Learning Based Tracking and Recognition with mmWave Radar for Smart Environments

Peijun Zhao

Hertford College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2021

Abstract

Ubiquitous sensing is a key enabler for smart environment applications, and mmWave radar is an ideal candidate due to its high usability, low cost and privacy preserving features. Tracking and recognition are two fundamental functions of mmWave radar sensing, but with conventional methods, their performance is often affected by many factors such as noise, limited resolution, and multi-path effects. In this thesis, we explore the use of novel deep learning techniques in radar signal processing chain for more accurate and robust tracking and recognition, to better support smart environment applications.

First we explore the problem of using radar point cloud for human tracking and identification. In the proposed system named 'mID', we first associate clusters between frames to track human targets, followed by using deep neural networks to take a sequence of voxelized point clusters as input for human identification, as well as using the latent features for intruder detection. In the following chapter we propose 'mBeats', a system to monitor user's heart rate using deep learning with a robot mounted mmWave radar. 'mBeats' shows superior performance compared to conventional methods for heart rate prediction, and can also estimate the prediction uncertainty, which is very important for health-related applications. Next, we introduce using a deep learning based method for 3D tracking of an unmodified drone. The proposed technique can achieve a tracking accuracy 3 times better than conventional methods, with improved robustness and computational efficiency. In the last work, we use an end-to-end neural network, replacing the conventional Discrete Fourier Transform (DFT) based pre-processing with 'CubeLearn', which is our proposed learnable pre-processing module, for human gesture and activity recognition. 'CubeLearn' can adaptively extract the most informative features from raw data which best fit the target task, and shows consistently better performance than the conventional DFT pre-processing module with different models and on different tasks.

Through the research introduced in this thesis, we demonstrate that applying learning based techniques for mmWave radar data processing can significantly enhance tracking and recognition performance, and benefit downstream smart environment applications.

Learning Based Tracking and Recognition with mmWave Radar for Smart Environments



Peijun Zhao

Hertford College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2021

Acknowledgements

First of all I would like to express my most sincere gratitude to my supervisors Prof. Andrew Markham and Prof. Niki Trigoni. They not only gave me the chance to have the wonderful four years at Oxford, but also guided me through the journey of exploring the unknowns. I am deeply impressed by their broad and cutting-edge visions, their ways of thinking, and their brilliant personalities. They inspired me countless times in my research, shaped me into an independent researcher and I really learned a lot from them. I feel extremely lucky to be their student.

And my special appreciation to Dr. Xiaoxuan Lu. Xiaoxuan mentored me, taught me research skills through our collaborations, and helped me build connections with people from academia and industry. Most importantly, he impressed me with his tremendous research enthusiasm and set me the perfect example of what an exceptional researcher should be like.

Thanks to all my collaborators from Oxford CPS group, including Dr. Changhao Chen, Bing Wang, Wei Wang, Jianan Wang, Dr. Linhai Xie, Dr. Pedro Porto Buarque de Gusmão, Dr. Stefano Rosa, Dr. Muhamad Risqi U. Saputra and Yasin Almalioglu. Their in-depth understanding of different topics and creative ideas always brought me new insights in a wide range of research fields.

Also thanks to my friends and colleagues in CPS group, including Dr. Bo Yang, Qingyong Hu, Zihang Lai, Dr. Zhihua Wang, Shuyu Lin, Jialu Wang, Kaichen Zhou, Yuhang He, Ada Alevizaki, Dr. Johan Wahlstrom and Dr. Prince Abudu. They made my life at Oxford colorful and I will always treasure the time we spent together.

Besides, I would like to express my thankfulness to my external examiner, Prof. Lu Su, and my internal examiner, Prof. Tam Vu, for their constructive feedback in the thesis examination, as well as Prof. Alex Rogers and Prof. Max Van Kleek, for their valuable comments on my research in the transfer and confirmation examinations. Their suggestions really helped me to improve my thesis a lot. Also, thanks to Prof. Jonathan Barrett and MPLS Graduate Studies for overseeing and guiding me through all the process.

Thanks to the staff at the Department of Computer Science and Hertford College, your warm welcome and accommodation brought me great sense of belonging.

And finally my deepest gratefulness to my parents who backed me up financially and emotionally throughout the years. I couldn't have gone through the difficult times without their unconditional support.

Abstract

Ubiquitous sensing is a key enabler for smart environment applications, and mmWave radar is an ideal candidate due to its high usability, low cost and privacy preserving features. Tracking and recognition are two fundamental functions of mmWave radar sensing, but with conventional methods, their performance is often affected by many factors such as noise, limited resolution, and multi-path effects. In this thesis, we explore the use of novel deep learning techniques in radar signal processing chain for more accurate and robust tracking and recognition, to better support smart environment applications.

First we explore the problem of using radar point cloud for human tracking and identification. In the proposed system named 'mID', we first associate clusters between frames to track human targets, followed by using deep neural networks to take a sequence of voxelized point clusters as input for human identification, as well as using the latent features for intruder detection. In the following chapter we propose 'mBeats', a system to monitor user's heart rate using deep learning with a robot mounted mmWave radar. 'mBeats' shows superior performance compared to conventional methods for heart rate prediction, and can also estimate the prediction uncertainty, which is very important for health-related applications. Next, we introduce using a deep learning based method for 3D tracking of an unmodified drone. The proposed technique can achieve a tracking accuracy 3 times better than conventional methods, with improved robustness and computational efficiency. In the last work, we use an end-to-end neural network, replacing the conventional Discrete Fourier Transform (DFT) based pre-processing with 'CubeLearn', which is our proposed learnable pre-processing module, for human gesture and activity recognition. 'CubeLearn' can adaptively extract the most informative features from raw data which best fit the target task, and shows consistently better performance than the conventional DFT pre-processing module with different models and on different tasks.

Through the research introduced in this thesis, we demonstrate that applying learning based techniques for mmWave radar data processing can significantly enhance tracking and recognition performance, and benefit downstream smart environment applications.

Contents

List of Figures viii

List of Abbreviations xiii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Challenges and Questions . . . 8
  - 1.2.1 Challenge 1: Point Cloud Target Recognition . . . 8
  - 1.2.2 Challenge 2: Micro Displacement Pattern Tracking . . . 9
  - 1.2.3 Challenge 3: Accurate Macro Movement Tracking . . . 9
  - 1.2.4 Challenge 4: Feature Extraction from Raw Data . . . 10
- 1.3 Contributions . . . 11
- 1.4 Thesis Outline . . . 15

2 Background 16
- 2.1 Principles of mmWave FMCW Radar . . . 17
  - 2.1.1 mmWave FMCW Radar signal model . . . 17
  - 2.1.2 Maximum Detection Range . . . 19
  - 2.1.3 Range Estimation . . . 20
  - 2.1.4 Micro-movement and Velocity Estimation . . . 23
  - 2.1.5 Angle of Arrival Estimation . . . 24
  - 2.1.6 Point Cloud Generation Pipeline . . . 26
- 2.2 Neural Networks . . . 28
  - 2.2.1 Fully Connected Layers . . . 28
  - 2.2.2 Convolutional Layers . . . 29
  - 2.2.3 Recurrent Layers . . . 30
  - 2.2.4 Classification and Regression . . . 32
  - 2.2.5 Complex-weight Neural Networks . . . 33
- 2.3 mmWave radar sensing methods for smart environments . . . 34
  - 2.3.1 Target Detection and Tracking . . . 34
  - 2.3.2 Micro Displacement Sensing . . . 35
  - 2.3.3 Human Pose Estimation and Identification . . . 37

2.3.4 Gesture and Activity Recognition . . . 38
2.3.5 Summary of Recent Breakthroughs in mmWave FMCW Radar Sensing . . . 39
2.3.6 Discussion . . . 40
2.4 Other Smart Environment Sensing Solutions . . . 42
2.4.1 Camera based Solutions . . . 43
2.4.2 Acoustic based Solutions . . . 43
2.4.3 RF based Solutions . . . 45

