cluster to the list $T_{radar}$
end for
return $T_{radar}$
```

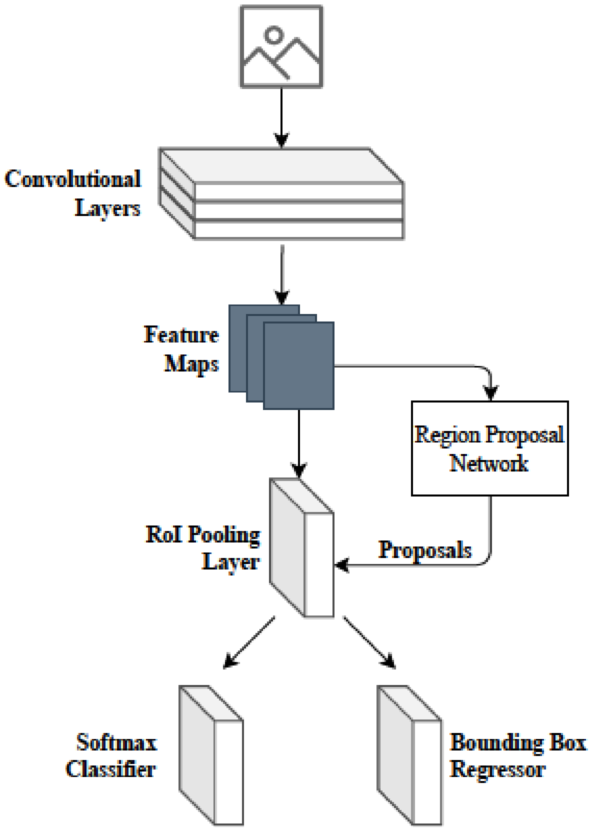


Figure 4.7: Faster R-CNN model design, used for camera object detection.

As illustrated in Figure 4.6, object detection is the first task that is performed in the camera processing pipeline. In order to realise camera object detection, a Faster Region-based Convolutional Neural Network (Faster R-CNN) is implemented. The structure implemented can be seen in Figure 4.7 and is based on the research presented in [95].

The generalised loss function adopted for camera object detection follows the multitask loss in Faster R-CNN [96]. The multitask loss function combines classification and regression objectives into a unified optimisation framework, which is defined as:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*), \tag{4.11}$$

where i refers to the index of an anchor (noting the definition of an anchor as per [95]), p_i is the predicted probability of anchor i being a detected object, p_i^* is the ground truth label for the given anchor i (derived as per Equation 4.12), t_i is the coordinate vector associated with the bounding box of the predicted anchor i, and

t_i^* is the ground truth of bounding box coordinate vector associated with anchor i that is an object.

The ground truth label p_i^* for a given anchor i is binary in value and serves as the target classification for training. This binary labelling strategy is represented as:

$$p_i^* = \begin{cases} 1, & \text{anchor } i \text{ is positive} \\ 0, & \text{otherwise} \end{cases}, \tag{4.12}$$

Furthermore, the terms L_{cls} and L_{reg} refer to the loss functions for the classifier and regressor respectively, illustrated in Figure 4.7. N_{cls} and N_{reg} are the normalisation of these two terms. The classification loss function implements cross-entropy loss to measure the prediction accuracy, expressed as:

$$L_{cls}(p_i, p_i^*) = -\frac{1}{N}\sum_{i}^{N} p_i \log(p_i) + (1 - p_i)\log(1 - p_i), \tag{4.13}$$

The regression loss function measures the accuracy of bounding box coordinate predictions using the smooth L1 loss function, which is defined as:

$$L_{reg}(t_i, t_i^*) = smooth_{L1}(t_i, t_i^*), \tag{4.14}$$

where the $smooth_{L1}$ function is defined as per [96].

After object detection has been performed, the classification model is then applied to the cropped detected objects. The purpose of the classification model in this implementation is to:

1. Formulate a 2D skeleton for each detected object in the field of view;
2. Classify the human activity that using the 2D skeleton.

In order to achieve this, each of the detected objects is run through AlphaPose [97] to generate the respective 2D skeleton for the detected object. AlphaPose is a state-of-the-art system for human pose estimation that provides accurate and real-time multi-person pose estimation. The architecture of AlphaPose consists of several key components: a backbone network, a detection module, a pose estimation module, a pose refinement module, and Pose Non-Maximum Suppression (NMS) [98].

The backbone network, typically a deep CNN like ResNet, extracts high-level features from the input image. The detection module, using networks such as Faster R-CNN or YOLO, detects human bounding boxes in the image [98]. These bounding boxes are used to crop ROIs from the feature maps generated by the backbone network.

The pose estimation module predicts the key points of human poses within the detected bounding boxes. It consists of deconvolutional layers that upsample the feature maps to a higher resolution, followed by convolutional layers that predict heatmaps for each key point. The Gaussian heatmap representation for each key point provides a probabilistic estimate of joint locations, expressed as:

$$H_k(x, y) = \exp\left(-\frac{(x - x_k)^2 + (y - y_k)^2}{2\sigma^2}\right), \tag{4.15}$$

where (x_k, y_k) is the ground truth position of key point k, and σ is the standard deviation of the Gaussian peak.

The pose refinement module improves the accuracy of the predicted key points using convolutional layers. The refinement process minimises the distance between the predicted key points and the ground truth key points. Pose NMS handles redundant detections and overlapping poses by comparing the similarity of predicted poses and retaining only the most confident ones. The similarity between two poses is measured using the Euclidean distance between corresponding key points [98].

During training, AlphaPose uses a combination of heatmap loss and refinement loss to optimise the network [98]. The total loss function is defined as:

$$L_{total} = L_{heatmap} + \lambda L_{refine}, \tag{4.16}$$

where $L_{heatmap}$ is the loss for the initial heatmap predictions, L_{refine} is the loss for the pose refinement, and λ is a weighting factor.

During inference, the input image is passed through the backbone network to extract feature maps. The detection module detects human bounding boxes, which are used to crop the ROIs from the feature maps. The cropped feature maps are passed through the pose estimation module to predict the initial key points, which are then refined using the pose refinement module [98]. Finally, the Pose NMS module eliminates redundant poses to produce the final pose predictions.

The result of the AlphaPose system is then passed as an image to a CNN that has been pre-trained to classify poses that are associated with:

- Walking;
- Running;
- Falling.

The pre-trained model is ensured to have an accuracy greater than 92%. The accuracy of this classifier network is important, as it will ultimately be built into the mmWave classification network during the fusion pipeline. In parallel with the classification of the detected objects, their location in the field of view is also jointly estimated using camera calibration. This ultimately results in each detected object j having a respective given coordinate (X_{jk}, Y_{jk}) for each camera frame k, where X is used to denote the horizontal coordinate and Y is used to represent the estimated range of the object (as opposed to height). Finally, a Kalman filter is applied to predict the detected object's true location more accurately whilst being tracked in the field of view. The entire camera pipeline process is detailed in Algorithm 6.

The camera pipeline computational complexity is dominated by deep learning operations that scale with image dimensions and detected objects. The pipeline consists of Faster R-CNN object detection with complexity $\mathcal{O}(W_I H_I + A + n_{obj})$ for image processing, anchor evaluation, and ROI operations, followed by AlphaPose pose estimation at $\mathcal{O}(n_{obj} \times W_{bb} H_{bb})$ for bounding box processing, activity classification at $\mathcal{O}(n_{obj})$, and Kalman filter tracking at $\mathcal{O}(n_{obj})$, as previously discussed for low-dimensional state spaces. The total camera pipeline complexity simplifies to $\mathcal{O}(W_I H_I + n_{obj} \times W_{bb} H_{bb})$ since the backbone network and pose estimation operations dominate when processing high-resolution images with multiple detected

Algorithm 6 Camera Pipeline for Object Detection and Classification

```
Input: Raw camera frames $F$, pre-trained Faster R-CNN model $M_{rcnn}$, pre-trained AlphaPose model $M_{pose}$, pre-trained activity classifier $M_{act}$
Output: Labelled objects with activity classifications $L$
Initialise an empty list $L$ to store labelled objects
for each frame $f \in F$ do
    Object Detection:
    Run Faster R-CNN model $M_{rcnn}$ on frame $f$ to detect objects
    Extract bounding boxes $B = \{b_1, b_2, \ldots, b_n\}$ and object scores $S = \{s_1, s_2, \ldots, s_n\}$ from $M_{rcnn}$
    for each bounding box $b_i \in B$ do
        Crop the object region $O_i$ from frame $f$ using bounding box $b_i$
        Pose Estimation:
        Run AlphaPose model $M_{pose}$ on cropped object $O_i$ to estimate 2D skeleton
        Extract 2D skeleton key points $K_i$ from $M_{pose}$
        Activity Classification:
        Convert 2D skeleton key points $K_i$ to an image representation $I_i$
        Run activity classifier $M_{act}$ on image $I_i$ to classify activity
        Extract activity label $A_i$ from $M_{act}$
        Object Tracking:
        Estimate object location $(X_i, Y_i)$ using camera calibration
        Apply Kalman filter to predict true location $(\hat{X}_i, \hat{Y}_i)$
        Store Labelled Object:
        Create labelled object $l_i = (b_i, s_i, K_i, A_i, (\hat{X}_i, \hat{Y}_i))$
        Add labelled object $l_i$ to list $L$
    end for
end for
return $L$
```

objects.

4.4.3 Fused Pipeline

The fusion pipeline is then finally responsible for associating the tracked objects in the radar domain with the tracked and classified objects in the camera domain. As mentioned in Section 4.3 of this paper, before fusing the two domains a time bias between the domain samples needs to be accommodated for. In our implementation, this is achieved by granulating the radar samples so that positional estimates are calculated between radar samples. This positional estimates are deduced so that they correspond with the sampling rate of the camera system.

The association and correlation of the detected objects is then made so that the tracked objects in the camera domain can be related to the tracked objects in the radar domain. This correlation is made using the deltas of the velocity and acceleration between the respective predicted locations of the camera and radar tracking algorithms. For both camera and radar, the displacement vector is used for correlation using Pearson's Correlation Coefficient. This approach consequently removes any detected objects that are not commonly identified across domains, accommodating for the scenario where one sensor picks up an object that the other does not. The mathematical formulation of the displacement vectors for camera and radar systems is given by:

$$\vec{CP}_l = [cp_n - cp_{n-1}, cp_{n-1} - cp_{n-2}, \cdots, cp_2 - cp_1], \tag{4.17}$$

$$\vec{RP}_m = [rp_n - rp_{n-1}, rp_{n-1} - rp_{n-2}, \cdots, rp_2 - rp_1], \tag{4.18}$$

where $\vec{CP}_l$ and $\vec{RP}_m$ are the displacement vectors for camera and radar respectively, each for a given camera detected object l and radar detected object m. For the given detected object, the delta between all camera positional estimates cp in a sliding sample window n is calculated. The same is applied to the given detected radar object and its radar positional estimates rp in the sliding sample window n.

Using the displacement vectors for camera and radar defined above, the Pearson Correlation Coefficient is calculated for each pair of detected objects in both the camera and radar domain. This correlation measure is mathematically expressed as:

$$r_{lm} = \frac{n\sum C\vec{P}_l R\vec{P}_m - (\sum C\vec{P}_l)(\sum R\vec{P}_m)}{\sqrt{[n\sum C\vec{P}_l^2 - (\sum C\vec{P}_l)^2][n\sum R\vec{P}_m^2 - (\sum R\vec{P}_m)^2]}}, \tag{4.19}$$

where r is computed for all combinations of l and m. The absolute Pearson Correlation Coefficient $|r_{lm}|$ is taken and the maximal l and m combination is deemed to be the correctly correlated pair.

After correlation of the radar and camera domains, we ultimately have a labelled dataset we can use to train a model against for classification in the radar domain. The structure of the model used for classification of the radar data is a CNN with the input shape pertaining to clustered point cloud data for a single detected object. Algorithm 7 details the fused pipeline process for correlating and labelling the radar frames.

The computational complexity of the fused pipeline depends primarily on the correlation operations between detected objects across modalities. For n_c camera objects and n_r radar objects, computing all pairwise Pearson correlation coefficients has complexity $\mathcal{O}(n_c \times n_r \times w)$, where w is the window size for displacement vector calculation. The displacement vector computation has complexity $\mathcal{O}(w)$ per object pair, and the correlation coefficient calculation requires $\mathcal{O}(w)$ operations per pair. For typical multi-object scenarios with $n_c, n_r \leq 10$ and window sizes $w \leq 20$, this represents a computationally efficient fusion process. The time synchronization through linear interpolation has complexity $\mathcal{O}(n_r \times w)$ for radar data granulation.

4.5 Results

The system described in Section 4.4 was experimentally tested in varying environmental conditions to prove its performance. The first task is to collect the necessary dataset that can be used to train the radar. The dataset compiled needs to jointly have both camera and radar samples, so that the respective data fusion can take place. This cannot be collected independently.

Algorithm 7 Fused Pipeline for Radar and Camera Data Correlation and Labelling

```
Input: Radar data $R$, Camera data $C$, Radar sampling rate $f_r$, Camera sampling rate $f_c$
Output: Labelled radar frames $F_l$
Initialise an empty list $F_l$ to store labelled radar frames
Calculate the time bias $\Delta t$ between radar and camera samples:
    $$\Delta t = \frac{1}{f_r} - \frac{1}{f_c} \qquad (4.20)$$
for each radar frame $r \in R$ do
    Granulate Radar Samples:
    Estimate radar positions $rp$ at camera sampling rate $f_c$ using linear interpolation:
    $$rp_n = rp_{n-1} + \left(\frac{rp_{n-1} - rp_{n-2}}{\Delta t}\right) \Delta t \qquad (4.21)$$
end for
for each camera frame $c \in C$ do
    Object Detection and Classification:
    Detect objects and classify activities in camera frame $c$ using Algorithm 6
    Extract detected objects $O_c = \{o_{c1}, o_{c2}, \dots, o_{cn}\}$ with positions $(X_c, Y_c)$
end for
for each detected object $o_c \in O_c$ do
    Correlation with Radar Data:
    Calculate displacement vectors for camera and radar        ▷ Eq. 4.17 and 4.18
    Calculate Pearson Correlation Coefficient                   ▷ Eq. 4.19
    Find the maximal $|r_{lm}|$ and correlate the corresponding camera and radar objects
end for
Label Radar Frames:
for each correlated pair of objects $(o_c, o_r)$ do
    Assign the activity label from camera object $o_c$ to radar object $o_r$
    Add labelled radar frame $F_l(o_r)$ to list $F_l$
end for
return $F_l$
```

A dataset containing 1000 images was collected across 4 different sessions, where each session had a different external environment. Two of the sessions were recorded indoors and the other two were in an outdoor setting. In all recorded sessions, we ensured we recorded situations that included:

- No targets in the field of view;
- A single target in the field of view;
- Multiple targets in the field of view.

Table 4.4: Distribution of activities in the recorded dataset.

Activity	Distribution
Running	26.69%
Walking	25.02%
Falling	23.34%
Unknown	24.95%

Additionally, the four types of activities were distributed along the 1000 images as per Table 4.4. The frequency in which these activities took place is not a factor of the 1000 images taken. This is due to the fact that one or more activities could be present several times in a single image. This is a result of the potential for multiple objects to be detected and independently processed in a single frame.

The total dataset, and inner classifications, were equally shuffled to prevent a bias of randomisation between classification types. The shuffled dataset was then divided into training, validation and testing subsets. The first 60% of the equally randomised recorded dataset was reserved exclusively for training of the camera classifier and subsequently the radar classifier. The next 20% was then used for validation of the trained models, allowing us to further refine the classifiers using the validation dataset. Lastly, once the best performance was obtained, the classifiers were tested against the final reserved 20% of the dataset.