3 People Tracking and Identification with mmWave Radar Point Cloud 47
3.1 Introduction . . . 49
3.2 System Requirements . . . 51
3.3 System Design . . . 52
3.3.1 mmWave Radar Point Cloud Generation . . . 53
3.3.2 Point Cloud Clustering . . . 54
3.3.3 Moving Object Tracking . . . 55
3.3.4 User Identification . . . 58
3.3.5 Intruder Detection . . . 59
3.4 Implementation . . . 61
3.4.1 Testbed Setup . . . 61
3.4.2 Dataset Collection . . . 62
3.4.3 mID Configuration . . . 63
3.5 Evaluation . . . 65
3.5.1 Sensitivity Analysis . . . 65
3.5.2 User Tracking . . . 66
3.5.3 Identification Evaluation . . . 67
3.5.4 Intruder Detection Evaluation . . . 72
3.6 Discussion on Scalability and Robustness . . . 73
3.7 Related Work . . . 75
3.7.1 Device-Free Gait Recognition . . . 75
3.7.2 Other RF-based Human-centric Applications . . . 76
3.8 Chapter Summary . . . 76

4 Sensing Micro Displacements 78
4.1 Introduction . . . 79
4.2 Related Work . . . 82
4.2.1 Electrocardiogram-based Techniques . . . 82
4.2.2 Vision-based Techniques . . . 82
4.2.3 RF-based Techniques . . . 83

4.3 System Requirements . . . 84
4.4 Proposed System . . . 85
4.4.1 User Tracking Module . . . 85
4.4.2 mmWave Servoing Module . . . 86
4.4.3 Heart Rate Estimation Module . . . 88
4.5 Implementation . . . 91
4.5.1 Data Collection . . . 91
4.5.2 Network Training . . . 92
4.6 Evaluation . . . 93
4.6.1 Setup . . . 93
4.6.2 Performance Comparison . . . 94
4.6.3 Uncertainty Evaluation . . . 96
4.6.4 Impact of the mmWave Servoing Module . . . 98
4.7 Discussion . . . 98
4.8 Chapter Summary . . . 99

5 Target Tracking with Radar Heatmaps 101
5.1 Introduction . . . 102
5.2 Related Work . . . 105
5.2.1 Cooperative Localization . . . 105
5.2.2 Non-cooperative Localization . . . 106
5.3 Conventional Drone Tracking Pipeline . . . 107
5.3.1 FMCW Radar Background . . . 107
5.3.2 Tracking the Drone with Radar Point Cloud . . . 108
5.3.3 Tracking the Drone with Range-Angle Heatmaps . . . 109
5.3.4 Problems with Conventional Drone Tracking Pipeline . . . 110
5.4 System Requirements . . . 111
5.5 Proposed Method . . . 111
5.5.1 Data Preprocessing . . . 112
5.5.2 Network Structure . . . 112
5.6 Implementation . . . 113
5.6.1 Data Collection . . . 113
5.6.2 Algorithm Implementation . . . 115
5.7 Evaluation . . . 115
5.7.1 Localization Evaluation . . . 115
5.7.2 Running Time Comparison . . . 118
5.7.3 Impact of Distance on Tracking Error . . . 119
5.7.4 Impact of Training Data Size . . . 120
5.7.5 Ablation Study: Impact of Network Structure . . . 121
5.8 Discussion . . . 122
5.9 Chapter Summary . . . 123

6 End-to-end Learning with Learnable Pre-processing 124
6.1 Introduction . . . 126
6.2 Related Work . . . 129
6.2.1 mmWave Radar for Human-centered Applications . . . 129
6.2.2 End-to-end mmWave Radar Gesture/Activity Recognition Methods . . . 131
6.2.3 Complex Neural Network on Radar . . . 131
6.3 Range, Doppler, and Angle-of-arrival estimation with DFT . . . 132
6.4 Gesture and activity recognition models . . . 134
6.5 Requirements . . . 135
6.6 Learnable pre-processing . . . 136
6.6.1 Problems with Conventional DFT pre-processing . . . 136
6.6.2 CubeLearn Module Architecture . . . 138
6.6.3 CubeLearn Module Weight Initialization . . . 140
6.6.4 Effect of CubeLearn Module . . . 141
6.7 Experiment Setup . . . 143
6.7.1 Experiment Design . . . 143
6.7.2 Data Collection Setup . . . 145
6.7.3 Neural Network Implementation . . . 146
6.8 Evaluation . . . 147
6.8.1 Hand Gesture Recognition Evaluation . . . 148
6.8.2 Generalization to AGR and HAR Task . . . 150
6.8.3 Convergence Analysis . . . 152
6.9 Ablation Study . . . 153
6.9.1 Impact of CubeLearn Module Learning Rate . . . 153
6.9.2 Impact of Weight Initialization Method . . . 155
6.9.3 Impact of Network Structure . . . 158
6.10 Complexity and Running Time Analysis . . . 163
6.11 Impact of Raw Data Size . . . 165
6.12 Discussion and Future Work . . . 166
6.12.1 Model Selection under Different Situations . . . 166
6.12.2 Future Work . . . 167
6.13 Chapter Summary . . . 168

7 Discussion 169
7.1 Applying Deep Learning on Different Radar Data Representations . 169
7.2 Future Directions . . . 173

8 Conclusion 175

References 177

List of Figures

1.1 An ideal sensor for smart environment should have high usability, low cost and privacy preserving features. 2
1.2 Single-chip mmWave FMCW radar. 5
1.3 Works in this thesis are aiming to build a smarter environment with mmWave sensing technologies. 6

2.1 Sawtooth modulation in frequency and time domain. 17
2.2 Basic mmWave FMCW Radar architecture. 18
2.3 A transmitted signal is reflected by three different targets at different distances in the scene. This results in different frequency components in the IF signal, which can be resolved by Range FFT. 21
2.4 Doppler FFT is applied on the peak phase vector across chirps. Targets at the same range with different radial velocities can be differentiated with Doppler FFT. 23
2.5 The slight difference in the distances from the target to the receivers in the receiver antenna array results in a phase shift, which can be used to estimate the angle of arrival. 24
2.6 TI IWR1443/IWR6843 MIMO layout. 26
2.7 Point Cloud generation pipeline. Peak detection algorithms like CFAR2D is applied on Range-Velocity Heatmap. For each detected peak, elevation and azimuth angle are estimated and converted to a point in the point cloud. 27
2.8 Fully Connected Layer 29
2.9 A Basic Recurrent Layer. 31
2.10 Long Short Term Memory Cell Structure. 31
2.11 Bi-directional Recurrent Layer Structure. 31
2.12 Typical mmWave FMCW Radar Vital Sign Monitoring. 36
2.13 DFT Pre-processing + Neural Network Classifier Hybrid Method for Gesture/Activity Recognition. 38

3.1 System Overview 52

3.2 2D Projections of a 3D Point Cloud (see Section 3.3.2). Red points corresponding to a human object are compact in the horizontal x-y plane but dispersed in the vertical z-axis. 54
3.3 mID tracking workflow. 57
3.4 Classification Network Structure. Here T represents the number of data frames used for identification. 58
3.5 Identification and Intruder Detection module workflow. The blue arrow represents the information flow of the human identification part, the green arrow represents the information flow of the center loss part and the yellow arrow represents the information flow of intruder detection part. The network is trained with the combination of cross-entropy loss and center loss. 60
3.6 Experiment Settings. Vicon system is a high-precision tracking system used for acquiring ground truth trajectories. 62
3.7 Clustering results with different α. A small α leads to loose clusters containing many noisy points. A large α splits a human object into two clusters. Setting $\alpha = 0.25$ gives the best empirical performance, as shown in the middle one. 66
3.8 Comparison of tracking performance between mmWave radar and Kinect v2. (a) Kinect v2 can only track people within $4.5m$. (b) Inferred trajectory. (GT: Ground Truth by Vicon) 67
3.9 Confusion Matrix of 12 Users. 68
3.10 Impact of number of people. mID works reliably with different numbers of participants and performs best when there are no more than 6 people in the dataset, e.g., in home scenarios. 69
3.11 Ablation study of neural network architecture and size. 69
3.12 Impact of Observation Time on identification performance. The total observation period is cut into testing samples and a majority voting method is adopted. The green color represents percentage of correct prediction, and the red color represents wrong prediction. Yellow represents Unable to Decide (U.D.), which happens when the top two predictions get the same number of votes. 71

4.1 mBeats is a robot mounted mmWave radar system. It is able to provide periodic heart rate measurements of a user, without interfering in a user's daily activities, and without being constrained to only operate at a certain location. 80

4.2 mBeats consists of three modules operating in a pipelined manner. (i) *user tracking module* actively tracks a target with the mmWave radar and instructs the mobile robot to move towards the user's proximity; (ii) *mmWave servoing* module adaptively rotates the mmWave radar that optimises the sensor angle for best heart rate observation; and (iii) *heart rate estimation* module senses the micro displacement of user's skin and estimates his/her heart rate with an uncertainty. . . 85
4.3 Impact of observation angles on heart rate estimation. We place the robot approximately 0.5m away from a user's lower leg and rotate the sensor from 75°to 105°at a stride of 5°and each angle for approximately 40 seconds. The error is obtained from a fixed estimation method [214]. PTA: Peak-to-average ratio. 86
4.4 Heart Rate Estimation. A bandpass filter is used to extract heartbeat waveforms (circled in orange) from phase variation signals. A convolutional neural network takes inputs as the extracted waveforms and predicts both heart rate value and uncertainty. 88
4.5 Experiment Platform. The mmWave sensor is installed on a Turtlebot 2, and an Arduino Uno serves to receive control signal from PC and rotates the Servo Motor. 91
4.6 The 8 different poses used in evaluation. 92
4.7 Overall performance comparison in different test categories. The 90% accuracy line is indicated for reference. 94
4.8 Uncertainty Estimation. Here we use the exponential of uncertainty estimation since the uncertainty is defined as the logarithm of the variance. 96

5.1 Compared to the optical tracking method and the UWB tracking method, our proposed system does not require any marker modification on the drone and only needs a low-cost mmWave radar. 104
5.2 Effects of the Clutter Removal algorithm. 107
5.3 Drone tracking with radar Point Cloud. 108
5.4 2D (top) and 3D (bottom) FFT and MUSIC pipelines. 109
5.5 Three factors that impact drone localization accuracy with conventional methods. 110
5.6 Proposed Method. For each chirp, the pre-processing module generates two range-azimuth heatmaps and four range-elevation heatmaps, which are sent into the Feature Extractor module respectively. The feature vectors of multiple chirps in a frame are input into a LSTM network and the output of the last timestamp is used to estimate the 3D location with fully-connected layers. 112

5.7 Experiment Settings . . . 113
5.8 Localization Error with Standard Deviation (cm). . . . 116
5.9 Cumulative Distribution Function (CDF) of the errors produced by different methods. . . . 116
5.10 Qualitative localization comparison on a sample testing sequence. Black dashed lines represent ground truth. . . . 117
5.11 Mean Error v.s. Distance. Errors of conventional baselines increase with distance, while the error of mDrone is relatively constant. . . . 119
5.12 Error v.s. Training Size . . . 120
5.13 Comparison between Different Network Structures. . . . 121

6.1 We propose learnable pre-processing module to replace conventional DFT and build end-to-end deep neural networks for mmWave FMCW radar interactive recognition tasks. . . . 127
6.2 Basic mmWave FMCW Radar architecture and conventional DFT signal processing. . . . 132
6.3 Two types of downstream classifier architectures we study in this work. 134
6.4 Three limitations of the conventional DFT pre-processing. In (a) and (b) Range dimension is used for illustration, and the problem also exists for Doppler and AoA dimensions. . . . 137
6.5 Architecture of the proposed CubeLearn module. . . . 139
6.6 t-SNE visualization of the conventional DFT pre-processed data and the learnable pre-processed data at different training epochs. Different colors represent different classes. Best viewed in color. . . . 142
6.7 Hand Gesture Set Design. . . . 143
6.8 Arm Gesture Set Design. . . . 143
6.9 Experiment Settings. . . . 144
6.10 Sensor Setup. . . . 145
6.11 Hand Gesture Recognition Accuracy. . . . 148
6.12 Arm Gesture Recognition Accuracy. . . . 151
6.13 Human Activity Recognition Accuracy. . . . 151
6.14 Visualization of the training processes. . . . 153
6.15 'In-set' test accuracy and 'out-of-set' test accuracy with different CubeLearn module learning rate. The learning rate of the neural network classifier is fixed at 0.0003. . . . 154
6.16 Loss and Accuracy curve. Left: Training Loss v.s. Validation Loss. Right: Training Accuracy v.s. Validation Accuracy. Red color represents training set, and green color represents validation set. The darker the color, the higher the learning rate. The learning rates are the same as in Fig. 6.15. . . . 154

6.17 Comparison of accuracies on 'in-set' and 'out-of-set' users with different complex layer initialization methods. 156
6.18 Comparison of accuracies on 'in-set' and 'out-of-set' users with different CubeLearn module structures. 161
6.19 Accuracy and inference time with different size of raw data input. . 165

7.1 Applying deep learning techniques on different radar representations. 170

List of Abbreviations

IoT . . . . . Internet of Things

HCI . . . . . Human-computer Interactions

mmWave . . . Millimeter wave

FMCW . . . . Frequency-modulated Continuous-wave

FFT . . . . . Fast Fourier Transform

DFT . . . . . Discrete Fourier Transform

LPF . . . . . Low-pass Filter

MIMO . . . . . Multiple-Input Multiple-Output

SIMO . . . . . Single-Input Multiple-Output

SISO . . . . . Single-Input Single-Output

AoA . . . . . Angle-of-arrival

DNN . . . . . Deep Neural Network

CNN . . . . . Convolutional Neural Network

RNN . . . . . Recurrent Neural Network

LSTM . . . . . Long Short-Term Memory

HGR . . . . . Hand Gesture Recognition

HAR . . . . . Human Activity Recognition

AGR . . . . . Arm Gesture Recognition

1 Introduction

Contents

1.1 Motivation 1
1.2 Research Challenges and Questions 8
1.2.1 Challenge 1: Point Cloud Target Recognition 8
1.2.2 Challenge 2: Micro Displacement Pattern Tracking . . . 9
1.2.3 Challenge 3: Accurate Macro Movement Tracking . . . 9
1.2.4 Challenge 4: Feature Extraction from Raw Data 10
1.3 Contributions 11
1.4 Thesis Outline 15

1.1 Motivation

Ever since the late 20th century, scientists have been imagining living in smart environments where computers 'disappear' into the background – they are invisible, but at the same time, everywhere [1]. With breakthroughs in high bandwidth communication technologies, the development of different sensing modalities, and the advances in intelligent algorithm design, there is little doubt that we are fast approaching the era of ubiquitous smart sensing [2].

An intelligent system in a smart environment should be able to interact with people, understand and predict a user's intentions, and provide personalized services, as well as sensing and interacting with other objects without human

intervention [3, 4]. Typical smart environment Human-computer Interaction (HCI) applications could include people tracking and identification, intruder detection, mid-air and touch-based gesture recognition, voice command recognition, health status monitoring, activity recognition and fall detection, as well as providing personalized services such as temperature control and ambient music playing. The smart environment also senses and interacts with other smart agents, such as floor-cleaning robots or delivery drones, to achieve better automation.

Cameras and microphones are currently the most widely adopted sensors for smart home applications. Different types of voice assistants have been implemented in the form of smart speakers, which typically serve as the center of a smart home intelligent system. Through interacting with voice assistants, people can retrieve information, issue commands like ordering a pizza or calling a taxi, as well as controlling other smart home appliances such as lights [5]. Cameras are also installed in smart homes for intruder detection and baby/elderly monitoring. However, despite the wide adoption of voice assistants and home cameras, people are increasingly concerned about the privacy issues they bring [6–10].