The accuracy results of our final trained radar system are presented in Figure 4.8. The camera trained radar classifier is compared with the accuracy of the trained standalone camera system and the manually labelled radar classifier, in varying

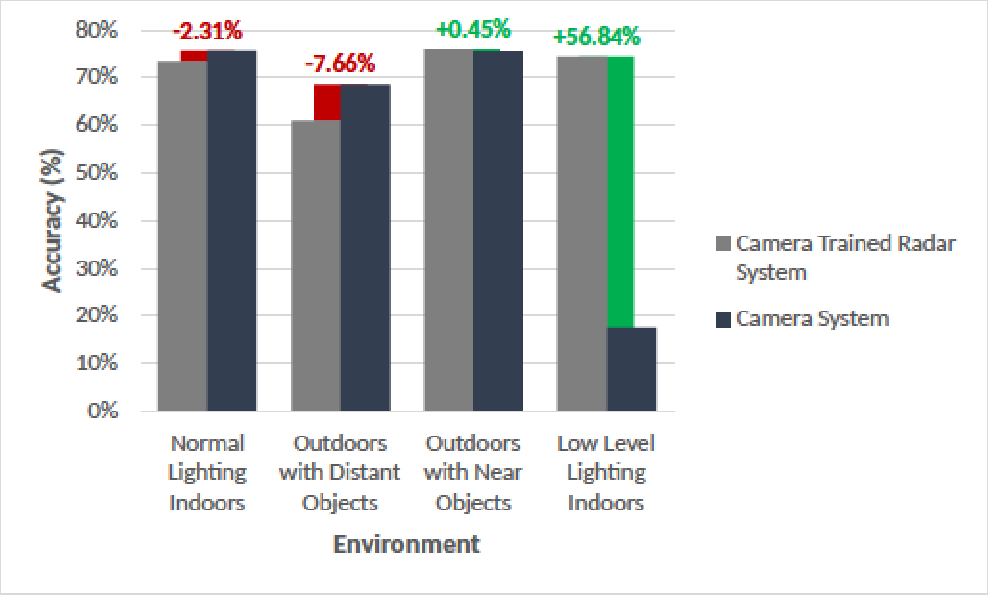


Figure 4.8: Camera trained radar system accuracy in contrast to a trained standalone camera system.

environmental setups. To clarify, each of the aforementioned systems is further described as:

- **Camera Trained Radar Classifier:** A radar classifier trained using camera labelled data via the framework proposed in this chapter.
- **Trained Standalone Camera System:** A camera classifier that is used to label the frames for the camera trained standalone radar classifier.
- **Manually Labelled Radar Classifier:** A radar classifier, of the same design as the camera trained standalone radar classifier, trained using manually labelled radar data.

In Figure 4.8, it can be seen that the radar classifier, that was trained using camera labelled data, produced an outcome similar, and in some circumstances more superior, to that of the standalone camera classifier. In most "normal" scenarios the radar classifier performed largely identical to the camera classifier. However, there are two environmental changes that should be noted as outliers, the first being objects that are distant.

In the scenario where the camera trained radar classifier was attempted with targets at a distance greater than 6 meters, the accuracy of the model was 7.66% less,

Table 4.5: Accuracy similarity between the standalone camera system and camera trained radar system.

Environment	**Trained Similarity**
Normal lighting indoors	97.69% ↓
Outdoors with distant objects	92.34% ↓
Outdoors with near objects	99.55% ↑
Low level lighting indoors	43.16% ↑

compared to the camera classifier. On further analysis of the results, it appears this is likely due to the fact that the point cloud data per cluster (i.e. detected object) is much leaner compared with objects that are within 6 meters of the radar. The leaner point cloud data results in a lack of distinguishing features between activities in the radar domain. This challenge could potentially be overcome through some additional design considerations with the chirp of the radar.

The second outlier, that is worth noting, is the experiment performed in an indoor room with low levels of light. As expected, the camera trained radar classifier is not impacted by the lighting conditions, and as a result demonstrates an accuracy that is 56.84% higher than the standalone camera classifier in the same lighting conditions.

Given the radar is being trained using camera labelled data, the best network we could theoretically achieve with the radar is one that is of equal performance to the teacher network (the standalone camera system). The exception to this is any sensor specific characteristics that might inhibit the performance of a given sensor, such as ambient lighting in the context of camera. This particular regard was evident in the second outlier identified, where the camera trained radar network was more performant than the standalone camera network, simply due to ambient lighting. Whilst acknowledging the aforementioned outliers, it is evident that the camera trained radar system performed with a high degree of similarity to the standalone camera system. The trained similarity between the two systems is summarised in Table 4.5, where a ↓ implies an inferior similarity and a ↑ implies a superior similarity, with respect to the trained radar system. The high degree of similarity between the

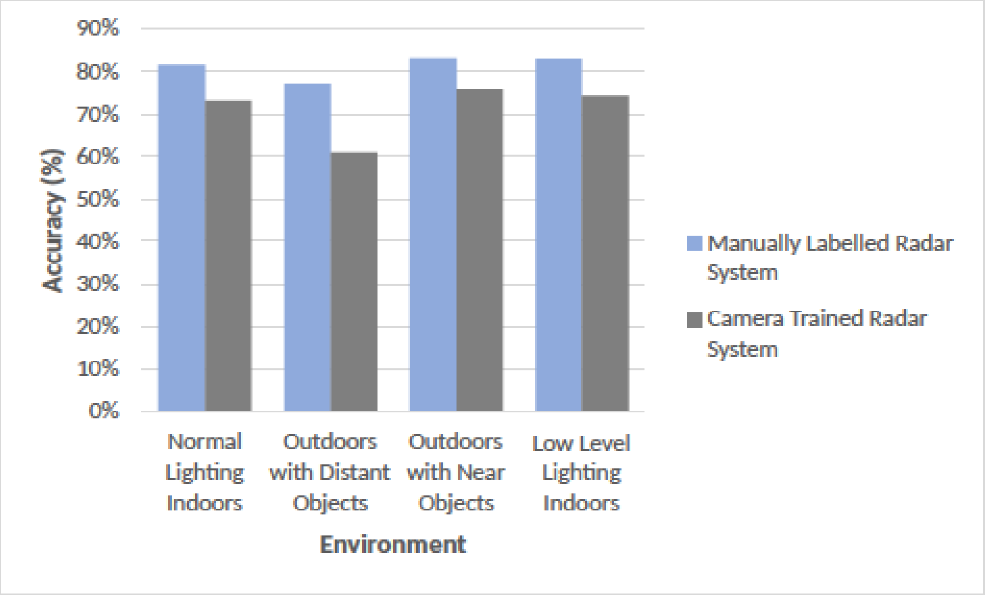


Figure 4.9: Manually labelled trained radar system accuracy in contrast to the camera trained radar system.

teacher network (the standalone camera system) and the student network (the camera trained radar system) demonstrates the suitability of the proposed generalised framework toward training a radar model with camera labelled data.

In order to better understand the theoretical potential of the radar classifier, the camera trained radar classifier was compared against a manually labelled radar classifier. The purpose of this comparison scheme was to demonstrate the potential of the implemented radar classifier. The significance of this experiment is to firstly highlight the capability that could potentially be expected with the design of the radar classifier, and secondly to gain an understanding of the pre-encoded errors the camera trained radar classifier incurs as a result of labelling errors in the camera domain. Figure 4.9 illustrates the accuracy of the manually labelled trained radar system in contrast to the camera trained radar system. Figure 4.9 highlights the theoretical potential the radar classifier can achieve when trained with a manually labelled dataset.

Assuming the manually labelled dataset is not incorrectly labelled, it is expected that training the radar system with a manually labelled dataset will yield higher results than a camera trained radar system. This is ultimately due to the fact that a camera trained radar classifier will incur labelling errors associated with the camera classifier. This hypothesis is ultimately supported by the results presented in Figure 4.9. It can be seen that the camera trained radar system does not meet the same performance as the manually labelled radar system. Despite the theoretical potential of the radar classifier, the performance of the camera trained radar system, implemented using the proposed framework, is on average 96.52% as performant as the camera classifier, as seen in Table 4.5 when negating the outperforming low level lighting environment.

4.6 Conclusion

The research presented in this chapter demonstrates a framework for developing a classifier for mmWave radars, using camera as a teacher for the mmWave radar student network. The example implementation, presented in Section 4.4, shows how the framework can be implemented to achieve a radar classifier that is as accurate as the teacher camera classifier. This performance is demonstrated without compromising on the beneficial characteristics of the radar, such as the non-sensitivity to illumination.

The proposed camera trained method can achieve a level of performance that approaches the manually labelled radar system, particularly in cases where the camera can generate accurate recognition performance. Hence, using the proposed framework can provide a significant saving on manual labelling for radar data. The performance of the camera-trained method is degraded where camera's recognition is limited. This is specifically seen in the results presented for the "Outdoors with Distant Objects" environment.

In order to further the research presented in this chapter, additional camera based labelling networks should be analysed, through the methodology presented in Section 4.3, for their ability to train an equally performant radar network. Furthermore, it would be of interest in future research to conduct radar classifier design optimisations and compare the network performance across a variety of different radar hardware. Performing such an experiment will allow us to better understand the impact of intrinsic radar characteristics, such as the ADC sampling rate and maximum resolution, on the generalised performance of the proposed framework.

The framework presented in Section 4.3 should be considered as a foundation to designing mmWave classifiers. Adopting the framework presented in this chapter can help researchers alleviate the burden associated with the labelling of mmWave data. This labelling challenge usually results in researchers under-collecting an adequate set of training data to design an mmWave classifier. In this scenario, due to the limited training dataset collected, the classification network attempting to be designed may not reach its full potential, simply as a result of being deprived of training data. Hence, the framework we present may assist future research by providing a model that researchers can follow to remove the need for manual labelling of data when designing a classifier for mmWave radar.

Chapter 5

Joint mmWave Sensing and Tracking with mmCLAE

In this chapter, we present a comprehensive approach to enhancing the performance of mmWave multi-object tracking systems in adverse weather conditions, particularly focusing on rain-induced noise reduction and rain intensity classification. Through the use of deep learning techniques, we propose a Millimetre Wave Convolutional Long Short-Term Memory Autoencoder (mmCLAE) for effective noise reduction and a CNN for accurate rain intensity classification. The proposed methods are evaluated through extensive experiments, demonstrating significant improvements in system robustness and accuracy.

This chapter outlines the motivation, methodology, and experimental results of our contributions, providing a practical implementation that utilises our research from prior chapters to produce a unified mmWave tracking and sensing system. The structure of this chapter is as follows: Section 5.2 reviews classical methodologies for noise reduction and rainfall sensing in mmWave radar systems. Section 5.3 details the proposed unified system architecture, including the integration and workflow of the noise reduction and rain intensity classification modules. Section 5.4 describes the proposed noise reduction approach, including the architecture mmCLAE and its training process. Section 5.5 presents the CNN-based method for rain intensity classification. Section 5.6 presents experimental results demonstrating the effectiveness of the proposed approaches in enhancing multi-object tracking performance and robustness to rain artefacts. Finally, Section 5.7 concludes the chapter with a discussion on the implications and future research directions.

5.1 Introduction

The utilisation of mmWave radar in multi-object tracking applications has shown great potential, as already seen from a variety of perspectives throughout this thesis. However, the external environment and the conditions of this environment can have a significant impact on the performance of these systems. Rain, in particular, can introduce noise and artefacts that can severely degrade the accuracy and reliability of object detection and tracking. Rain-induced noise can manifest in various forms, including multipath effects, signal attenuation, and speckle noise, which can complicate the tracking process and reduce the overall system performance [99]–[101].

Typical methods for noise reduction in mmWave radar systems often rely on filtering techniques, such as adaptive filters or wavelet-based denoising algorithms [102]–[104]. Although these methods can be quite effective in reducing noise, they may not be well-suited to capturing the patterns and relationships present in more complex noise profiles. With the recent advancements in deep learning literature, it proves to be a promising avenue in addressing this challenge, with various studies exploring the application of CNNs and RNNs for noise reduction in mmWave radar signals [105].

In this chapter, we introduce one primary technique to mitigate this challenge during tracking, along with a secondary joint sensing proposal. To address the challenge of rain-induced noise, we present a novel noise reduction technique, mmCLAE, specifically designed to remove rain-induced artefacts from mmWave signals. The mmCLAE is trained and evaluated on our own dataset of both simulated and real mmWave signals with varying levels of noise and rain intensity. Additionally, we propose a CNN-based method for classifying rain intensities using features extracted from mmWave radar data. This approach leverages the strengths of CNNs in processing the spatially correlated data and detecting relevant patterns.

5.2 Classical Methodologies

In this section, in order to better understand the domain and solution we arrived at for mmCLAE, we review classical methodologies on noise reduction and rainfall sensing in mmWave radar systems. The section is organised into two main subsections: methodologies of noise reduction in mmWave radar and rainfall sensing in mmWave radar. We will first discuss traditional signal processing techniques specifically for noise reduction in more details and their associated challenges, followed by an exploration of deep learning approaches that have shown promising results in addressing these challenges. Subsequently, we will then discuss various methods for rainfall sensing, ranging from empirical models to advanced machine learning and deep learning techniques.

5.2.1 Noise Reduction

Noise reduction in mmWave radar systems is an essential topic to address for ensuring accurate and reliable object detection and tracking. This problem is especially exacerbated in scenarios where the external environment conditions are adverse, such as during rain and storms. Traditional signal processing techniques, such as adaptive filters and wavelet-based denoising, have been widely used to address this challenge in a more traditional sense [106], [107]. These methods, however, often struggle to capture the complex patterns and relationships inherent in noise profiles, particularly in the presence of rain-induced artefacts.

Deep learning has opened up new possibilities for more complex approaches towards noise reduction in mmWave radar systems. CNNs and RNNs have shown great potential in this domain, ultimately through their ability to learn hierarchical features and temporal dependencies from raw radar data [108], [109].

Traditional Signal Processing Techniques

Traditional signal processing techniques for noise reduction in mmWave radar systems stem from the use of adaptive filters, wavelet-based denoising, and typical statistical methods. Adaptive filters, such as the well known Least Mean Squares (LMS) and Recursive Least Squares (RLS) algorithms, have been commonly utilised to reduce noise by ultimately adjusting filter coefficients dynamically based on the input signal characteristics. For example, the LMS algorithm, known for its low computational complexity and stability, has been effectively applied in various signal processing applications to reduce noise and improve the Signal-to-Noise Ratio (SNR) [106].

Wavelet denoising techniques decompose the signal into separate frequency components to then selectively remove noise by thresholding the wavelet coefficients [110]. These techniques have been effectively applied in various radar systems, including automotive FMCW radar and atmospheric radar, to fundamentally mitigate noise and improve the SNR. In automotive radar systems for example, wavelet denoising has been used to suppress mutual interference by extracting and subtracting interference signals from the original radar signal [107]. Similarly, in atmospheric radar systems, multi-band wavelet transforms have been employed to denoise the Doppler spectrum, ultimately enhancing the detection of wind velocity parameters [111].

Lastly, statistical methods, such as the Kalman filter and its many variants, such as EKF and Adaptive Kalman Filter (AKF), ultimately make use of probabilistic models to estimate the true signal from noisy observations, as discussed in Chapter 2. The EKF is particularly useful for a system that is considered nonlinear, as it linearises the system around the current estimate to provide more accurate predictions [112]. AKF on the other hand, further enhances this by altering the process noise level in accordance to the dynamics of the system, ultimately making it more robust to sudden and sporadic changes [113].

Deep Learning Approaches

There have been a number of studies that explore the applications of CNNs and RNNs for the mitigation of noise in mmWave radar. CNNs have been particularly effective in encoding spatial correlations within the radar data, that can ultimately be used for the removal of noise artefacts that traditional methods may find challenging to isolate. Using the hierarchical feature extraction capabilities of CNNs, it is possible to design a model that can effectively identify and remove various types of noise that may be present in mmWave radar signals, as demonstrated in the research presented by [108], [114].

In addition to CNNs, RNNs, especially LSTM models, are being used to address the temporal dependencies naturally inherent in radar signals. LSTMs are especially well-suited for this due to their ability to maintain and update memory cells, making them ideal for modelling the temporal information. Through the incorporation of LSTMs into noise reduction frameworks, it is possible to achieve a noise reduced filtered signal, while ultimately preserving the temporal characteristics of the natural radar data [17], [109].

5.2.2 Rainfall Sensing

Rainfall sensing using mmWave radar is an interesting potential for providing accurate and high-resolution precipitation measurements. Several methodologies have been proposed for estimating rainfall intensity from radar data, including empirical models, machine learning techniques, and deep learning approaches.

Empirical models, for rainfall sensing with radar, are based on the inherent relationship between radar reflectivity and the rate of rainfall. The most commonly used empirical model is the Z-R relationship, which ultimately relates the radar reflectivity factor Z to the rainfall rate R through a power-law equation [115]. The parameters used as part of the Z-R relationship are usually determined from experimental data and can widely vary depending on the radar operating frequency and external environmental conditions. Although empirical models are relatively simple to implement, the accuracy of these models can be limited by the assumptions and estimations that are fundamentally encoded in the empirical design of the model [116].