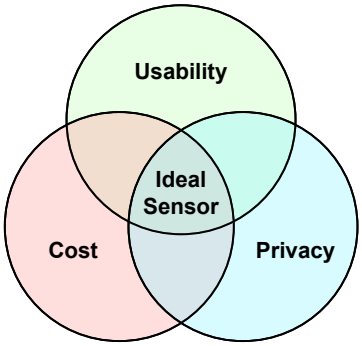


Figure 1.1: An ideal sensor for smart environment should have high usability, low cost and privacy preserving features.

Researchers are actively looking for suitable sensors to support smart environment applications [11]. An ideal smart environments sensors should have three key properties, as shown in Fig. 1.1: 1) Usability: the sensor must be able to collect accurate and useful information under different conditions as required by the downstream application(s); 2) Cost: the sensor should be low-cost for both

production and installation, as typically a large number of the sensors are deployed; 3) Privacy: the sensor should not gather additional sensitive information, such as image and voice, especially when used in smart homes, so that there will not be privacy leakage even if the system is compromised. Based on these criteria, we believe that millimeter wave (mmWave) radar can be one of the best candidates. A comparison of mmWave radar to other sensing modalities can be found in Section 2.4.

mmWave refers to electromagnetic waves with frequency between 30 GHz and 300 GHz, which corresponds to wavelengths in the order of millimeters (1mm to 10mm). mmWave radar typically refers to mmWave Frequency-modulated Continuous-wave (FMCW) radar, whose working principles are introduced in details in Chapter 2. mmWave Radar has many attractive features which brings great potential for wide adoption in smart environments, including:

- **Robustness to lighting and non-line-of-sight conditions** The wavelength in the order of millimeter level enables mmWave radar to penetrate certain objects and materials including plastic, paper, thin wall, smoke, fog, etc. As a result, mmWave radar is not sensitive to lighting conditions, and can be used in dark rooms, e.g., for sleep monitoring. Besides, mmWave radar can be concealed inside smart devices, walls or furniture, which makes it a psychologically non-obtrusive sensor for better user experiences. Also in extreme scenarios such as when the smart environment is on fire, the mmWave radar can sense through smoke [12] to provide useful information (e.g., the environment dynamics and the location of the trapped people) to the fire-fighters.

- **Micro-movement and Vibration Sensing Abilities** Through analyzing the phase variation of the received reflections, mmWave radar can detect very tiny movements, down to a fraction of the wavelength, where the typical wavelength is several millimeters. As a result, mmWave radar can track target displacement in the micron level, opening up a promising space for usage in micro-level sensing applications in smart environments. mmWave radar is

thus a good fit for applications like detecting human vital sign (heart rate and breath rate) [13–15], as well as detecting machine vibration in industrial applications [16]. mmWave radar can also sense the vocal cord vibration of human subjects, for voice recognition and authentication [17–19].

- **Generation of Sparse Point Cloud** mmWave FMCW radar can generate a sparse 4D point cloud for each frame. In self-driving applications, such 4D point cloud can be used for detecting and classifying trees, cars and pedestrians. It can also be used for tracking and counting people, robot navigation and obstacle avoidance, etc. Typical low-cost single-chip mmWave radar can only generate sparse point clouds, because of hardware based resolution limitations [20, 21]. Though the sparsity increases the difficulty for algorithm design, it makes this sensing modality inherently privacy preserving, compared to fine-grained point cloud generated with LiDAR or RGB image with camera. Such feature makes it suitable for applications like sleep monitoring in the bedroom and fall detection in the bathroom, where privacy is a very important concern.

However, early mmWave radars are quite bulky in size and also very expensive, which limited their adoptions for ubiquitous computing. mmWave radar sensing technologies were mainly applied in two areas five years ago, before the development of single-chip solutions: assisted driving and military use. For automobiles, mmWave radars are installed for obstacle detection and collision avoidance, which dates back to 1990s [22]. mmWave radar can generate point clouds with radial velocity information, producing a 3D point cloud (x, y, velocity) or a 4D point cloud (x, y, z, velocity), which can greatly benefit downstream decision making algorithms. For example in self-driving cars, when following the vehicle ahead, the algorithm can make decisions based not only on the distance to the vehicle in front, but also on the relative velocity. Together with other sensors like LiDAR and cameras, millimeter wave radars can make auto-piloting of cars more feasible in the real world [23–26].



(a) TI AWR1642[1] **(b)** Infineon BGT60LTR11AIP[2] **(c)** Google Soli[3]

Figure 1.2: Single-chip mmWave FMCW radar.

Also, the sensing abilities of mmWave radar can be used to enhance the vision systems in military vehicles, e.g., a carrier aircraft [27].

In recent years, mmWave FMCW radar has become increasingly popular and greatly expanded its usage in smart environments and Human-Computer Interaction (HCI) applications, thanks to the development of single-chip radars. Texas Instruments presented the world's first single-chip mmWave radar, bringing down the price to tens of dollars and making it affordable for low-end cars and smart devices [28, 29]. Infineon and Google's Advanced Technology and Projects (ATAP) group developed a tiny mmWave FMCW radar, called 'Soli' [30], which was first presented at Google Developer Conference in 2016. Soli is smaller than a coin and has already been installed on Google Pixel 4 smart phones for mid-air gesture interactions [31]. The ongoing decrease in size and price of mmWave radar brings possibilities for it to be widely embedded into smart home devices, and to enable different kinds of device-free human-centered applications, e.g., vital sign monitoring, gesture recognition, human identification, intruder detection, sleep monitoring, etc. Because of the fast advancement in mmWave sensors and algorithms, we believe that we are moving towards an era of ubiquitous mmWave sensing.

As demonstrated in Fig. 1.3, we envision that in the future single-chip mmWave radar sensors will be an indispensable part of smart homes, and will be installed in multiple locations in the house, e.g., inside smart devices, walls and furniture.

[1] https://www.ti.com/tool/AWR1642BOOST

[2] https://www.infineon.com/cms/en/product/evaluation-boards/demo-bgt60ltr11aip/

[3] https://atap.google.com/soli/

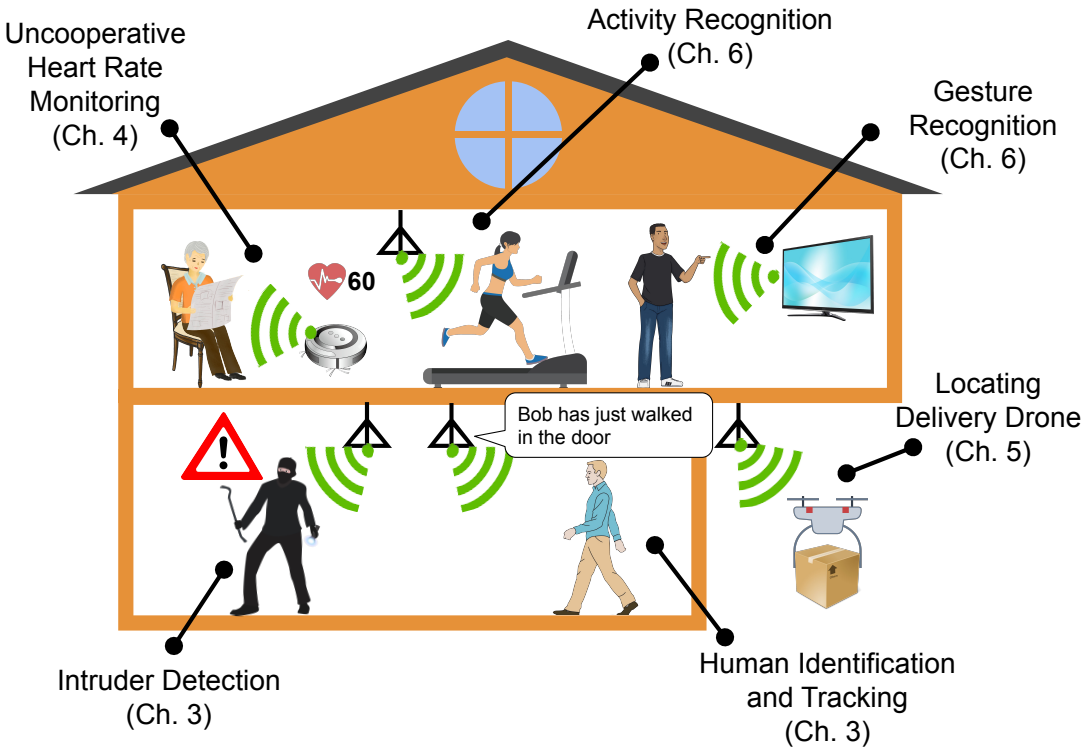


Figure 1.3: Works in this thesis are aiming to build a smarter environment with mmWave sensing technologies.