Machine learning techniques have been employed to improve the accuracy of rainfall sensing in mmWave radar by leveraging the rich feature set available in radar data. SVM, random forest, and k-NN are some machine learning algorithms that have been used to classify rainfall intensity based on radar reflectivity and other derived features [117]. These techniques can capture complex relationships between the radar measurements and rainfall intensity, leading to improved estimation performance compared to empirical models.

A deep learning based approach has shown great potential in rainfall sensing, specif-

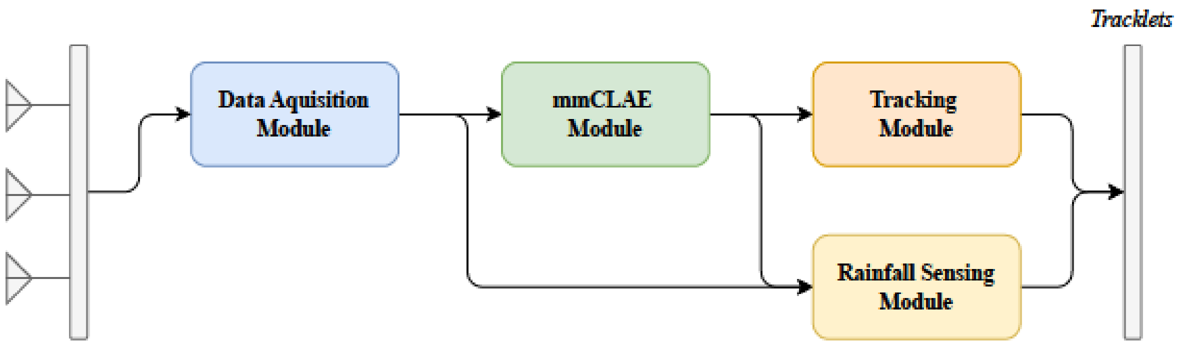


Figure 5.1: Unified system overview architecture for joint mmWave tracking and sensing with mmCLAE.

ically using mmWave radar, due to their ability to learn inherent relevant features from raw radar data, whether it be known or hidden in nature. CNNs have been widely used to determine spatial features from radar images and then classify rainfall intensity [118]. RNNs, particularly LSTM networks, have been utilised to model temporal characteristics of the rainfall artefacts in radar time series data [119]. Lastly, more recently, the use of hybrid architectures that ultimately combine CNNs and RNNs have been proposed to utilise both the spatial and temporal information that rainfall leaves in radar data [119].

5.3 Unified System Architecture

In this section, we discuss an overview of the unified system architecture that integrates the proposed joint tracking with mmCLAE and rainfall classification system. The unified system architecture at a high-level consists of four main components: the data acquisition module, the noise reduction module (mmCLAE), the rain sensing classification module, and lastly the tracking module. These components fundamentally integrate with one-another to serve as the wholistic joint tracking and sensing system, as seen in Figure 5.1.

5.3.1 Integration and Workflow

The integration of the noise reduction and rain intensity classification modules is achieved through processing the radar data in a parallel fashion, highlighted in Figure 5.1. The workflow is achieved following the below stream of events:

1. The mmWave radar data acquisition module captures raw radar signals from the environment.

2. The pre-processed radar data is fed into mmCLAE to remove rain-induced noise artefacts.

3. Tracking and sensing are then performed jointly in parallel.

   (a) The noise-reduced radar data is fed into the tracking module to perform multi-object tracking.

   (b) Simultaneously, the noise-reduced radar data, along with raw radar data features, is fed into the rain intensity classification module.

To ensure seamless integration and efficient training pipelines, the system has been designed as a modular architecture where each component operates independently but communicates through defined interface. This modularity allows for easy updates and improvements to individual components without affecting the overall system. The system also incorporates a feedback loop for future consideration where the output of the rain intensity classification module could potentially be used to dynamically optimise the parameters of the tracking module.

5.3.2 Advantages of the Unified System

The proposed unified system architecture offers several technical advantages:

- **Enhanced Noise Reduction:** mmCLAE is specifically designed to address the complex noise profiles induced by rain in mmWave radar signals. By leveraging the temporal dependencies captured by LSTM layers and the spatial features extracted by convolutional layers, mmCLAE effectively mitigates rain-induced noise artefacts. This results in significantly cleaner radar signals, which directly improves the accuracy and reliability of the multi-object tracking system.

- **Precision Rain Intensity Classification:** The CNN-based rain intensity classification module has been trained on a comprehensive dataset of mmWave data with varying rain intensities, learning to accurately classify different levels of rain intensity.

- **System Robustness and Adaptability:** The unified system architecture is designed to be modular and scalable, allowing for the future integration of additional sensing and processing modules. The parallel processing of noise reduction and rain intensity classification ensures that the system can adapt to a wide range of environmental conditions without compromising performance.
- **Comprehensive Data Utilisation:** Through the integration of both raw and noise-reduced radar data, the system leverages a richer set of features for both tracking and classification tasks. This ultimately enhances the system's ability to detect and track multiple objects accurately, even in challenging weather conditions.

The unified system architecture combines the advanced noise reduction capabilities of mmCLAE with the precise rain intensity classification of the CNN module, providing a robust and adaptable solution for joint tracking and sensing in mmWave radar systems. This integrated approach significantly enhances the performance and reliability of multi-object tracking systems in adverse weather conditions, making it suitable for a variety of applications, including autonomous driving, surveillance, and environmental monitoring.

5.4 Proposed Rain-induced Noise Reduction: mmCLAE

The proposed noise reduction approach, mmCLAE, employs a convolutional LSTM autoencoder to remove rain-induced noise artefacts from mmWave radar frames. This section provides an overview of the architecture and training process for mmCLAE.

5.4.1 mmCLAE Architecture

The mmCLAE architecture is fundamentally a convolutional LSTM autoencoder, that essentially eliminates the rain-induced noise from the mmWave radar frame by compressing the radar frame into a rich latent representation and then reconstructing a radar frame from this latent vector. The architectural design is illustrated in Figure

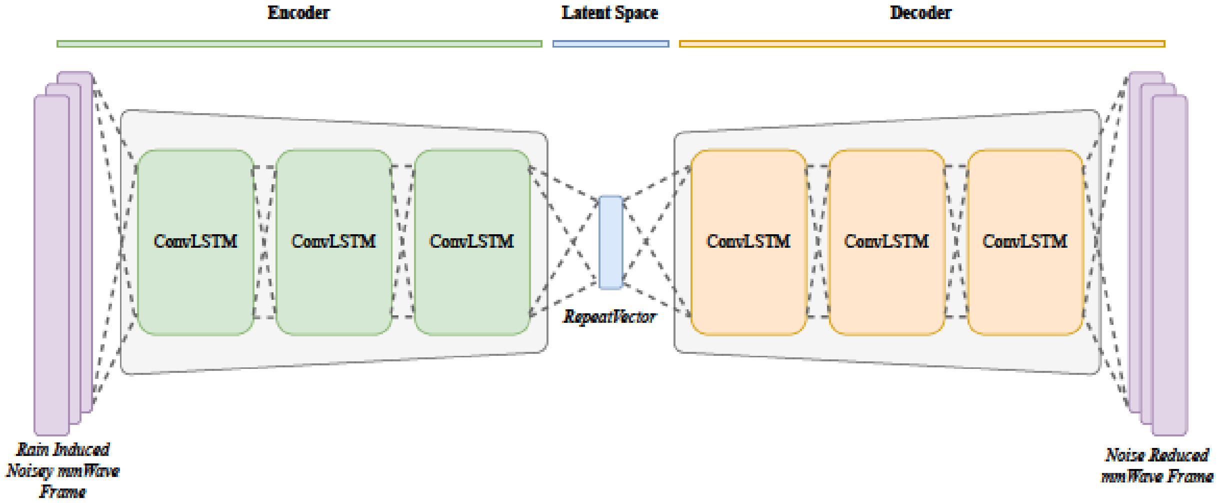


Figure 5.2: Architecture of mmCLAE for rain-induced noise reduction in mmWave radar.

5.2.

The model architecture consists of an encoder and decoder, each consisting of 3 Convolutional Long Short-Term Memory (ConvLSTM) layers. The encoder ConvLSTM layers all adopt a kernel size of 5 with the first ConvLSTM layer having 128 filters, which is then halved in each subsequent until the repeat vector layer. The decoder ConvLSTM layers are the reverse of the encoder, with the first ConvLSTM layer having 32 filters and doubling in each subsequent layer until the final output layer. The repeat vector layer is used to compress the radar frame into a rich latent representation, which is then used to reconstruct the radar frame in the decoder.

A key component of the architecture is the ConvLSTM layer which is a variant of the traditional LSTM layer that is designed to process spatial-temporal data by incorporating convolutional operations within the LSTM gates [120]. Our ConvLSTM layer consists of a ConvLSTM cell, a maximum pooling layer and a dropout layer. The intrinsic architecture of the ConvLSTM cell is illustrated in Figure 5.3.

Assuming an input sequence $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_T\}$, where each $\mathbf{X}_t \in \mathbb{R}^{H\times W\times C}$ are a 3D tensor with height H, width W, and C channels, the ConvLSTM cell calculates the hidden state $\mathbf{H}_t$ and cell state $\mathbf{C}_t$ at each time step t [121]. The hidden state ($\mathbf{H}_t$) in the cell is essentially the short-term memory of the network, retaining the current output based on the input and previous hidden state. The cell state ($\mathbf{C}_t$) on

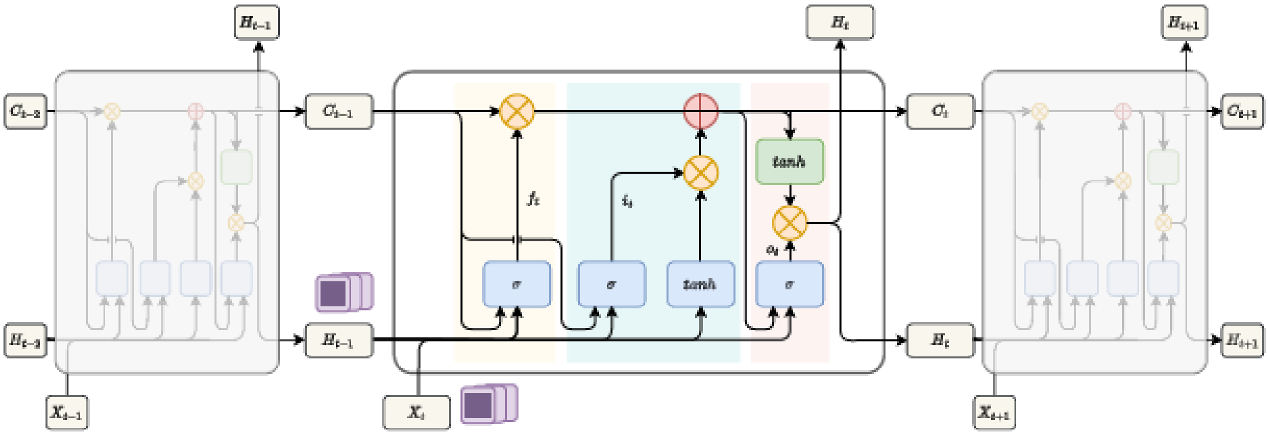


Figure 5.3: ConvLSTM cell architecture.

the other hand acts as the long-term memory, essentially storing information across longer sequences and enabling the network to maintain context over time t steps [122].

The ConvLSTM cell is essentially constructed with three main components:

1. **Forget Gate:** Responsible for controlling the extent in which the previous cell state is retained.

2. **Input Gate:** Responsible for controlling the extent in which the new input is used to update the new cell state.

3. **Output Gate:** Responsible for controlling the extent in which the new cell state is used to compute the hidden state.

The mathematical foundation of these gates, along with the cell state and hidden state computations, is expressed through the following set of equations [123]:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i), \tag{5.1}$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f), \tag{5.2}$$

$$g_t = \tanh(W_{xg} * X_t + W_{hg} * H_{t-1} + b_g), \tag{5.3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot g_t, \tag{5.4}$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_t + b_o), \tag{5.5}$$

$$H_t = o_t \odot \tanh(C_t), \tag{5.6}$$

where i_t, f_t, o_t, and g_t are the input gate, forget gate, output gate, and cell input activation, respectively. W_{xi}, W_{xf}, W_{xo}, and W_{xg} are the convolutional kernels for the input X_t. W_{hi}, W_{hf}, W_{ho}, and W_{hg} are the convolutional kernels for the hidden state H_{t-1}. b_i, b_f, b_o, and b_g are the bias terms. σ is the sigmoid activation function, tanh is the hyperbolic tangent activation function, $\odot$ denotes the Hadamard product (element-wise multiplication) and lastly $*$ denotes the convolution operation.

5.4.2 mmCLAE Training Process

One of the key design advantages of the mmCLAE network is the fact that it can be trained in an unsupervised manner, where the network is trained to minimise the MSE between the input and reconstructed radar frames. This is an important point when considering the nature and difficulty of curating a large dataset of mmWave radar frames with varying levels of rain-induced noise. Instead, the network is trained on a large dataset of "normal" radar signals, where it learns to reconstruct the input signal without the rain-induced noise. Therefore, when exposed to radar frames with rain-induced noise, the network fundamentally removes these artefacts as their features are not preserved in the latent representation. The objective function for mmCLAE is defined as:

$$J(\theta) = \frac{1}{2N} \sum_{n=1}^{N} (X_n - \hat{X}_n)^2 + \lambda ||\theta||^2, \tag{5.7}$$

where θ represents the model parameters, N is the total number of training samples, and λ is the regularisation strength. The MSE is minimised between the input radar frame X_n and the reconstructed frame $\hat{X}_n$, while also penalising the model complexity through the regularisation term $||\theta||^2$.

The autoencoder is trained using backpropagation with Adam optimiser [124] and a batch size of 128. The learning rate is set to 0.001, and the maximum number of epochs is 100.

Hyperparameter Tuning

The hyperparameters discussed to this point for mmCLAE were used as a starting point for the base model trained. The performance of convolutional LSTM autoencodrs heavily depends on the configuration of these hyperparameters. In order to

Algorithm 8 mmCLAE Bayesian Optimisation for Hyperparameter Tuning

```
Input: Initial hyperparameter set $\mathcal{H}_0$, objective function $J(\theta)$, maximum iterations $T$, convergence threshold $\epsilon$
Output: Optimal hyperparameters $\mathcal{H}^*$
Initialise $\mathcal{H} \leftarrow \mathcal{H}_0$
Initialise $J_{best} \leftarrow \infty$
for $t = 1$ to $T$ do
    Evaluate $J(\mathcal{H}_t)$
    if $J(\mathcal{H}_t) < J_{best}$ then
        $J_{best} \leftarrow J(\mathcal{H}_t)$
        $\mathcal{H}^* \leftarrow \mathcal{H}_t$
    end if
    Update the probabilistic model of $J(\theta)$    ▷ Eq. 5.7
    Select next hyperparameters $\mathcal{H}_{t+1}$ using acquisition function $U(\mathcal{H})$    ▷ Eq. 5.8
    if $|J_{best}^{(t)} - J_{best}^{(t-1)}| < \epsilon$ then    ▷ Eq. 5.9
        break
    end if
end for
return $\mathcal{H}^*$
```

further improve the performance of mmCLAE, we attempt to adopt an algorithmic optimisation approach towards tuning the model hyperparameters.

In order to achieve this we implement a Bayesian optimisation algorithm, which is a probabilistic model-based technique that attempts to converge on the optimal set of hyperparameters by minimising the objective function. The algorithm does this by constructing a probabilistic model of the objective function and then using this model to select the next set of hyperparameters to evaluate [125]. This process is then continuously repeated iteratively until the optimal set of hyperparameters is found, we present this in Algorithm 8.

The hyperparameters $\mathcal{H}$ we tune for are:

- α: the learning rate;

- L: the number of layers in the convolutional LSTM network;
- F_l: the number of filters in each layer;
- K_l: the kernel size of the convolutional layers;
- E: the number of epochs.

To perform Bayesian optimisation, we use the acquisition function known as the Upper Confidence Bound (UCB) function. This acquisition function is mathematically expressed as:

$$U(\mathcal{H}_t \leftarrow \alpha^{(t)}, L^{(t)}, F_l^{(t)}, K_l^{(t)}, E^{(t)}) = \mu(\mathcal{H}_t) + \kappa\sigma(\mathcal{H}_t), \tag{5.8}$$

where for a given set of hyperparameter values $\mathcal{H}_t$; μ is the mean of the predicted objective function, σ is the standard deviation of the predicted objective function, and κ is the exploration parameter. The exploration parameter κ is used to balance between exploration and exploitation, where a higher value of κ will encourage more exploration of the hyperparameter space.