The mmWave radar sensing area will cover a majority of the space in the house. mmWave radars can localize and identify human subjects, as well as detect intruders. mmWave sensors are embedded inside smart devices, such as smart TVs, to allow users to interact with these devices through mid-air gestures, so there is no need to look for the remote control anymore. Also, through sensing human motion with mmWave radar, the smart home is able to know what a person is doing based on activity recognition algorithms, and infer their intentions and needs. With robot mounted millimeter wave radar, the heart rate of specific human subjects can be monitored seamlessly without requiring active human cooperation, for downstream applications like health monitoring, sleep monitoring or post-operative recovery monitoring. Sensing non-human objects is important as well in future smart homes, such as tracking and guiding delivery drones, or detecting and tracking a hostile foreign drone. And most importantly, all of the above-mentioned sensing is done in a privacy-preserving way. In later chapters of this thesis, we provide in-detail

discussions around these topics.

The key for supporting these applications is the ability to accurately and robustly track and recognize targets. mmWave radar tracking can be done on two levels: macro and micro. On the macro level, tracking is to predict the position of the target, such as the location of the human subject or the drone, while on micro level tracking often refers to extracting displacement patterns, such as the vibrating frequency of motors in industrial monitoring and heart rate/breath rate in vital sign monitoring. Recognition is also a critical part of the smart home system. Recognizing the reflection patterns from user movements can help us identify the user, read the user's command through mid-air gestures, and tell what the user is currently doing. Such information is vital for the smart home intelligence to respond to a user's requests, predicting the user's intentions, and offer the best possible services to the user.

However, developing algorithms for mmWave radar tracking and recognition applications is not trivial, especially when using conventional methods. The tracking performance is affected by many factors, e.g., noise, multi-path effect, and limited resolution. As a result, conventional tracking algorithms have difficulties in producing accurate and robust results. Besides, it is often very difficult to design hand-crafted features for recognition tasks from the radar data like sparse point cloud or radar heatmaps.

Deep learning has attracted huge amounts of academic and commercial interest in recent years for its strong abilities in feature learning and has demonstrated exceptional performances in many research areas. One of the biggest advantages of Deep Learning is that no manual feature extraction is required, which greatly lowers the burden for mmWave radar recognition system designers. Also through training it is able to produce better predictions for object tracking and recognition which best suits the target task and environment. Deep learning can sometimes be even more efficient than traditional algorithms with the help of high-performance GPUs. We believe that deep learning techniques have huge potential for mmWave radar signal processing. However, using deep learning in the mmWave radar

data processing chain is not adequately studied. As a result, our main research motivation of this thesis is to answer:

How can we benefit from deep learning within the radar processing pipeline to increase sensing accuracy and robustness to better support smart environment applications?

1.2 Research Challenges and Questions

Though mmWave radar has the potential of being widely adopted in the future, there are still many challenges in fully utilizing its potential. In this section we introduce the research challenges of mmWave radar tracking and recognition in the context of smart environment sensing, and raise research questions that we aim to answer in this thesis.

1.2.1 Challenge 1: Point Cloud Target Recognition

Point clouds are one of the most commonly adopted data representations of mmWave radar. Unlike LiDAR which manually scans the environment and generates a fine-grained dense point cloud, typical low-cost single-chip mmWave FMCW radar can only generate an sparse point cloud due to MIMO array size limitations. mmWave radar receives the reflections from different objects in the environment all at once, and identifies the targets in the generated heatmap by peak detection algorithms, such as Constant False Alarm Rate (CFAR). The density of the generated point cloud also depends largely on the parameters of the peak detection algorithm, and a high SNR requirement will make the generated point cloud even sparser. The sparsity of point cloud has little impact on tracking algorithms, since the aim of tracking algorithms is just to estimate a location. However, in smart home applications we may also wish to recognize the target, e.g., the human subject, to offer better personalized services. This can be quite difficult, as targets often appear in point cloud as limited number of points, and it can be very hard to estimate the shape and size of the target, or to manually extract features from the point cloud. Especially for

targets that are similar, e.g., two people, it would be extremely hard to distinguish them through conventional methods. As a result, the first research question is:

Research Question 1 *How can we recognize and track targets from radar point clouds with the help of deep neural networks?*

1.2.2 Challenge 2: Micro Displacement Pattern Tracking

mmWave radar is able to track targets on both macro and micro level. On a micro level, we are often more interested in estimating the pattern of reciprocating motions, e.g., sensing the vibration frequency of motors in industrial environments, vital sign monitoring and voice recognition through the vibration of vocal cords. The micro-displacement of the target results in a phase shift of the receiving signal, from which the frequency can be extracted through conventional methods like Fourier Transform, peak counting, auto-correlation, etc. However, in practice, conventional methods have two drawbacks. First, the sensing angle and distance impacts the accuracy a lot, and conventional signal processing based methods do not always have accurate measurements. Besides, with conventional methods, it is not easy to tell whether our prediction is accurate or not, which is very important especially for healthcare applications like vital sign monitoring. As a result, we raise the second research question based on the above mentioned challenge:

Research Question 2 *How do we track micro-displacement patterns accurately and robustly with deep neural networks and estimate the prediction uncertainty?*

1.2.3 Challenge 3: Accurate Macro Movement Tracking

As mentioned before, besides interacting with human, localizing other intelligent agents is also an important function for smart environments, which can be a key component for applications like locating and monitoring a delivery drone. mmWave radar can generate point cloud and heatmaps, which can be used to estimate the location of the targets. Localizing the target seems to be straight forward: with point cloud, we can use the center of the target point cluster as the location prediction; and with range-angle heatmaps, we can use the index of the peak

corresponding to the target as the location prediction. However in practice, the tracking quality is not always guaranteed. Noise, multi-path effect and limited resolution can result in performance downgrade with conventional signal processing algorithms, and even lead to extremely unreasonable results sometimes. Besides, the signal processing models typically does not take the characteristics of the target and the environment into consideration, it is likely to produce sub-optimal results. As a result, we define the third research question as:

Research Question 3 *Can we track target macro movements more accurately and robustly using learning based techniques?*

1.2.4 Challenge 4: Feature Extraction from Raw Data

Deep learning has been widely adopted in mmWave radar applications in recent years, especially in recognition tasks. A majority of the existing approaches are hybrid methods, based on conventional signal pre-processing and downstream deep learning classifier. For conventional pre-processing, Discrete Fourier Transform (DFT) is mostly adopted to generate radar heatmaps, whose absolute value are used as input into deep neural network classifier. DFT can extract frequency information from ADC samples in a chirp for target distance estimation, and Doppler information across chirps, as well as angle-of-arrival (AoA) along receiver antenna array. Despite the fact that DFT is an simple and universal way to extract range, velocity and AoA information from raw radar data cube, such fixed process might not be the optimal choice for every single task and scenario, leading to sub-optimal recognition accuracy. With strong feature extracting ability, deep learning has the potential to directly take raw radar data cube as input and produce better results than the hybrid methods through end-to-end training. However, directly extracting features from raw radar data is not an easy task. The main difficulty here is that raw radar data is complex-valued, and both the magnitude and phase information are equally important, so we need to keep and analyze them both. This is a great challenge for deep learning, as most deep neural networks are real-valued, and can only take real-valued inputs. So here we have the fourth research question:

Research Question 4 *How can we build an end-to-end deep neural network to directly extract information from raw radar data to benefit motion recognition applications?*

1.3 Contributions

Our works included in this thesis are mainly focused on answering the above mentioned research questions. Each contribution introduced in this section aims to address a corresponding research question listed in the previous section. The contributions are as follows:

Contribution 1 In Chapter 3 we present a human tracking and identification system called 'mID', which uses deep neural networks for human target identification through the sparse point cloud, addressing Question 1 in the previous section. To the best of our knowledge, this is the first work to use mmWave radar point cloud for human identification. In the system proposed, the point cloud is first segmented into different clusters, which are associated across frames for tracking. The sequence of point clusters that belong to the human subject is voxelized and sent into deep neural network for identification. Furthermore, we demonstrate intruder detection through classifying the latent features with a random forest classifier. Part of the proposed 'mID' system introduced are included in the following publications:

- **Peijun Zhao**, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, Andrew Markham. "mID: Tracking and Identifying People with Millimeter Wave Radar." In *The 15th International Conference on Distributed Computing in Sensor Systems (DCOSS'2019)* (Best Paper Nominee).

- **Peijun Zhao**, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, Andrew Markham. "Human Tracking and Identification through a Millimeter Wave Radar." In *Ad Hoc Networks*.

- **Peijun Zhao**, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, Andrew Markham. "Poster: Deep Gait Recognition via

Millimeter Wave." In *2019 European Conference on Wireless Sensor Networks (EWSN'2019).* (Poster)

Contribution 2 In Chapter 4 we present 'mBeats', a deep learning based heart rate monitoring system with a robot mounted mmWave radar. In this work, we show that when monitoring heart rate in an uncooperative way, deep neural networks can better extract the micro-displacement information and produce more accurate heart rate estimation than conventional signal processing methods. Furthermore, we can also model and estimate the uncertainty of the prediction based on latent features, which could be vital for health monitoring as trusting wrong predictions could lead to serious consequences. Part of Chapter 4 is published in the paper:

- **Peijun Zhao**, Chris Xiaoxuan Lu, Bing Wang, Changhao Chen, Linhai Xie, Mengyu Wang, Niki Trigoni, Andrew Markham. "Heart Rate Sensing with a Robot Mounted mmMave Radar." In *2020 IEEE International Conference on Robotics and Automation (ICRA'2020).*

Contribution 3 Another contribution we claim is presented in Chapter 5, which is a 3D indoor drone tracking method based on deep learning. Compared to other indoor drone tracking systems, such as optical based (e.g., Vicon) and UWB, tracking with mmWave radar is much cheaper and easier to setup. Rotating propellers on drones are naturally strong reflectors to the radar and no other tag or drone modification is required. Though it is quite straight forward to use conventional signal processing methods for target tracking, we find that it often experience performance downgrade for reasons including limitation of resolution, drone orientation, multi-path, etc. We propose using a deep neural network which takes DFT range-angle heatmaps as input, and can estimate the drone location with approximately 10cm of error on our own collected dataset, more than 3 times better than conventional signal processing baselines. Part of this chapter is published in:

- **Peijun Zhao**, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, Andrew Markham. "3-D Motion Capture of an Unmodified Drone with Single-chip Millimeter Wave

Radar." In *2021 IEEE International Conference on Robotics and Automation (ICRA'2021)*.

Contribution 4 In Chapter 6 we propose a method of replacing the DFT preprocessing in conventional signal processing and deep learning hybrid architectures with stacked complex linear layers, to directly extract information from raw radar data cube and build end-to-end deep neural networks for recognition tasks including gesture/activity recognition. This type of neural network design could extract more useful and informative features from raw radar data than conventional DFT, which could consistently improve the classifier performance. We provide a comprehensive and extensive evaluation on three datasets (hand gesture, arm gesture, activity) we collected, as well as ablation study on complex layer weight initialization, network structure, complexity analysis and running time comparison on both PC and Raspberry Pi platforms. Part of Chapter 6 are summarized in the following paper, which is currently under review:

- **Peijun Zhao**, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, Andrew Markham. "CubeLearn: End-to-end Learning for Human Motion Recognition from Raw mmWave Radar Signals." (Under Review)

During my time as a D.Phil. student, I have also actively participated in and contributed to many other research works. They are not discussed here since they are less related to the main topic of this thesis. My other contributions include:

- Changhao Chen*, **Peijun Zhao***, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, Niki Trigoni. "Deep-Learning-Based Pedestrian Inertial Navigation: Methods, Data Set, and On-Device Inference." In *IEEE Internet of Things Journal (IOT Journal)*. (Co-primary Author)

- Chris Xiaoxuan Lu, Yang Li, **Peijun Zhao**, Changhao Chen, Linhai Xie, Hongkai Wen, Rui Tan, Niki Trigoni. "Simultaneous Localization and Mapping with Power Network Electromagnetic Field." In *The 24th Annual International Conference on Mobile Computing and Networking (MobiCom'2018)*.

- Chris Xiaoxuan Lu, Bowen Du, **Peijun Zhao**, Hongkai Wen, Yiran Shen, Andrew Markham, Niki Trigoni. "Deepauth: In-situ Authentication for Smartwatches via Deeply Learned Behavioural Biometrics." In *2018 ACM International Symposium on Wearable Computers (ISWC'2018).*

- Chris Xiaoxuan Lu, **Peijun Zhao**, Bowen Du, Hongkai Wen, Andrew Markham, Stefano Rosa, Niki Trigoni. "Automatic Face Recognition Adaptation via Ambient Wireless Identifiers." In *The 16th ACM Conference on Embedded Networked Sensor Systems (SenSys'2018).* (Poster)

- Chris Xiaoxuan Lu, Yuanbo Xiangli, **Peijun Zhao**, Changhao Chen, Niki Trigoni, Andrew Markham. "Autonomous Learning of Speaker Identity and Wifi Geofence from Noisy Sensor Data." In *IEEE Internet of Things Journal (IOT Journal).*

- Yuanbo Xiangli, Chris Xiaoxuan Lu, **Peijun Zhao**, Changhao Chen, Andrew Markham. "iSCAN: Automatic Speaker Adaptation via Iterative Cross-modality Association." In *Continual and Multimodal Learning for Internet of Things (CML-IOT'2019).* (UbiComp'2019 Workshop)

- Wei Wang, Muhamad Risqi U Saputra, **Peijun Zhao**, Pedro Gusmao, Bo Yang, Changhao Chen, Andrew Markham, Niki Trigoni. "DeepPCO: End-to-End Point Cloud Odometry through Deep Parallel Neural Network." In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'2019).*

- Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, **Peijun Zhao**, Niki Trigoni, Andrew Markham. "Atloc: Attention Guided Camera Localization." In *The 34th AAAI Conference on Artificial Intelligence (AAAI'2020).*

- Chris Xiaoxuan Lu, Stefano Rosa, **Peijun Zhao**, Bing Wang, Changhao Chen, John A Stankovic, Niki Trigoni, Andrew Markham. "See Through Smoke: Robust Indoor Mapping with Low-cost mmWave Radar." In *The*

18th International Conference on Mobile Systems, Applications, and Services (MobiSys'2020).