A simple convergence threshold was adopted as a stopping criterion for the hyperparameter optimisation process. The mathematical criterion for convergence is expressed as:

$$|J_{best}^{(t)} - J_{best}^{(t-1)}| < \epsilon, \tag{5.9}$$

where $J_{best}^{(t)}$ is the best objective function value at iteration t, $J_{best}^{(t-1)}$ is the best objective function value at iteration $t-1$, and ϵ is the convergence threshold.

Rainfall Simulator

In addition to collecting mmWave radar data from real-world scenarios, to evaluate mmCLAE, we also built a physical rainfall simulator to artificially collect mmWave radar data with varying levels of rain-induced noise. The rainfall simulator consists of a water pump, a water tank, and a series of jet nozzles that spray water droplets into the field of view of the mmWave radar. Additionally, we also installed and built a rain gauge, based on a tipping bucket sensor, to serve as the ground truth

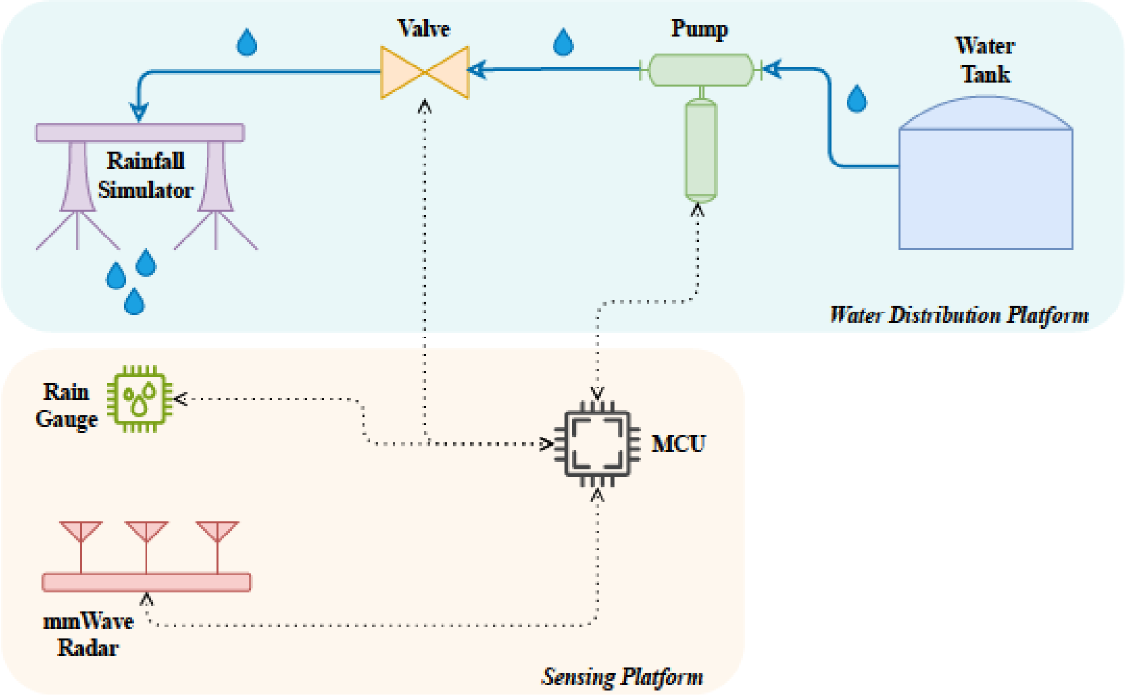


Figure 5.4: Architecture of the rainfall simulator for generating mmWave radar data with varying levels of rain-induced noise.

of the rainfall measurement. This means of ground truth was installed so that the same simulator could be utilised for our rain intensity classification model, discussed further in Section 5.5. Figure 5.4 and Figure 5.5 illustrate the setup of the rainfall simulator.

5.5 Proposed Sensing Method: CNN Rainfall Intensity Classification

In this section, we propose a sensing method for rain intensity classification using a CNN. The CNN-based approach is designed to classify the rain intensity based on the mmWave radar signal features extracted from the noisy and non-noisy regions. This approach leverages the high-frequency characteristics of mmWave radar to capture detailed precipitation features, which is then used to estimate rainfall intensity.

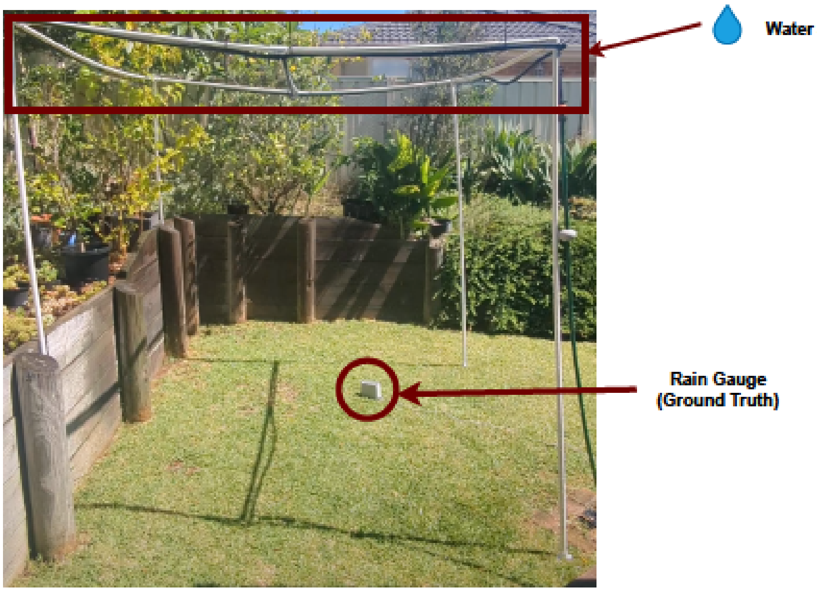


Figure 5.5: Physical setup of the rainfall simulator for generating mmWave radar data with varying levels of rain-induced noise.

5.5.1 Rainfall Classification Architecture

The proposed rainfall classification model is a CNN designed to classify rain intensity based on mmWave radar signal features. The architectural design of the model is illustrated in Figure 5.6.

The input to the model is a feature vector derived from the delta of the raw range-Doppler heatmap and the mmCLAE heatmap, as well as the raw heatmap vector. The model processes this input through several layers to classify the rain intensity.

The architecture of the rainfall intensity classification model consists of the following layers:

- **Convolutional Layers:** A total of 4 convolutional layers are used. Each layer applies a set of learnable filters to the input, performing a dot product between the filter weights and the input region. This process extracts spatial features from the input data. The convolutional layers use Rectified Linear Unit (ReLU) activation functions to introduce non-linearity into the model. The computational complexity of each convolutional layer is $\mathcal{O}(F \times K^2 \times H \times W \times C)$, where F is the number of filters, K is the kernel size, $H \times W$ are the spatial dimensions, and C is the number of input channels.

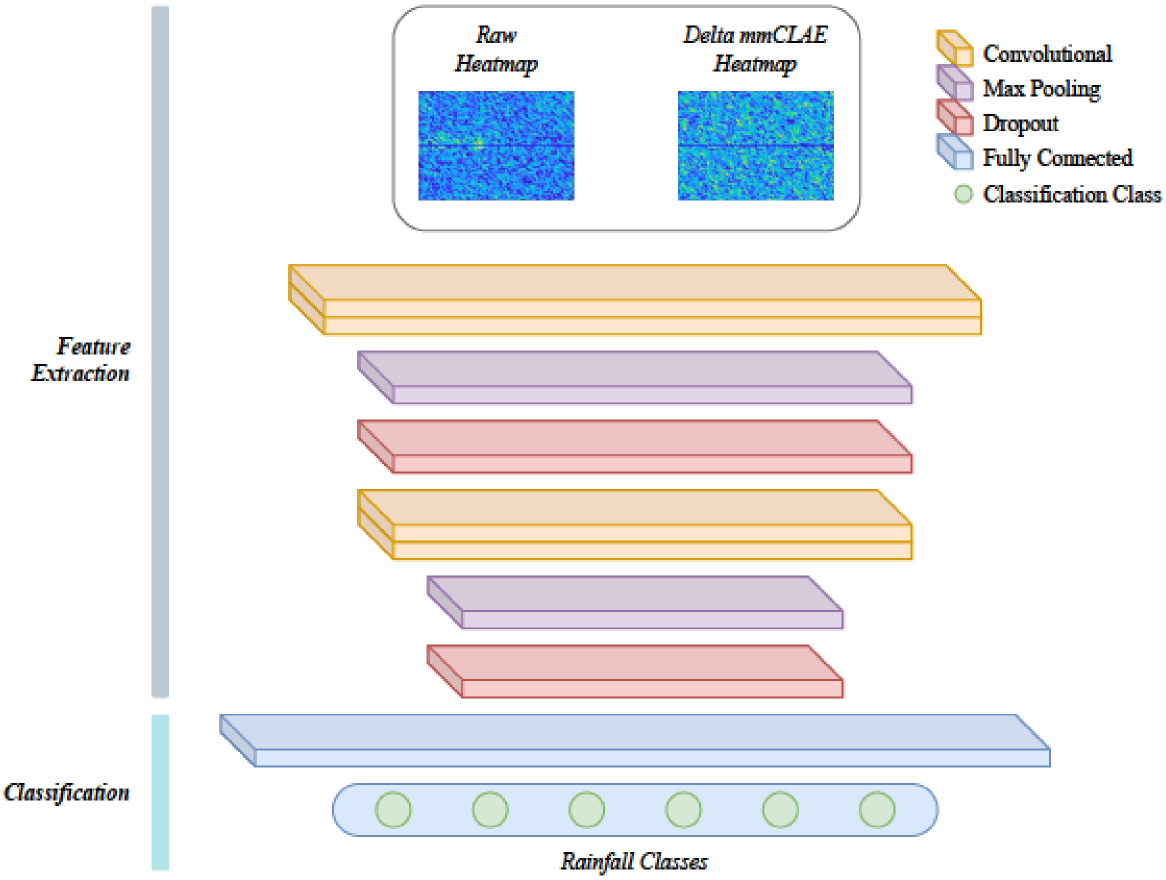


Figure 5.6: Architecture of the rainfall intensity classification model.

- **Max Pooling Layers:** Following each pair of convolutional layers, a max pooling layer is used. These layers reduce the spatial dimensions of the feature maps, which helps in reducing the number of parameters and computational load, as well as mitigating overfitting. Max pooling layers take the maximum value from each region of the feature map, preserving the most important features. The computational complexity is $\mathcal{O}(H \times W \times C)$ for pooling operations.

- **Dropout Layers:** Dropout layers are used after the max pooling layers and Fully Connected (FC) layers. These layers randomly set a fraction of input units to zero at each update during training time, which helps prevent overfitting by ensuring that the model does not rely too heavily on any single given feature.

- **FC Layers:** The data is then flattened and fed into FC layers. These layers perform high-level reasoning about the input data with complexity $\mathcal{O}(N_{in} \times N_{out})$ for matrix multiplication operations. The final FC layer uses a softmax

Table 5.1: Rainfall classification types.

Type	Rainfall
None	$< 0.5mm/h$
Light	$0.5 - 2mm/h$
Medium	$2 - 5mm/h$
Heavy	$5 - 10mm/h$
Very Heavy	$10 - 20mm/h$
Extreme	$> 20mm/h$

activation function to classify the input into a $[1 \times N]$ matrix, where N is the number of rain intensity classes.

The output of this classifier, as seen in Table 5.1, is a classification of rain type, which could be None, Light, Medium, Heavy, Very Heavy, or Extreme. This architecture ensures that the classifier can effectively learn from the range-Doppler heatmaps and accurately classify the rainfall intensity.

Loss Function

The rainfall intensity classification model implements a Weighted Cross-Entropy (WCE) function to ultimately handle class imbalance in the variations of collected rainfall data, optimised with the Adam optimiser. The mathematical formulation of the WCE loss function is:

$$L = -\frac{1}{N} \sum_{i=1}^{N} w_{c_i} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right], \tag{5.10}$$

where N is the total number of training samples, w_{c_i} is the weight associated with the class c_i of the i^{th} sample, y_i is the true label for the i^{th} sample, which is a binary indicator (0 or 1) if class label c_i is the correct classification for sample i, and $\hat{y}_i$ is the predicted probability of the i^{th} sample belonging to class c_i.

The WCE loss function alters the contribution of each sample to the loss based on the class weights, therefore it gives more importance to underrepresented classes. This is particularly useful in our case, as it ensures that the model pays adequate

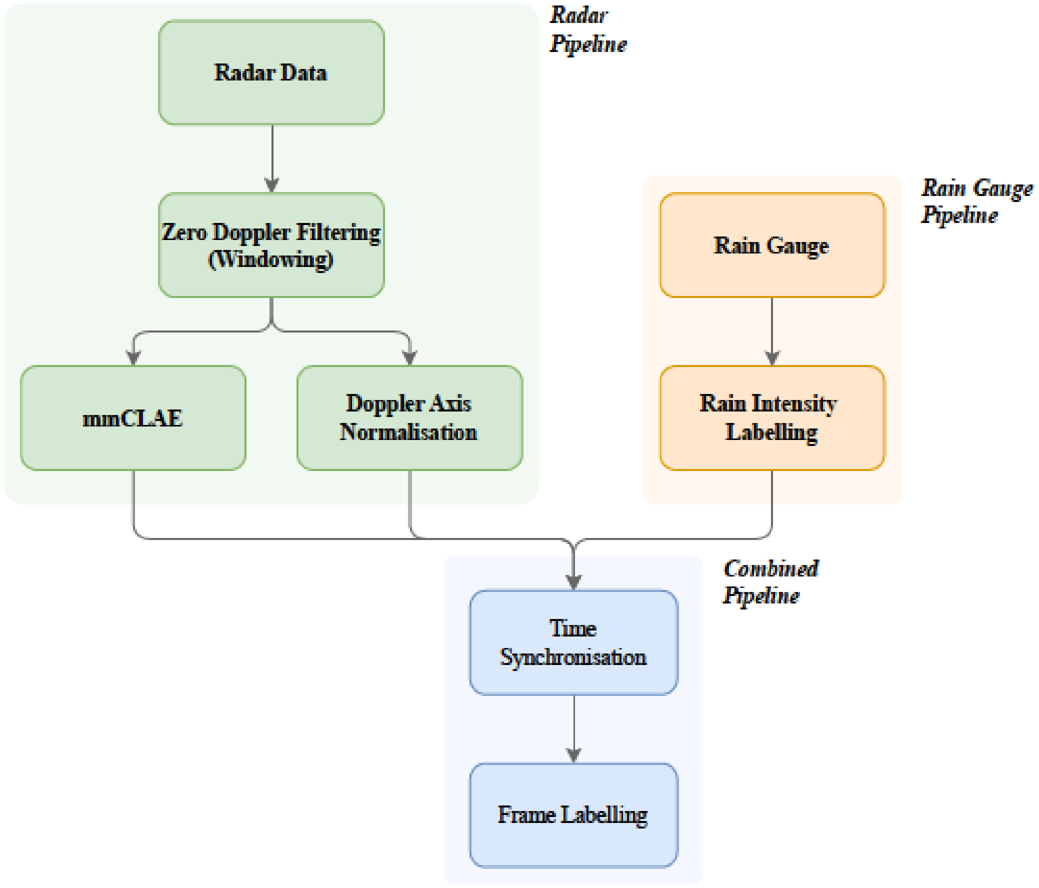


Figure 5.7: Training pipeline for rainfall intensity classification.

attention to correctly predicting heavy rainfall events, which are less frequent but critical for accurate rainfall intensity estimation.

The weights w_{c_i} are calculated using the following mathematical relationship:

$$w_{c_i} = \frac{N}{|C|N_{c_i}}, \tag{5.11}$$

where $|C|$ is the total number of classes, as seen in Table 5.1 and N_{c_i} is the number of samples in class c_i.

5.5.2 Rainfall Classification Training Process

The training pipeline for the rainfall intensity classification system consists of three main components: the radar pipeline, the rain gauge pipeline, and the combined pipeline.