- Chris Xiaoxuan Lu, Muhamad Risqi U Saputra, **Peijun Zhao**, Yasin Almalioglu, Pedro PB de Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, Andrew Markham. "milliEgo: Single-chip mmWave Radar Aided Egomotion Estimation via Deep Sensor Fusion." In *The 18th Conference on Embedded Networked Sensor Systems (SenSys'2020).*

- Bing Wang, Changhao Chen, Zhaopeng Cui, Jie Qin, Chris Xiaoxuan Lu, Zhengdi Yu, **Peijun Zhao**, Zhen Dong, Fan Zhu, Niki Trigoni, Andrew Markham. "P2-Net: Joint Description and Detection of Local Features for Pixel and Point Matching." In *2021 International Conference on Computer Vision (ICCV'2021).*

- Wei Wang, Bing Wang, **Peijun Zhao**, Changhao Chen, Ronald Clark, Bo Yang, Andrew Markham, Niki Trigoni. "PointLoc: Deep Pose Regressor for LiDAR Point Cloud Localization." In *IEEE Sensors Journal.*

1.4 Thesis Outline

The remaining chapters of this thesis are organized and presented as follows: in Chapter 2 we provide a review of the background of this work, including the principles of mmWave FMCW radar, common neural network structures used in this thesis, and a literature review of the related topics in mmWave radar sensing; in Chapter 3 we present a novel human tracking and identification system called 'mID', which is the first to use sparse point cloud for human identification; we present using deep learning methods to track micro displacement for heart rate estimation in Chapter 4, and to track macro movements for tracking an unmodified drone in Chapter 5; in Chapter 6 we introduce our proposed learnable pre-processing module; we give a discussion on combining deep learning with different radar data representations and point out potential future directions in Chapter 7 and Chapter 8 finally concludes this thesis.

2
Background

Contents

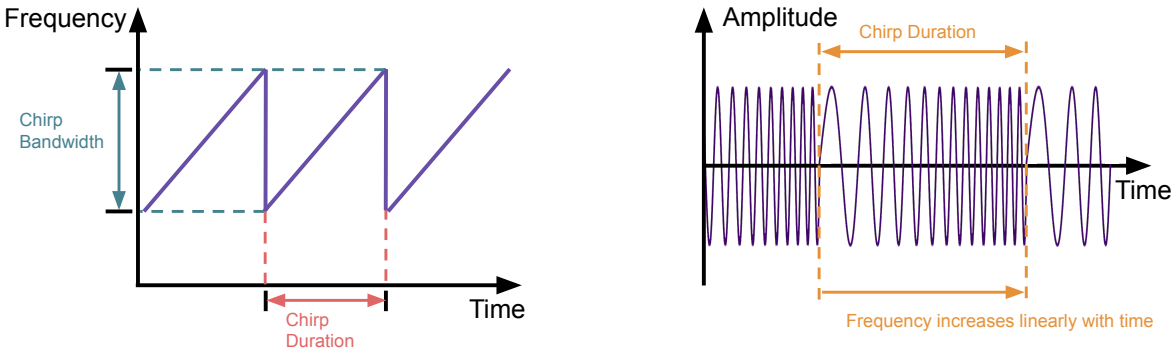
2.1 Principles of mmWave FMCW Radar 17
2.1.1 mmWave FMCW Radar signal model 17
2.1.2 Maximum Detection Range 19
2.1.3 Range Estimation 20
2.1.4 Micro-movement and Velocity Estimation 23
2.1.5 Angle of Arrival Estimation 24
2.1.6 Point Cloud Generation Pipeline 26
2.2 Neural Networks 28
2.2.1 Fully Connected Layers 28
2.2.2 Convolutional Layers 29
2.2.3 Recurrent Layers 30
2.2.4 Classification and Regression 32
2.2.5 Complex-weight Neural Networks 33
2.3 mmWave radar sensing methods for smart environments 34
2.3.1 Target Detection and Tracking 34
2.3.2 Micro Displacement Sensing 35
2.3.3 Human Pose Estimation and Identification 37
2.3.4 Gesture and Activity Recognition 38
2.3.5 Summary of Recent Breakthroughs in mmWave FMCW Radar Sensing 39
2.3.6 Discussion 40
2.4 Other Smart Environment Sensing Solutions 42
2.4.1 Camera based Solutions 43
2.4.2 Acoustic based Solutions 43
2.4.3 RF based Solutions 45

In this chapter we first introduce some basic principles of mmWave FMCW Radar, as well as the conventional signal processing methods. As deep learning methods are heavily used in this thesis, we provide an introduction of the basic neural network structures in the second section. We also offer a comprehensive review breakthroughs in mmWave radar sensing in recent years, and a comparison to other sensor modalities.

2.1 Principles of mmWave FMCW Radar

2.1.1 mmWave FMCW Radar signal model

FMCW radar refers to Frequency-modulated Continuous-wave radar. Unlike typical Continuous-wave (CW) radar that transmits signal at a fixed frequency, FMCW radars modulate the transmitting frequency, which allows it to sense the object distance in addition to Doppler. There are many modulation patterns for FMCW radars, such as sawtooth, triangular, rectangular, and staircase. Different modulations have different properties [32]. Sawtooth modulation is the most common modulation for mmWave FMCW Radar, and in many papers sawtooth modulation is assumed by default [33, 34]. In this thesis we only focus on mmWave FMCW radar with sawtooth frequency modulation.



(a) Sawtooth modulation in frequency domain **(b)** Chirps in time domain (real part)

Figure 2.1: Sawtooth modulation in frequency and time domain.

To sense the environment, mmWave FMCW radar emits frequency-modulated chirps with wavelengths in the order of millimeters. A chirp is short period of electromagnetic emission whose frequency increases linearly with time, as shown in Fig. 2.1.

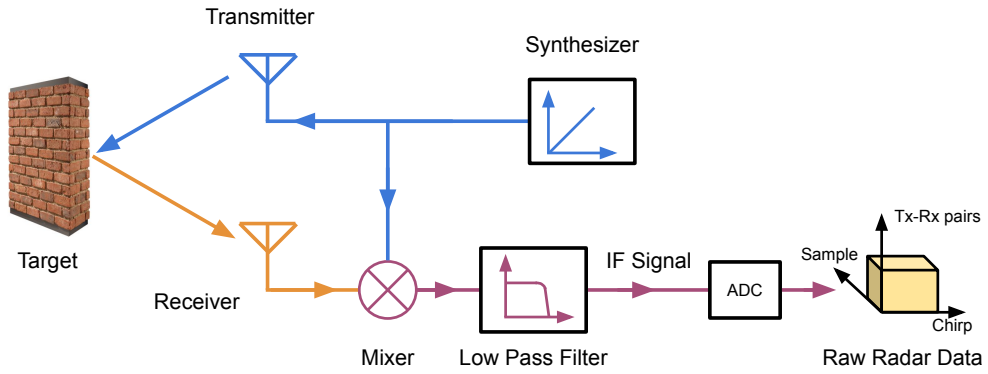


Figure 2.2: Basic mmWave FMCW Radar architecture.

The basic architecture of the radar is shown in Fig. 2.2. The received signal is mixed with the transmitting signal with a mixer which basically calculates the frequency difference between the transmitting and receiving signal. The low pass filter (LPF) serves to filter out those unrelated frequency components and produces the IF (Intermediate frequency) signal. Analog-digital Converters samples the IF signal during a chirp.

Note that in Fig. 6.2 only a single Tx-Rx pair is illustrated. Typically mmWave radar, e.g., the mmWave radar series from Texas Instruments, features multiple-input multiple-output (MIMO) antenna array [35], and for each receiver there is a set of mixer, LPF, and ADC for raw data sampling. with Time Division Multiplexing (TDM) [35], for each frame, a mmWave radar is able to generate a raw data cube of size (Chirp $\times$ Sample $\times$ Tx-Rx pairs). The raw data cube is used to extract information with further signal processing (e.g., range estimation, velocity estimation, angle of arrival estimation, etc.), which is followed by higher level application algorithms.