In the radar pipeline, data acquisition is the first step where the radar data is collected. The mmWave radar system captures the back scattered signals from the raindrops. This raw radar data is then processed using FFT for spectral analysis, and pulse compression for improving range resolution. The processed data undergoes standard noise filtering to remove unwanted general noise and interference in the radar frame. At this stage the radar frame is additionally run through mmCLAE to remove the rain induced noise artefacts. Finally, the radar data is normalised to ensure consistency and to prepare it for the next steps.

The rain gauge pipeline starts with data collection where rainfall data is collected from the rain gauge sensor. This data is then cleaned to remove any anomalies or errors in the readings. The cleaned data is resampled to match the temporal resolution of the radar data. Similar to the radar pipeline, the rain gauge data is also normalised to ensure it's on the same scale as the radar data.

The combined pipeline begins with data integration where the processed radar and rain gauge data are integrated. This involves the fusion of the radar and rain gauge data, where each is combined to create a comprehensive dataset. An important aspect of this integration is time synchronisation, ensuring that the radar and rain gauge data align correctly in time. This is crucial because the radar and rain gauge may not record data at the exact same time intervals. We solve this time synchronisation challenge by following a similar methodology to what we presented in Chapter 4. After the data integration and time synchronisation, relevant features for rainfall intensity prediction are extracted from the integrated data in the form of range-Doppler heatmaps. Finally, the dataset is split into training and testing sets for model development and evaluation. This process is further detailed in Algorithm 9.

5.6 Experimental Results

In this section, we present the experimental results of the joint tracking and sensing framework that utilises mmCLAE and our CNN rainfall intensity classification network, demonstrating the effectiveness of the proposed approaches in improving system performance and robustness to noise and rain artefacts.

Algorithm 9 Rainfall Classification Training Process

```
Input: Raw radar data R, Rain gauge data G
Output: Trained CNN model M
Radar Pipeline:
Collect raw radar data R
Process radar data using FFT and pulse compression
Apply noise filtering to radar data
Apply mmCLAE to remove rain-induced noise artefacts
Normalise radar data
Rain Gauge Pipeline:
Collect rain gauge data G
Clean rain gauge data
Resample rain gauge data to match radar data temporal resolution
Normalise rain gauge data
Combined Pipeline:
Integrate processed radar data R and rain gauge data G
Synchronise radar and rain gauge data in time
Extract features from integrated data
Split dataset into training and testing sets
Training:
Initialise CNN model M
Define WCE loss function and Adam optimiser
for each epoch do
    Train CNN model M on training set
    Validate CNN model M on validation set
end for
Evaluation:
Evaluate trained CNN model M on testing set
Calculate precision, recall, and F1 score for each class
return Trained CNN model M
```

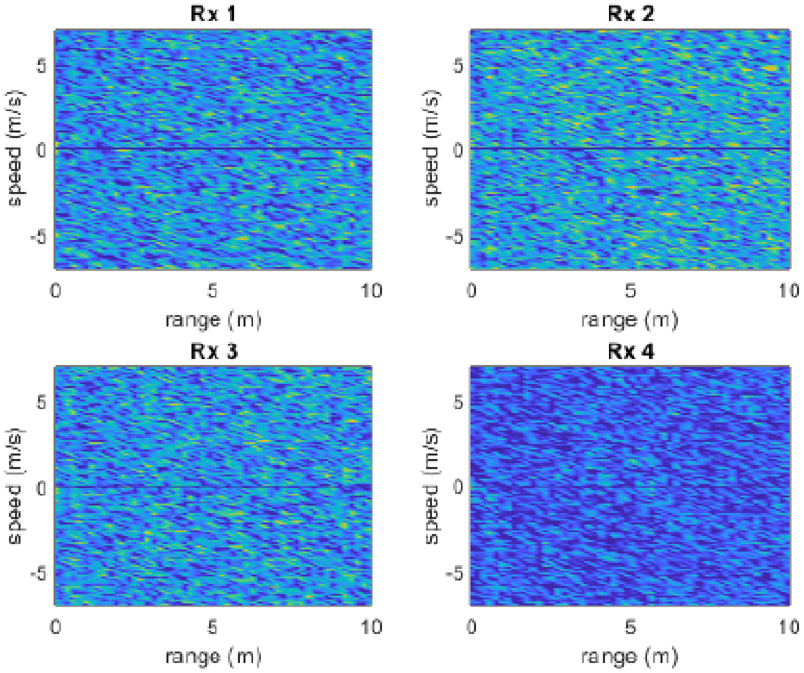


Figure 5.8: Range-Doppler heatmap illustrating no rainfall.

5.6.1 Multi-Object Tracking Performance Comparison

To evaluate the effectiveness of our proposed noise reduction approach on multi-object tracking performance, we compare the results with a typical standalone EKF Bayesian method in both no rain and rain. We use a dataset containing 500 mmWave radar signals with multiple objects (1-5) in varying environments. The signals are divided into training (70%), validation (15%), and testing (15%) sets. Figures 5.8 to 5.12 illustrate range-Doppler heatmaps that were collected, highlighting varying levels of rain intensity with and without the presence of people walking in the field of view.

We compare the performance of our proposed approach with that of the following methods:

1. Standalone EKF - No Rain;
2. Standalone EKF - Rain.

The results are presented in Table 5.2.

The results show that our proposed approach significantly outperforms the stan-

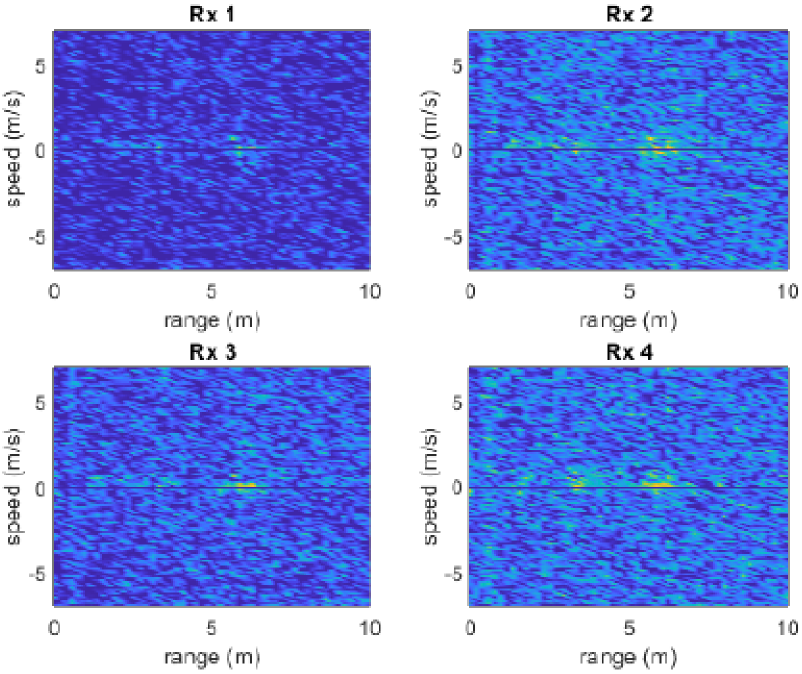


Figure 5.9: Range-Doppler heatmap illustrating light rainfall.

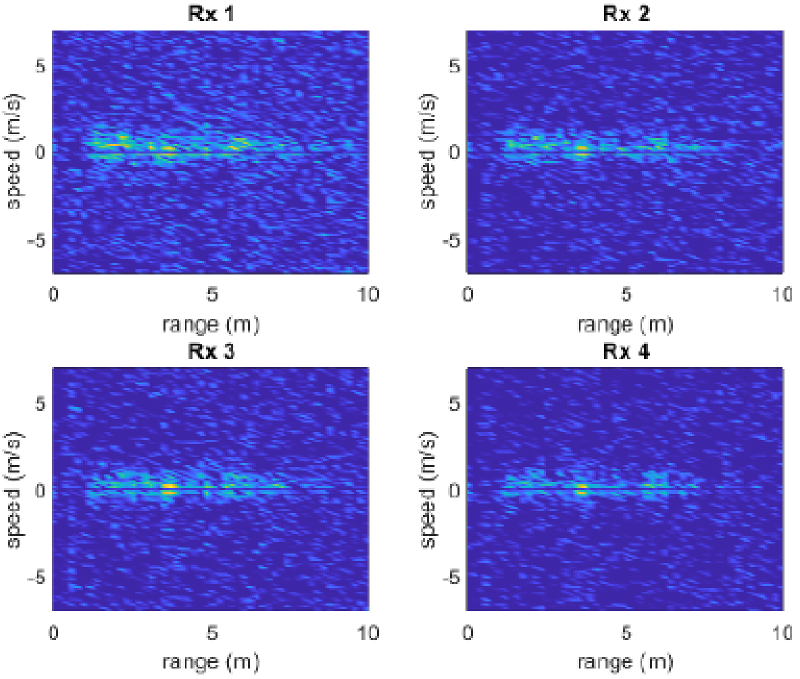


Figure 5.10: Range-Doppler heatmap illustrating medium rainfall.

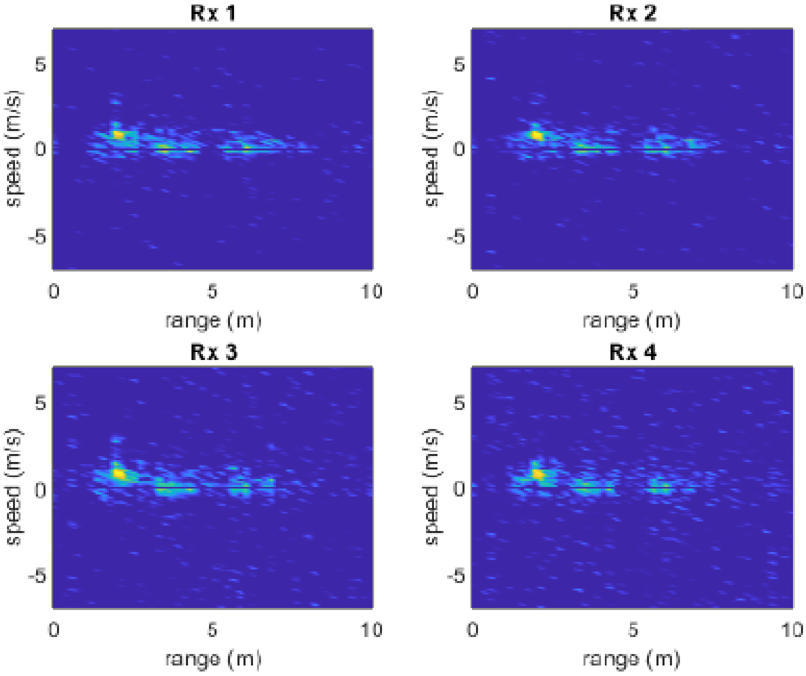


Figure 5.11: Range-Doppler heatmap illustrating medium rainfall with a single person walking.

Table 5.2: mmCLAE multi-object tracking performance comparison.

Method	RMSE	MAE
Standalone EKF - Rain	2.15 m	1.65 m
mmCLAE - Rain	0.25 m	0.19 m

dalone EKF Bayesian method when tracking objects in rain conditions. Our approach shows an improvement of 71% in Root Mean Squared Error (RMSE) and 69% in Mean Absolute Error (MAE) compared to the EKF Bayesian method.

Figure 5.13 provides a visual comparison of the multi-object tracking performance between the different approaches to tracking a person following a predetermined simple movement path. The figure presents tracking points as a scatter plot with three distinct sets of data points: the baseline EKF no rain tracking positions, the EKF-based tracking under rainy conditions, and the mmCLAE-enhanced tracking under the same rainy conditions. The EKF rain tracking points demonstrate significant scatter and deviations from the baseline EKF no rain performance, exhibiting erratic behaviour and substantial positional errors. In contrast, the mmCLAE rain tracking

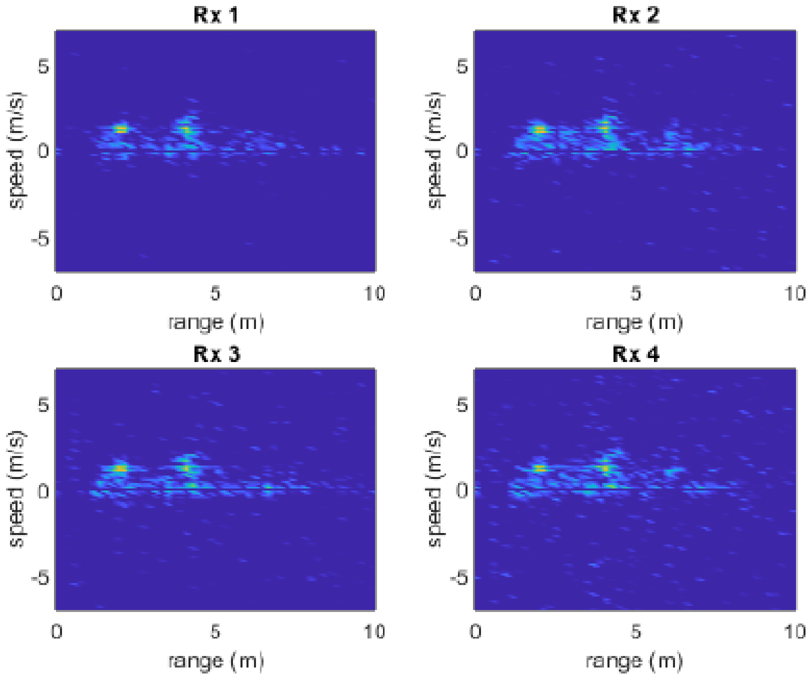


Figure 5.12: Range-Doppler heatmap illustrating medium rainfall with two people walking.

points cluster closely around the baseline EKF no rain path with minimal deviation, demonstrating the effectiveness of the noise reduction approach in maintaining tracking accuracy even under adverse weather conditions. The visual comparison clearly illustrates how rain-induced noise severely degrades conventional EKF tracking performance, while the mmCLAE approach successfully mitigates these effects.

These results clearly demonstrate the effectiveness of our proposed mmCLAE noise reduction approach in improving the accuracy and robustness of multi-object tracking systems in mmWave radar environments with and without rain. The improved multi-object tracking performance can be attributed to the effective noise reduction capabilities of our convolutional LSTM autoencoder, which enables better object detection and tracking. This, in turn, enhances the overall system robustness to noise and rain artefacts.

5.6.2 Evaluation Metrics

We evaluate the performance of the proposed tracking procedures using two key metrics: RMSE and MAE.

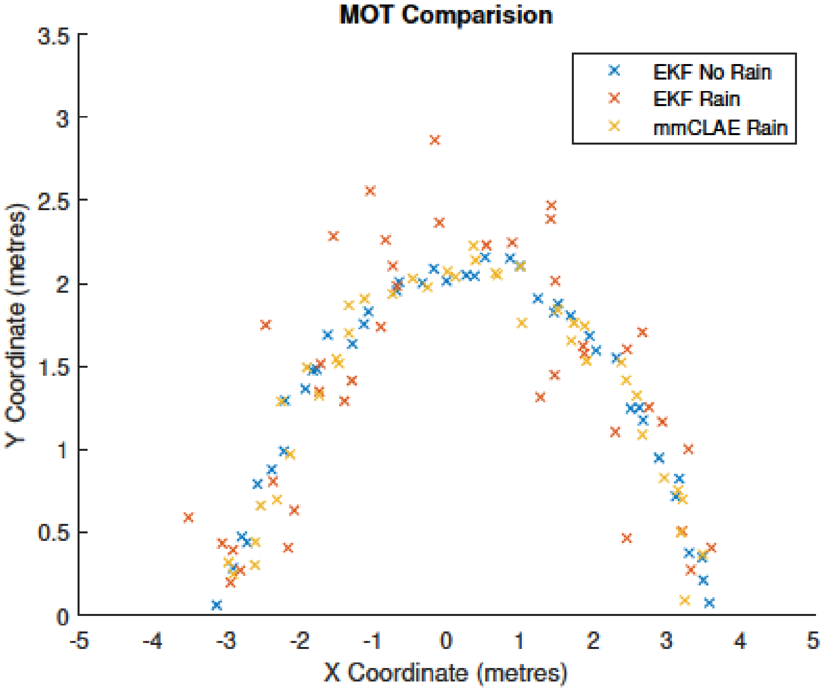


Figure 5.13: Multi-object tracking mmCLAE and standalone EKF comparison of a simple movement path.

RMSE

The RMSE measures the average magnitude of the errors between the predicted and ground truth positions, expressed as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{x}_n - x_n)^2}, \tag{5.12}$$

where $\hat{x}_n$ and x_n are the predicted and ground truth positions at radar frame n, respectively, and N is the total number of radar frames. An important characteristic of RMSE is that it is sensitive to large errors, making it useful for highlighting significant deviations.

MAE

The MAE measures the average absolute difference between the predicted and ground truth positions. The mathematical formulation is expressed as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N}|\hat{x}_n - x_n|, \tag{5.13}$$

Table 5.3: Rain intensity classification evaluation metrics.

Class	Precision	Recall	F1 Score
No Rain	90.12%	89.85%	0.8995
Light Rain	83.45%	81.3%	0.8236
Moderate Rain	79.43%	78.55%	0.7896
Heavy Rain	81.52%	80.75%	0.8110
Very Heavy Rain	91.35%	90.55%	0.9092
Extreme Rain	88.53%	82.15%	0.8519

where $\hat{x}_n$ and x_n are the predicted and ground truth positions at frame n, respectively, and N is the total number of radar frames. MAE provides a straightforward interpretation of the average error magnitude, making it less sensitive to outliers when compared to RMSE.

5.6.3 Rain Intensity Classification Evaluation

The evaluation metrics used to assess the performance of our classification model are precision, recall, and F1 score for each class (No Rain, Light Rain, Moderate Rain, Heavy Rain, Very Heavy Rain, and Extreme Rain). The results, presented in Table 5.3, show that our model achieves high levels of precision and recall for most classes, with an average precision of 85.73%, recall of 83.86%.

The F1 score provides a balanced view of both precision and recall, and our model achieves an average F1 score of 0.8386 across all classes. This suggests that our model is able to achieve a good balance between precision and recall, ultimately demonstrating its reliability in classifying rainfall conditions.

Additionally, the accuracy and loss graphs demonstrate the stable learning rate of the model. This can be seen in Figure 5.14 and Figure 5.15 respectively.

Overall, the evaluation metrics demonstrate that our classification model performs well on most classes, with high levels of accuracy and reliability.

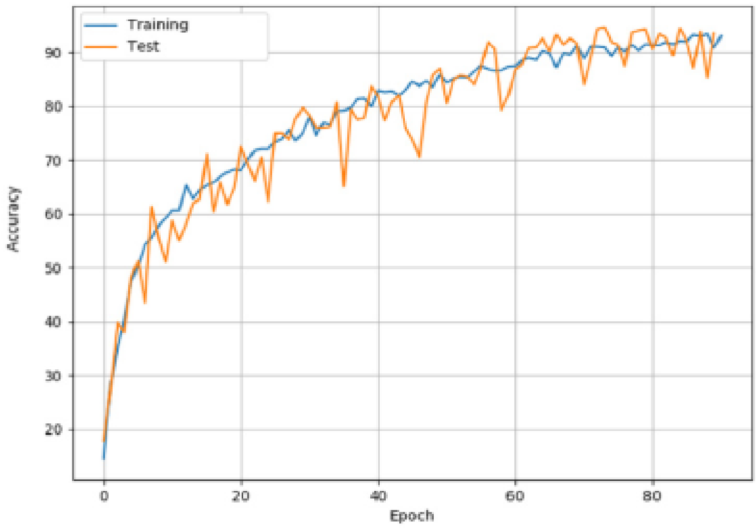


Figure 5.14: Accuracy graph of the rainfall intensity classification model.

5.7 Conclusion

In this chapter, we demonstrated the effectiveness of the proposed approaches in improving mmWave tracking performance and robustness to noise and rain artefacts. The results show that mmCLAE is superior to state-of-the-art methods in terms of noise reduction, while the CNN-based sensing method produces above 80% accuracy in rain intensity classification.

While our proposed approaches have demonstrated promising results, there are several perspectives that are worth mentioning for further exploration and research:

- **Adaptation to different environmental conditions:** Our approach has been specifically designed for rain noise reduction and classification. However, it is essential to investigate the generalised nature of our method to other environmental conditions such as fog, snow, or hail.

- **Integration with advanced sensor fusion techniques:** Combining mmWave radar data with other sensing modalities (e.g., cameras, LiDAR) could further enhance the robustness and accuracy of the system. This would require exploring novel approaches to fuse data from different sensors and develop joint processing pipelines, similar to that presented in Chapter 4.

- **Real-time processing and latency optimisation:** In applications such as autonomous vehicles or surveillance systems, real-time processing is a require-

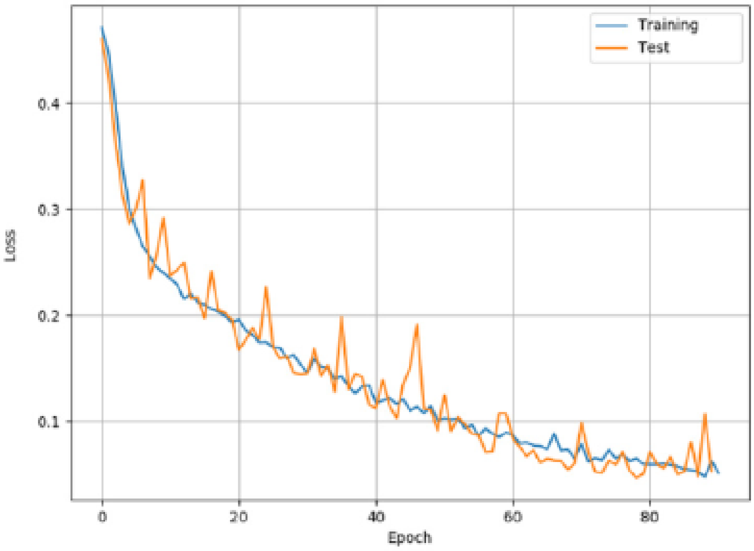


Figure 5.15: Loss graph of the rainfall intensity classification model.

ment. Therefore, optimising our approach for real-time performance while maintaining its effectiveness in adverse environmental conditions is a valuable direction to explore in future research.

The potential applications of this technology are vast, particularly in the context of autonomous vehicles and other radar based sensing platforms. By integrating our mmCLAE tracking system and rainfall intensity estimation module, we can improve the accuracy and reliability of navigation in rainy conditions, ultimately improving passenger safety and reducing the risk of accidents.

Chapter 6

Conclusions and Future Work

This chapter concludes this thesis with a summary of the key findings and contributions presented in this thesis, along with recommendations for future work in the field of mmWave radar multi-object tracking and sensing. The chapter is structured as follows: Section 6.1 provides a detailed summary of the significant contributions made by this thesis, highlighting the advancements in environmental characterisation, sensor fusion, and joint tracking and sensing. Following this, Section 6.2 outlines the recommended future work, categorising potential research directions into advanced environmental characterisation, improved sensor fusion techniques, enhanced noise reduction methods, and real-world deployment.

6.1 Summary of Contributions

This thesis has made several significant contributions to the field of mmWave radar multi-object tracking and sensing. The research presented in this thesis addresses key challenges in the domain, including environmental characterisation, sensor fusion, and noise reduction in adverse weather conditions through joint tracking and sensing. By developing novel frameworks and methodologies, this work enhances the robustness, accuracy, and applicability of mmWave multi-object tracking systems in complex and dynamic environments.

One of the primary contributions of this thesis is the development of a framework for extracting and utilising environmental characteristics from multi-object trajectory data. This framework involves the creation of regional activity heatmaps and the classification of entry and exit points using CNN. By projecting these classified points onto the multi-object tracking plane, the proposed approach provides a foundational basis for improving mmWave multi-object tracking performance through a better understanding of the observed environment. This contribution addresses the challenges posed by occlusions and disturbances, leading to more reliable tracking and reduced errors in dynamic environments.

Another significant contribution is the integration of mmWave radar and camera data for enhanced tracking and classification capabilities. The proposed sensor fusion framework leverages the strengths of both modalities to address the challenges associated with labelling and training deep learning models for radar data. By fusing radar and camera data, the framework displays accurate classification and tracking of objects in various environments. This contribution not only improves the robustness and accuracy of mmWave multi-object tracking systems but also provides a novel approach to automated labelling of mmWave radar data using camera information.

Lastly, we present in this thesis a comprehensive approach to enhancing mmWave multi-object tracking systems in adverse weather conditions, particularly focusing on rain-induced noise reduction and rain intensity classification. The proposed mm-CLAE architecture, effectively removes rain-induced artefacts from mmWave signals, while the CNN-based rainfall intensity model jointly accurately classifies rain intensities. Through extensive experiments, these methods demonstrate significant improvements in mmWave multi-object tracking accuracy, providing a practical implementation that utilises deep learning techniques to address the challenges associated with adverse weather conditions.

In summary, this thesis makes substantial contributions to the field of mmWave radar multi-object tracking and sensing by addressing key challenges through innovative frameworks and methodologies. The research enhances the understanding of environmental characteristics, improves sensor fusion techniques, and mitigates the impact of adverse weather conditions on tracking performance. These contributions provide a solid foundation for future advancements in mmWave radar technologies, providing opportunities for more reliable and accurate joint multi-object tracking and sensing systems.

6.2 Recommended Future Work

The research presented in this thesis opens several streams for future work that can further enhance the capabilities and applications of mmWave radar multi-object tracking and sensing systems. An attempt to illustrate the constellation of potential future directions is presented in the mind map provided in Figure 6.1.

These potential directions for future research can be categorised into four main areas: advanced environmental characterisation, improved sensor fusion techniques, enhanced noise reduction methods, and real-world deployment and validation. The remainder of this section elaborates on each of these areas and provides specific recommendations for future work.

6.2.1 Advanced Environmental Characterisation

Future work can focus on developing more sophisticated methods for environmental characterisation using mmWave radar data. One potential direction is to explore the use of advanced machine learning techniques, such as Generative Adversarial Networks (GANs) and reinforcement learning, to dynamically adapt the environmental models based on real-time data. Additionally, incorporating data from other sensors, such as LiDAR and ultrasonic sensors, can provide a more comprehensive understanding of the environment, potentially leading to further improvements in tracking accuracy. Another area of interest is the development of algorithms that can automatically detect and adapt to changes in the environment, such as the movement of furniture or the presence of new obstacles, to maintain high tracking performance in dynamic settings.

6.2.2 Improved Sensor Fusion Techniques

The integration of mmWave radar and camera data demonstrated in this thesis can be extended to include additional sensor modalities, such as thermal cameras and

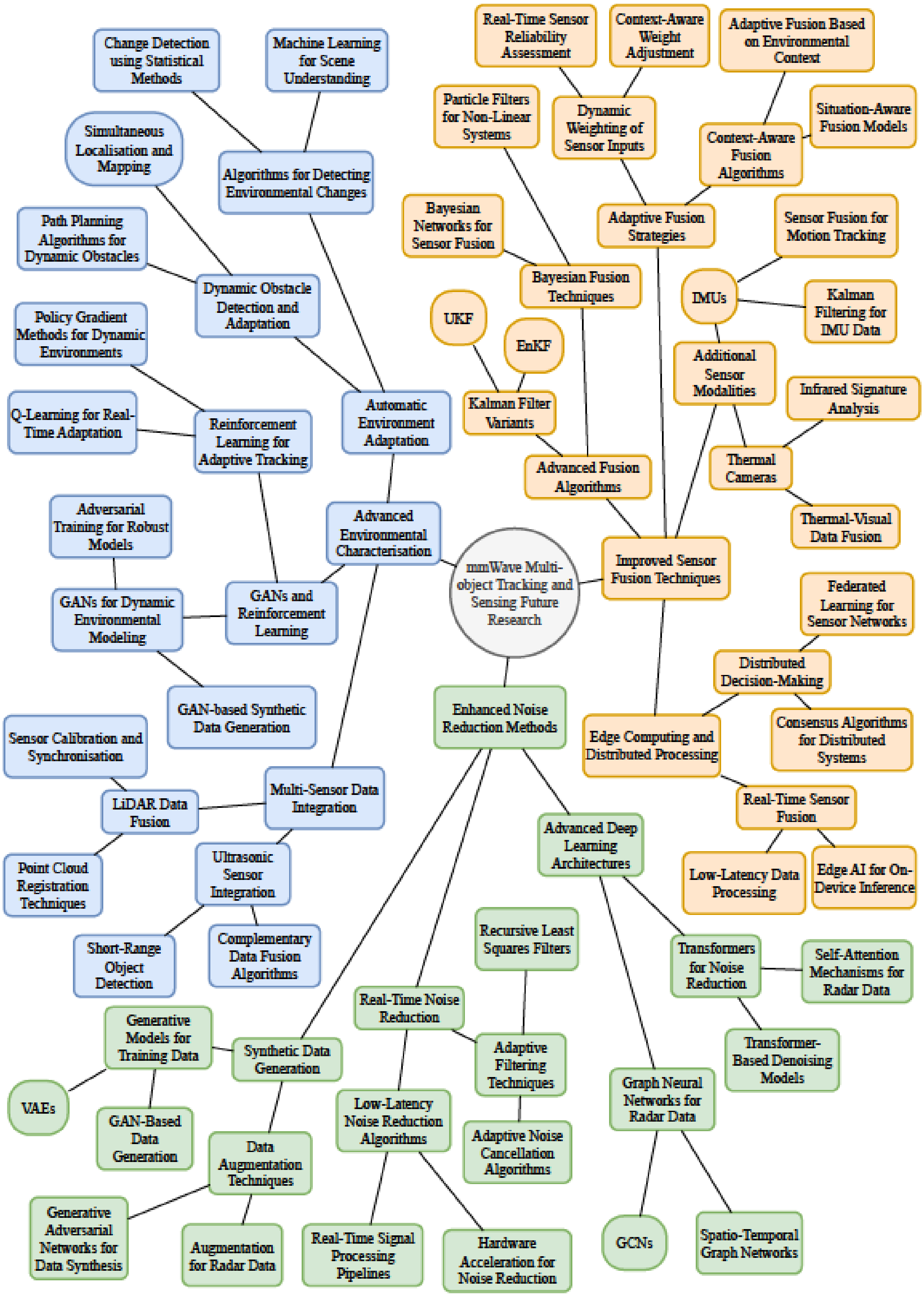

mmWave Multi-object Tracking and Sensing Future Research
Change Detection using Statistical Methods
Machine Learning for Scene Understanding
Simultaneous Localisation and Mapping
Algorithms for Detecting Environmental Changes
Path Planning Algorithms for Dynamic Obstacles
Dynamic Obstacle Detection and Adaptation
Policy Gradient Methods for Dynamic Environments
Automatic Environment Adaptation
Q-Learning for Real-Time Adaptation
Reinforcement Learning for Adaptive Tracking
Advanced Environmental Characterisation
Adversarial Training for Robust Models
GANs for Dynamic Environmental Modeling
GANs and Reinforcement Learning
GAN-based Synthetic Data Generation
Sensor Calibration and Synchronisation
LiDAR Data Fusion
Multi-Sensor Data Integration
Point Cloud Registration Techniques
Ultrasonic Sensor Integration
Short-Range Object Detection
Complementary Data Fusion Algorithms
Real-Time Sensor Reliability Assessment
Context-Aware Weight Adjustment
Adaptive Fusion Based on Environmental Context
Particle Filters for Non-Linear Systems
Dynamic Weighting of Sensor Inputs
Situation-Aware Fusion Models
Context-Aware Fusion Algorithms
Bayesian Networks for Sensor Fusion
Adaptive Fusion Strategies
Sensor Fusion for Motion Tracking
Bayesian Fusion Techniques
UKF
IMUs
Kalman Filtering for IMU Data
EnKF
Additional Sensor Modalities
Kalman Filter Variants
Infrared Signature Analysis
Advanced Fusion Algorithms
Thermal Cameras
Thermal-Visual Data Fusion
Improved Sensor Fusion Techniques
Federated Learning for Sensor Networks
Distributed Decision-Making
Enhanced Noise Reduction Methods
Edge Computing and Distributed Processing
Consensus Algorithms for Distributed Systems
Real-Time Sensor Fusion
Advanced Deep Learning Architectures
Low-Latency Data Processing
Edge AI for On-Device Inference
Recursive Least Squares Filters
Transformers for Noise Reduction
Self-Attention Mechanisms for Radar Data
Real-Time Noise Reduction
Transformer-Based Denoising Models
Generative Models for Training Data
Synthetic Data Generation
Adaptive Filtering Techniques
Graph Neural Networks for Radar Data
VAEs
GAN-Based Data Generation
Low-Latency Noise Reduction Algorithms
Adaptive Noise Cancellation Algorithms
Data Augmentation Techniques
Generative Adversarial Networks for Data Synthesis
Augmentation for Radar Data
Real-Time Signal Processing Pipelines
Hardware Acceleration for Noise Reduction
GCNs
Spatio-Temporal Graph Networks

Figure 6.1: Mind map illustrating potential future research opportunities in mmWave radar multi-object tracking and sensing.

IMUs. Investigating the development of more advanced algorithms for sensor fusion, such as Ensemble Kalman Filters (EnKFs) and Unscented Kalman Filters (UKFs), could potentially leverage the complementary strengths of each sensor to achieve even higher levels of accuracy in object tracking and classification. Furthermore, exploring the use of edge computing and distributed processing techniques, such as parallel sensor data processing across multiple edge nodes and federated learning for collaborative model updates, can enable real-time sensor fusion and decision-making, which is critical for applications such as autonomous vehicles and robotics. Another potential direction is the development of adaptive fusion strategies that can dynamically adjust the weighting of different sensor inputs based on the current environmental conditions and task requirements.

6.2.3 Enhanced Noise Reduction Methods

While the proposed mmCLAE has shown promising results in reducing rain-induced noise, there is still room for improvement in noise reduction techniques for mmWave radar systems. Future work can explore the use of more advanced deep learning architectures, such as transformers, Graph Convolutional Networks (GCNs), and Variational Autoencoders (VAEs), to better capture the complex relationships and patterns in the radar data. Additionally, developing methods for real-time noise reduction and adaptive filtering can further enhance the robustness of mmWave radar systems in adverse weather conditions. Lastly, another potential avenue to investigate is the use of synthetic data generation and data augmentation techniques to improve the training of noise reduction models, especially in scenarios where labelled data is scarce.

Bibliography

[1] S. Björklund, T. Johansson, and H. Petersson, "Evaluation of a micro-Doppler classification method on mm-wave data," in *2012 IEEE Radar Conference*, 2012, pp. 0934–0939. [Online]. Available: https://doi.org/10.1109/RADAR.2012.6212271.

[2] T. Gu, Z. Fang, Z. Yang, P. Hu, and P. Mohapatra, "Mmsense: Multi-person detection and identification via mmWave sensing," in *Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*, ser. mmNets'19, Los Cabos, Mexico: Association for Computing Machinery, 2019, pp. 45–50. [Online]. Available: https://doi.org/10.1145/3349624.3356765.

[3] Z. Shen, J. Nunez-Yanez, and N. Dahnoun, "Advanced millimeter-wave radar system for real-time multiple-human tracking and fall detection," *Sensors*, vol. 24, no. 11, 2024. [Online]. Available: https://doi.org/10.3390/s24113660.

[4] T. Zhou, M. Yang, K. Jiang, H. Wong, and D. Yang, "Mmw radar-based technologies in autonomous driving: A review," *Sensors*, vol. 20, no. 24, 2020. [Online]. Available: https://doi.org/10.3390/s20247283.

[5] L. Reddy Cenkeramaddi, J. Bhatia, A. Jha, S. Kumar Vishkarma, and J. Soumya, "A survey on sensors for autonomous systems," in *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2020, pp. 1182–1187. [Online]. Available: https://doi.org/10.1109/ICIEA48937.2020.9248282.

[6] R. Liu, T. Yao, R. Shi, *et al.*, "Mission: mmWave radar person identification with RGB cameras," in *Proceedings of the 22nd ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '24, Hangzhou, China: As-

sociation for Computing Machinery, 2024, pp. 309–321. [Online]. Available: https://doi.org/10.1145/3666025.3699340.

[7] P. Zhao, C. X. Lu, J. Wang, *et al.*, "Human tracking and identification through a millimeter wave radar," *Ad Hoc Networks*, vol. 116, p. 102 475, 2021. [Online]. Available: https://doi.org/10.1016/j.adhoc.2021.102475.

[8] Z. Li, B. Chen, X. Chen, *et al.*, "Spiralspy: Exploring a stealthy and practical covert channel to attack air-gapped computing devices via mmWave sensing," Jan. 2022. [Online]. Available: https://doi.org/10.14722/ndss.2022.23023.

[9] Z. Li, F. Ma, A. S. Rathore, *et al.*, "WaveSpy: Remote and through-wall screen attack via mmWave sensing," in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 217–232. [Online]. Available: https://doi.org/10.1109/SP40000.2020.00004.

[10] C. Dowling, H. Larijani, M. Mannion, M. Marais, and S. Black, "Improving the accuracy of mmWave radar for ethical patient monitoring in mental health settings," *Sensors*, vol. 24, no. 18, 2024. [Online]. Available: https://doi.org/10.3390/s24186074.

[11] E. Sadeghi, K. Skurule, A. Chiumento, and P. Havinga, *Comprehensive mm-wave FMCW radar dataset for vital sign monitoring: Embracing extreme physiological scenarios*, 2024. [Online]. Available: https://arxiv.org/abs/2405.12659.

[12] M. Alizadeh, G. Shaker, J. C. M. D. Almeida, P. P. Morita, and S. Safavi-Naeini, "Remote monitoring of human vital signs using mm-wave FMCW radar," *IEEE Access*, vol. 7, pp. 54 958–54 968, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2912956.

[13] K. Guo, C. Liu, S. Zhao, J. Lu, S. Zhang, and H. Yang, "Design of a millimeter-wave radar remote monitoring system for the elderly living alone using WiFi communication," *Sensors*, vol. 21, no. 23, 2021. [Online]. Available: https://doi.org/10.3390/s21237893.

[14] Y. Wang, W. Wang, M. Zhou, A. Ren, and Z. Tian, "Remote monitoring of human vital signs based on 77-GHz mm-wave FMCW radar," *Sensors*, vol. 20, no. 10, 2020. [Online]. Available: https://doi.org/10.3390/s20102999.

[15] Z. Yang, P. H. Pathak, Y. Zeng, X. Liran, and P. Mohapatra, "Vital sign and sleep monitoring using millimeter wave," *ACM Trans. Sen. Netw.*, vol. 13, no. 2, Apr. 2017. [Online]. Available: https://doi.org/10.1145/3051124.

[16] N. Techaphangam and M. Wongsaisuwan, "Obstacle avoidance using mmWave radar imaging system," in *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2020, pp. 466–469. [Online]. Available: https://doi.org/10.1109/ECTI-CON49241.2020.9158273.

[17] K. Harlow, H. Jang, T. D. Barfoot, A. Kim, and C. Heckman, "A new wave in robotics: Survey on recent mmWave radar applications in robotics," *IEEE Transactions on Robotics*, vol. 40, pp. 4544–4560, 2024. [Online]. Available: https://doi.org/10.1109/tro.2024.3463504.

[18] Z. W. Pylyshyn and R. W. Storm, "Tracking multiple independent targets: Evidence for a parallel tracking mechanism.," *Spatial vision.*, vol. 3, no. 3, 1988.

[19] Z. Pylyshyn, "The role of location indexes in spatial perception: A sketch of the finst spatial-index model," *Cognition*, vol. 32, no. 1, pp. 65–97, 1989. [Online]. Available: https://doi.org/10.1016/0010-0277(89)90014-0.

[20] Z. Pylyshyn, "Some primitive mechanisms of spatial attention," *Cognition*, vol. 50, no. 1, pp. 363–384, 1994. [Online]. Available: https://doi.org/10.1016/0010-0277(94)90036-1.

[21] Z. Pylyshyn, "Mental pictures on the brain," *Nature*, vol. 372, no. 6503, pp. 289–290, Nov. 1994. [Online]. Available: https://doi.org/10.1038/372289a0.

[22] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960. [Online]. Available: https://doi.org/10.1115/1.3662552.

[23] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002. [Online]. Available: https://doi.org/10.1109/78.978374.

[24] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3047–3055. [Online]. Available: https://doi.org/10.1109/ICCV.2015.349.

[25] K.-C. Chang, C.-Y. Chong, and Y. Bar-Shalom, "Joint probabilistic data association in distributed sensor networks," *IEEE Transactions on Automatic Control*, vol. 31, no. 10, pp. 889–897, 1986. [Online]. Available: https://doi.org/10.1109/TAC.1986.1104143.

[26] S. Wu, X. Zhao, H. Zhou, and J. Lu, "Multi object tracking based on detection with deep learning and hierarchical clustering," in *2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, 2019, pp. 367–370. [Online]. Available: https://doi.org/10.1109/ICIVC47709.2019.8981026.

[27] B. Shuai, A. Berneshawi, X. Li, D. Modolo, and J. Tighe, "Siammot: Siamese multi-object tracking," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 367–12 377. [Online]. Available: https://doi.org/10.1109/CVPR46437.2021.01219.

[28] J. Peng, F. Qiu, J. See, *et al.*, "Tracklet siamese network with constrained clustering for multiple object tracking," in *2018 IEEE Visual Communications and Image Processing (VCIP)*, 2018, pp. 1–4. [Online]. Available: https://doi.org/10.1109/VCIP.2018.8698623.

[29] Y. Wang, Z. Mao, X. Wang, J. Ren, C. Meng, and J. Shen, "Deep adaptive discriminate siamese network with multi-level response for visual object tracking," in *2023 3rd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT)*, 2023, pp. 197–203. [Online]. Available: https://doi.org/10.1109/ICFEICT59519.2023.00042.

[30] M. C. Dang and D. D. Nguyen, "Attention mechanics for improving online multi-object tracking," in *2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, 2022, pp. 200–205. [Online]. Available: https://doi.org/10.1109/CACML55074.2022.00040.

[31] N. Senel, K. Kefferpütz, K. Doycheva, and G. Elger, "Multi-sensor data fusion for real-time multi-object tracking," *Processes*, vol. 11, no. 2, 2023. [Online]. Available: https://doi.org/10.3390/pr11020501.

[32] M. Chiani, A. Giorgetti, and E. Paolini, "Sensor radar for object tracking," *Proceedings of the IEEE*, vol. 106, no. 6, pp. 1022–1041, 2018. [Online]. Available: https://doi.org/10.1109/JPROC.2018.2819697.

[33] J. W. Choi, S. S. Nam, and S. H. Cho, "Multi-human detection algorithm based on an impulse radio ultra-wideband radar system," *IEEE Access*, vol. 4, pp. 10 300–10 309, 2016.

[34] A. Soumya, C. Krishna Mohan, and L. R. Cenkeramaddi, "Recent advances in mmWave-radar-based sensing, its applications, and machine learning techniques: A review," *Sensors*, vol. 23, no. 21, 2023. [Online]. Available: https://doi.org/10.3390/s23218901.

[35] Zeng Jiankui and Dong Ziming, "Some MIMO radar advantages over phased array radar," in *2010 The 2nd International Conference on Industrial Mechatronics and Automation*, vol. 2, 2010, pp. 211–213. [Online]. Available: https://doi.org/10.1109/ICINDMA.2010.5538331.

[36] E. Fishler, A. Haimovich, R. S. Blum, L. J. Cimini, D. Chizhik, and R. A. Valenzuela, "Spatial diversity in radars—models and detection performance," *IEEE Transactions on Signal Processing*, vol. 54, no. 3, pp. 823–838, 2006. [Online]. Available: https://doi.org/10.1109/TSP.2005.862813.

[37] I. Bekkerman and J. Tabrikian, "Target detection and localization using MIMO radars and sonars," *IEEE Transactions on Signal Processing*, vol. 54, no. 10, pp. 3873–3883, 2006. [Online]. Available: https://doi.org/10.1109/TSP.2006.879267.

[38] Y. Jang, H. Lim, and D. Yoon, "Multipath effect on radar detection of nonfluctuating targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 792–795, 2015. [Online]. Available: https://doi.org/10.1109/TAES.2014.130653.

[39] 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*, ser. KDD'96, Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[40] S. Lim, S. Lee, and S.-C. Kim, "Clustering of detected targets using DBSCAN in automotive radar systems," in *2018 19th International Radar Symposium*

(IRS), 2018, pp. 1–7. [Online]. Available: https://doi.org/10.23919/IRS.2018.8448228.

[41] D. Kellner, J. Klappstein, and K. Dietmayer, "Grid-based DBSCAN for clustering extended objects in radar data," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 365–370. [Online]. Available: https://doi.org/10.1109/IVS.2012.6232167.

[42] T. Wagner, R. Feger, and A. Stelzer, "Modification of DBSCAN and application to range/Doppler/DoA measurements for pedestrian recognition with an automotive radar system," Sep. 2015, pp. 269–272. [Online]. Available: https://doi.org/10.1109/EuRAD.2015.7346289.

[43] E. Schubert, F. Meinl, M. Kunert, and W. Menzel, "Clustering of high resolution automotive radar detections and subsequent feature extraction for classification of road users," Jun. 2015. [Online]. Available: https://doi.org/10.1109/IRS.2015.7226315.

[44] J. Schlichenmaier, F. Roos, M. Kunert, and C. Waldschmidt, "Adaptive clustering for contour estimation of vehicles for high-resolution radar," in *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 2016, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ICMIM.2016.7533930.

[45] S. Xu, H.-S. Shin, and A. Tsourdos, "Distributed multi-target tracking with D-DBSCAN clustering," in *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, 2019, pp. 148–155. [Online]. Available: https://doi.org/10.1109/REDUAS47371.2019.8999712.

[46] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI*, I. Kadar, Ed., International Society for Optics and Photonics, vol. 3068, SPIE, 1997, pp. 182–193. [Online]. Available: https://doi.org/10.1117/12.280797.

[47] M. Z. Ikram and M. Ali, "3-d object tracking in millimeter-wave radar for advanced driver assistance systems," in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 723–726. [Online]. Available: https://doi.org/10.1109/GlobalSIP.2013.6736993.

[48] İ. Sisman, A. O. Canbaz, and K. Yegin, "Micro-Doppler radar for human breathing and heartbeat detection," in *2015 Computational Electromagnetics International Workshop (CEM)*, 2015, pp. 1–2. [Online]. Available: https://doi.org/10.1109/CEM.2015.7237422.

[49] Y. Huang, Y. Wang, K. Shi, *et al.*, "HDNet: Hierarchical dynamic network for gait recognition using millimeter-wave radar," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICASSP49357.2023.10096835.

[50] P. A. Scharf, M. A. Mutschler, J. Iberle, H. Mantz, T. Walter, and C. Waldschmidt, "Spectroscopic estimation of surface roughness depth for mm-wave radar sensors," in *2019 16th European Radar Conference (EuRAD)*, 2019, pp. 93–96.

[51] A. L. Narayanan, A. B. K. T., H. Wu, J. Ma, and W. M. Huang, "mm-Wave radar hand shape classification using deformable transformers," in *2022 19th European Radar Conference (EuRAD)*, 2022, pp. 37–40. [Online]. Available: https://doi.org/10.23919/EuRAD54643.2022.9924850.

[52] W. Li, D. Li, J. Jiang, and Y. Gao, "3D contour imaging based on a millimeter wave MIMO radar," in *2022 7th International Conference on Communication, Image and Signal Processing (CCISP)*, 2022, pp. 297–301. [Online]. Available: https://doi.org/10.1109/CCISP55629.2022.9974203.

[53] R. Zhang and S. Cao, "Extending reliability of mmWave radar tracking and detection via fusion with camera," *IEEE Access*, vol. 7, pp. 137 065–137 079, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2942382.

[54] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person localization via RF body reflections," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, Oakland, CA: USENIX Association, May 2015, pp. 279–292. [Online]. Available: https://www.usenix.org/conference/nsdi15/technical-sessions/presentation/adib.

[55] O. Barnich and M. Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *Image Processing, IEEE Transactions*

on, vol. 20, pp. 1709–1724, Jul. 2011. [Online]. Available: https://doi.org/10.1109/TIP.2010.2101613.

[56] K. A. Smith, C. Csech, D. Murdoch, and G. Shaker, "Gesture recognition using mm-wave sensor for human-car interface," *IEEE Sensors Letters*, vol. 2, no. 2, pp. 1–4, 2018. [Online]. Available: https://doi.org/10.1109/LSENS.2018.2810093.

[57] Z. Zhou, Z. Cao, and Y. Pi, "Dynamic gesture recognition with a terahertz radar based on range profile sequences and Doppler signatures," *Sensors (Basel, Switzerland)*, vol. 18, Dec. 2017. [Online]. Available: https://doi.org/10.3390/s18010010.

[58] Y. Kim and H. Ling, "Human activity classification based on micro-Doppler signatures using a support vector machine," *IEEE T. Geoscience and Remote Sensing*, vol. 47, pp. 1328–1337, May 2009. [Online]. Available: https://doi.org/10.1109/TGRS.2009.2012849.

[59] X. Li, Y. He, and X. Jing, "A survey of deep learning-based human activity recognition in radar," *Remote Sensing*, vol. 11, p. 1068, May 2019. [Online]. Available: https://doi.org/10.3390/rs11091068.

[60] L. Senigagliesi, G. Ciattaglia, A. De santis, and E. Gambi, "People walking classification using automotive radar," *Electronics*, vol. 9, p. 588, Mar. 2020. [Online]. Available: https://doi.org/10.3390/electronics9040588.

[61] A. Singh, S. Sandha, L. Garcia, and M. Srivastava, "Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar," Oct. 2019, pp. 51–56. [Online]. Available: https://doi.org/10.1145/3349624.3356768.

[62] P. Zhao, C. X. Lu, B. Wang, *et al.*, "Heart rate sensing with a robot mounted mmWave radar," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2812–2818. [Online]. Available: https://doi.org/10.1109/ICRA40945.2020.9197437.

[63] L. Anitori, A. de Jong, and F. Nennie, "FMCW radar for life-sign detection," in *2009 IEEE Radar Conference*, 2009, pp. 1–6. [Online]. Available: https://doi.org/10.1109/RADAR.2009.4976934.

[64] D. J. Ewing, J. M. Neilson, and P. Travis, "New method for assessing cardiac parasympathetic activity using 24 hour electrocardiograms.," *Heart*, vol. 52, no. 4, pp. 396–402, 1984. [Online]. Available: https://doi.org/10.1136/hrt.52.4.396.

[65] M. Sekine and K. Maeno, "Non-contact heart rate detection using periodic variation in Doppler frequency," in *2011 IEEE Sensors Applications Symposium*, 2011, pp. 318–322. [Online]. Available: https://doi.org/10.1109/SAS.2011.5739803.

[66] X. Yang, J. Liu, Y. Chen, X. Guo, and Y. Xie, "Mu-id: Multi-user identification through gaits using millimeter wave radios," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 2589–2598. [Online]. Available: https://doi.org/10.1109/INFOCOM41043.2020.9155471.

[67] Z. Li, B. Chen, Z. Yang, *et al.*, "Ferrotag: A paper-based mmWave-scannable tagging infrastructure," in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, ser. SenSys '19, New York, New York: Association for Computing Machinery, 2019, pp. 324–337. [Online]. Available: https://doi.org/10.1145/3356250.3360019.

[68] P. Zhao, C. X. Lu, J. Wang, *et al.*, "Mid: Tracking and identifying people with millimeter wave radar," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019, pp. 33–40. [Online]. Available: https://doi.org/10.1109/DCOSS.2019.00028.

[69] C. Yang and G. Gidófalvi, "Detecting regional dominant movement patterns in trajectory data with a convolutional neural network," *International Journal of Geographical Information Science*, vol. 34, no. 5, pp. 996–1021, 2020. [Online]. Available: https://doi.org/10.1080/13658816.2019.1700510.

[70] Z. Zhang, X. Zhao, Y. Zhang, J. Zhang, H. Nie, and Y. Lou, "Efficient mining of hotspot regional patterns with multi-semantic trajectories," *Big Data Research*, vol. 22, p. 100 157, 2020. [Online]. Available: https://doi.org/10.1016/j.bdr.2020.100157.

[71] C. Du, C. Lin, R. Jin, B. Chai, Y. Yao, and S. Su, "Exploring the state-of-the-art in multi-object tracking: A comprehensive survey, evaluation, challenges, and future directions," *Multimedia Tools and Applications*, vol. 83, no. 29,

pp. 73 151–73 189, Sep. 2024. [Online]. Available: https://doi.org/10.1007/s11042-023-17983-2.

[72] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [Online]. Available: https://doi.org/10.1109/5.726791.

[73] G. Cai, X. Wang, J. Shi, X. Lan, T. Su, and Y. Guo, “Vehicle detection based on information fusion of mmWave radar and monocular vision,” *Electronics*, vol. 12, no. 13, 2023. [Online]. Available: https://doi.org/10.3390/electronics12132840.

[74] Y. Zhou, Y. Dong, F. Hou, and J. Wu, “Review on millimeter-wave radar and camera fusion technology,” *Sustainability*, vol. 14, no. 9, p. 5114, Apr. 2022. [Online]. Available: https://doi.org/10.3390/su14095114.

[75] G. Alessandretti, A. Broggi, and P. Cerri, “Vehicle and guard rail detection using radar and vision data fusion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 95–105, 2007. [Online]. Available: https://doi.org/10.1109/TITS.2006.888597.

[76] L. Zhao, “Multi-sensor information fusion technology and its applications,” *Infrared*, vol. 42, no. 1, p. 21, 2021. [Online]. Available: http://doi.org/10.3969/j.issn.1672-8785.2021.01.005.

[77] C. Cao, J. Gao, and Y. C. Liu, “Research on space fusion method of millimeter wave radar and vision sensor,” *Procedia Computer Science*, vol. 166, pp. 68–72, 2020, Proceedings of the 3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2019). [Online]. Available: https://doi.org/10.1016/j.procs.2020.02.015.

[78] C. Song, S. Gukjin, H. Kim, D. Gu, J.-H. Lee, and Y. Kim, “A novel method of spatial calibration for camera and 2D radar based on registration,” Jul. 2017, pp. 1055–1056. [Online]. Available: https://doi.org/10.1109/IIAI-AAI.2017.62.

[79] H. Junlong, “Unified calibration method for millimeter-wave radar and machine vision,” *International Journal of Engineering Research and*, vol. V7, Nov. 2018. [Online]. Available: https://doi.org/10.17577/IJERTV7IS100086.

[80] X. Liu and Z. Cai, “Advanced obstacles detection and tracking by fusing millimeter wave radar and image sensor data,” in *ICCAS 2010*, 2010, pp. 1115–1120. [Online]. Available: https://doi.org/10.1109/ICCAS.2010.5669740.

[81] T. Wang, J. Xin, and N. Zheng, “A method integrating human visual attention and consciousness of radar and vision fusion for autonomous vehicle navigation,” in *2011 IEEE Fourth International Conference on Space Mission Challenges for Information Technology*, 2011, pp. 192–197. [Online]. Available: https://doi.org/10.1109/SMC-IT.2011.15.

[82] Z. Ji and D. Prokhorov, “Radar-vision fusion for object classification,” in *2008 11th International Conference on Information Fusion*, IEEE, 2008, pp. 1–7.

[83] X. Zhang, M. Zhou, P. Qiu, Y. Huang, and J. Li, “Radar and vision fusion for the real-time obstacle detection and identification,” *Industrial Robot: the international journal of robotics research and application*, vol. 46, pp. 391–395, May 2019. [Online]. Available: https://doi.org/10.1108/IR-06-2018-0113.

[84] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587. [Online]. Available: https://doi.org/10.1109/CVPR.2014.81.

[85] T.-Y. Lim, S. A. Markowitz, and M. N. Do, “RaDICaL: A synchronized FMCW radar, depth, IMU and RGB camera data dataset with low-level FMCW radar signals,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 941–953, 2021. [Online]. Available: https://doi.org/10.1109/JSTSP.2021.3061270.

[86] Z. Wei, F. Zhang, S. Chang, Y. Liu, H. Wu, and Z. Feng, “mmWave radar and vision fusion for object detection in autonomous driving: A review,” *Sensors*, vol. 22, no. 7, 2022. [Online]. Available: https://doi.org/10.3390/s22072542.

[87] T. Kato, Y. Ninomiya, and I. Masaki, “An obstacle detection method by fusion of radar and motion stereo,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 182–188, 2002. [Online]. Available: https://doi.org/10.1109/TITS.2002.802932.

[88] X.-p. Guo, J.-s. Du, J. Gao, and W. Wang, "Pedestrian detection based on fusion of millimeter wave radar and vision," in *Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition*, ser. AIPR '18, Beijing, China: Association for Computing Machinery, 2018, pp. 38–42. [Online]. Available: https://doi.org/10.1145/3268866.3268868.

[89] R. Streubel and B. Yang, "Fusion of stereo camera and MIMO-FMCW radar for pedestrian tracking in indoor environments," in *2016 19th International Conference on Information Fusion (FUSION)*, 2016, pp. 565–572.

[90] W. Huang, Z. Zhang, W. Li, and J. Tian, "Moving object tracking based on millimeter-wave radar and vision sensor," *Journal of Applied Science and Engineering*, vol. 21, pp. 609–614, Jan. 2018. [Online]. Available: https://doi.org/10.6180/jase.201812_21(4).0014.

[91] E. Richter, R. Schubert, and G. Wanielik, "Radar and vision based data fusion - advanced filtering techniques for a multi object vehicle tracking system," in *2008 IEEE Intelligent Vehicles Symposium*, 2008, pp. 120–125. [Online]. Available: https://doi.org/10.1109/IVS.2008.4621245.

[92] T.-Y. Lim, A. Ansari, B. Major, *et al.*, "Radar and camera early fusion for vehicle detection in advanced driver assistance systems," in *Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems*, vol. 2, 2019, p. 7.

[93] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp, "A deep learning-based radar and camera sensor fusion architecture for object detection," in *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2019, pp. 1–7. [Online]. Available: https://doi.org/10.1109/SDF.2019.8916629.

[94] T. Winterling, J. Lombacher, M. Hahn, J. Dickmann, and C. Wöhler, "Optimizing labelling on radar-based grid maps using active learning," in *2017 18th International Radar Symposium (IRS)*, 2017, pp. 1–6. [Online]. Available: https://doi.org/10.23919/IRS.2017.8008123.

[95] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pat-*

tern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2016.2577031.

[96] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448. [Online]. Available: https://doi.org/10.1109/ICCV.2015.169.

[97] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2353–2362. [Online]. Available: https://doi.org/10.1109/ICCV.2017.256.

[98] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, *Rmpe: Regional multi-person pose estimation*, 2018. [Online]. Available: https://arxiv.org/abs/1612.00137.

[99] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 103–111, 2019. [Online]. Available: https://doi.org/10.1109/MVT.2019.2892497.

[100] H. Wallace, "Millimeter-wave propagation measurements at the ballistic research laboratory," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 26, no. 3, pp. 253–258, 1988. [Online]. Available: https://doi.org/10.1109/36.3028.

[101] G. P. Kulemin, "Influence of propagation effects on a millimeter-wave radar operation," in *Radar Sensor Technology IV*, R. Trebits and J. L. Kurtz, Eds., International Society for Optics and Photonics, vol. 3704, SPIE, 1999, pp. 170–178. [Online]. Available: https://doi.org/10.1117/12.354594.

[102] B. Jiu, H. Liu, D. Feng, and Z. Liu, "Minimax robust transmission waveform and receiving filter design for extended target detection with imprecise prior knowledge," *Signal Processing*, vol. 92, no. 1, pp. 210–218, 2012. [Online]. Available: https://doi.org/10.1016/j.sigpro.2011.07.008.

[103] S. M. Karbasi, A. Aubry, A. De Maio, and M. H. Bastani, "Robust transmit code and receive filter design for extended targets in clutter," *IEEE Transactions on Signal Processing*, vol. 63, no. 8, pp. 1965–1976, 2015. [Online]. Available: https://doi.org/10.1109/TSP.2015.2404301.

[104] B. Tang and J. Tang, "Robust waveform design of wideband cognitive radar for extended target detection," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 3096–3100. [Online]. Available: https://doi.org/10.1109/ICASSP.2016.7472247.

[105] J. Pegoraro and M. Rossi, "Human tracking with mmWave radars: A deep learning approach with uncertainty estimation," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022, pp. 1–5. [Online]. Available: https://doi.org/10.1109/SPAWC51304.2022.9833987.

[106] N. Sireesha, K. Chithra, and T. Sudhakar, "Adaptive filtering based on least mean square algorithm," in *2013 Ocean Electronics (SYMPOL)*, 2013, pp. 42–48. [Online]. Available: https://doi.org/10.1109/SYMPOL.2013.6701910.

[107] S. Lee, J.-Y. Lee, and S.-C. Kim, "Mutual interference suppression using wavelet denoising in automotive FMCW radar systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 887–897, 2021. [Online]. Available: https://doi.org/10.1109/TITS.2019.2961235.

[108] Y. Sun, Z. Huang, H. Zhang, Z. Cao, and D. Xu, *3DRIMR: 3D reconstruction and imaging via mmWave radar based on deep learning*, 2021. [Online]. Available: https://arxiv.org/abs/2108.02858.

[109] Q. Fang, Y. Yan, and G. Ma, *Gesture recognition in millimeter-wave radar based on spatio-temporal feature sequences*, 2023. [Online]. Available: https://arxiv.org/abs/2309.09528.

[110] R. Ahmed, N. Maheshwari, and P. Lalla, "Wavelet based iterative thresholding for denoising of remotely sensed optical and synthetic aperture radar images," in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, 2014, pp. 1331–1335. [Online]. Available: https://doi.org/10.1109/ICACCCT.2014.7019316.

[111] G. Chandraiah and T. Sreenivasulu Reddy, "Denoising of mst radar signal using multi-band wavelet transform with improved thresholding," in *2018 Second International Conference on Inventive Communication and Compu-*

tational Technologies (ICICCT), 2018, pp. 1026–1030. [Online]. Available: https://doi.org/10.1109/ICICCT.2018.8473060.

[112] S. Shaham, M. Kokshoorn, M. Ding, Z. Lin, and M. Shirvanimoghaddam, "Extended Kalman filter beam tracking for millimeter wave vehicular communications," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICCWorkshops49005.2020.9145366.

[113] V. T. Dang, "An adaptive Kalman filter for radar tracking application," in *2008 Microwaves, Radar and Remote Sensing Symposium*, 2008, pp. 261–264. [Online]. Available: https://doi.org/10.1109/MRRS.2008.4669591.

[114] F. J. Abdu, Y. Zhang, M. Fu, Y. Li, and Z. Deng, "Application of deep learning on millimeter-wave radar signals: A review," *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: https://doi.org/10.3390/s21061951.

[115] A. Bournas and E. Baltas, "Determination of the Z-R relationship through spatial analysis of X-Band weather radar and rain gauge data," *Hydrology*, vol. 9, no. 8, 2022. [Online]. Available: https://doi.org/10.3390/hydrology9080137.

[116] O. P. Prat and A. P. Barros, "Exploring the transient behavior of Z–R relationships: Implications for radar rainfall estimation," *Journal of Applied Meteorology and Climatology*, vol. 48, no. 10, pp. 2127–2143, 2009. [Online]. Available: https://doi.org/10.1175/2009JAMC2165.1.

[117] F. Rajabi, N. Faraji, and M. Hashemi, "An efficient video-based rainfall intensity estimation employing different recurrent neural network models," *Earth Science Informatics*, vol. 17, no. 3, pp. 2367–2380, Jun. 2024. [Online]. Available: https://doi.org/10.1007/s12145-024-01290-x.

[118] I. Ebtehaj and H. Bonakdari, "CNN vs. LSTM: A comparative study of hourly precipitation intensity prediction as a key factor in flood forecasting frameworks," *Atmosphere*, vol. 15, no. 9, 2024. [Online]. Available: https://doi.org/10.3390/atmos15091082.

[119] S. Kim, S. Hong, M. Joh, and S. Song, "Deeprain: ConvLSTM network for precipitation prediction using multichannel radar data," *CoRR*, vol. abs/1711.02316, 2017. [Online]. Available: http://arxiv.org/abs/1711.02316.

[120] L. Mou, L. Bruzzone, and X. X. Zhu, “Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 924–935, Feb. 2019. [Online]. Available: https://doi.org/10.1109/tgrs.2018.2863224.

[121] C. Shi, Z. Zhang, W. Zhang, C. Zhang, and Q. Xu, “Learning multiscale temporal–spatial–spectral features via a multipath convolutional LSTM neural network for change detection with hyperspectral images,” *IEEE Transactions on Geoscience and Remote Sensing*, May 2022. [Online]. Available: https://doi.org/10.1109/TGRS.2022.3176642.

[122] J. Li, L. Xueyi, and D. He, “A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction,” *IEEE Access*, vol. PP, pp. 1–1, May 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2919566.

[123] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.

[124] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, Dec. 2014.

[125] M. Masum, H. Shahriar, H. Haddad, *et al.*, “Bayesian hyperparameter optimization for deep neural network-based network intrusion detection,” in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 5413–5419. [Online]. Available: https://doi.org/10.1109/BigData52589.2021.9671576.