The maximum detection range, typical methods for estimation of range, velocity and angle of arrival, as well as pipeline for generating 4D point cloud are provided in the following subsections. The range, velocity and angle resolution calculation are based on conventional Discrete Fourier Transform (DFT). DFT is often chosen as the processing method for its low complexity (the computational complexity of Fast Fourier Transform (FFT) is only $\mathcal{O}(n \log n)$) and simplicity of hardware

acceleration implementation. Super-resolution algorithms [36–38] can further increase the resolution, with the cost of much higher computational complexity. Deep neural networks [39, 40] can also achieve better resolution than conventional DFT with relatively low computational complexity. However, for data-driven methods, sufficient training data is needed to be either collected or simulated.

2.1.2 Maximum Detection Range

The detection range of a certain target is limited by the Signal-to-Noise Ratio (SNR) [41]. If the reflected signal is too weak, it could be overwhelmed by sensor noise and cannot be detected. The power captured at the receiver antenna P_r reflected from a target at distance d can be written as:

$$P_r = P_t G_t \frac{1}{4\pi d^2} \sigma \frac{1}{4\pi d^2} A_r \tag{2.1}$$

The equation can be divided into five parts, corresponding to the five steps of detection:

- the transmitter transmits the signal ($P_t G_t$)
- the signal travels to the target ($\frac{1}{4\pi d^2}$)
- the signal reflected by the target (σ)
- the signal travels back ($\frac{1}{4\pi d^2}$)
- the signal received by the receiver (A_r)

In the equation, P_t is the transmitter output power, and G_t is the transmitter antenna gain. The signal decays as the square of distance, following a standard spherical loss model. σ is the Radar Cross Section (RCS), which quantifies the reflectance of the target. The larger the RCS an object has, the more electromagnetic wave it reflects, and thus the object is more detectable by radar. A_r is the effective aperture area of the receiver antenna, which can be expressed as:

$$A_r = \frac{G_r \lambda^2}{4\pi} \tag{2.2}$$

where G_r is the receiver antenna gain and λ is the wavelength of the signal.

The noise N at the receiver can be expressed as

$$N = kTF \tag{2.3}$$

where k is the Boltzman constant, T is the antenna temperature and F is noise figure of the receiver antenna which represents additional noise brought by the radar device.

For FMCW radar, according to [41], the SNR can be expressed as:

$$SNR = \frac{P_r T_{meas}}{N} = \frac{\sigma P_t G_t G_r \lambda^2 T_{meas}}{(4\pi)^3 d^4 kTF} \tag{2.4}$$

T_{meas} is the total measurement time, $T_{meas} = NT_c$ where N is the number of chirps and T_c is the chirp time. As the signal is deterministic and noise is random, in FMCW radar processing, longer observation time leads to the accumulation of useful signal and thus improves the SNR [41].

We can set an SNR threshold for target detection, which is a trade-off between false alarms and missed detections. The larger the SNR threshold, the more probability of missed detections. The smaller the SNR threshold, the more probability of false alarms. And according to Eq. 2.4, given a threshold SNR T_{SNR}, we have the following expression for the maximum range as:

$$d_{max} = (\frac{\sigma P_t G_t G_r \lambda^2 T_{meas}}{(4\pi)^3 T_{SNR} kTF})^{\frac{1}{4}} \tag{2.5}$$

From the equation we can see that, larger transmitting power, SNR and observation time will lead to easier detection of the target, and improve the maximum detection range.

2.1.3 Range Estimation

One of the benefits that FMCW radar brings is that the distance to the objects can be easily measured through the analyzing the frequency of the IF signal. Since the transmitting frequency is linearly increasing and the transmitted signal travels at the speed of light, there will be a resultant frequency difference between the transmitting signal and the receiving signal reflected by a specific object, which forms a corresponding frequency component in the IF signal, as shown in Fig. 2.3.

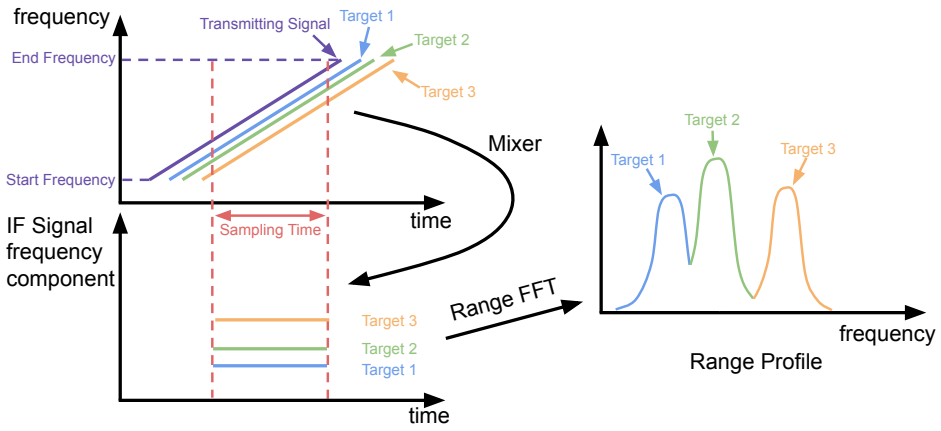


Figure 2.3: A transmitted signal is reflected by three different targets at different distances in the scene. This results in different frequency components in the IF signal, which can be resolved by Range FFT.

Suppose we have a static target at distance d, by calculating the frequency difference of the transmitting signal and receiving signal, we have the frequency of the IF signal corresponding to the target at distance d, which can be expressed as:

$$f = \frac{S2d}{c} \tag{2.6}$$

where S is the frequency slope of the chirp, and c is the speed of light. Thus the distance to the object can be calculated as:

$$d = \frac{fc}{2S} \tag{2.7}$$

For moving targets, as the duration of a chirp is small enough, we tend to ignore the change of target distance during a chirp for simplicity, and estimate the target distance in the same way as static targets.

Objects at different distances result in different frequency components in the IF signal. To estimate the distances to the objects, the Fast Fourier Transform is often applied to the IF signal to extract different frequency components. This is called 'Range FFT', which produces a 'Range Profile'.

Two different objects with the same distance to the sensor cannot be resolved by the Range FFT, as the frequency components they introduce appear at the same location in range profile. In Range FFT based processing, distinguishing two

closely spaced objects requires that the corresponding two frequency components in the IF signal can be resolved in the Fourier domain, in other words the frequency difference needs to be at least

$$\delta f = \frac{F_s}{N} = \frac{1}{T_s} \tag{2.8}$$

where δf represents the minimum frequency difference that can be resolved, F_s is the sample rate, N is the sample number, and T_s is the sampling time.

By substituting Eq. 2.6 into Eq. 2.8, we have

$$\frac{S2d_{res}}{c} = \frac{1}{T_s} \tag{2.9}$$

$$d_{res} = \frac{c}{2ST_s} = \frac{c}{2B_s} \tag{2.10}$$

where B_s is the sampling bandwidth, which is smaller or equal to the chirp bandwidth. We can see that the range resolution depends only on the sampling bandwidth.

For a chirp bandwidth of 4GHz, we have maximum range resolution of around $3.75cm$. Note this is different from range accuracy, as we can always have a more fine-grained frequency spectrum by performing a larger size FFT. The range estimation accuracy depends on sensor SNR.

The sampling rate F_s limits the bandwidth of the IF signal, and thus limits the maximum unambiguous range as

$$d_{max} = \frac{F_s c}{2S} \tag{2.11}$$

Note that the maximum distance is also limited by the SNR requirements introduced before.

The range profile contains information of both static clutter and moving targets in the environment. Often we want to remove the static clutter and keep only the information from moving targets. To do this, we can subtract the average range profile for the corresponding Tx-Rx pair across chirps in a frame. If there is no configuration of frames (i.e., the radar just transmits chirps one by one continuously), we can subtract the average range profile of consecutive chirps with a moving window. This technique is often called 'clutter removal'. It is straight

forward to understand that if we subtract the average from the current range profile, only the peaks corresponding to moving targets would remain.

2.1.4 Micro-movement and Velocity Estimation

With wavelengths in the level of millimeters, mmWave radar can sense target micro-movement down to micron level. A very small change in distance to the detected obstacle would result in a phase shift in the reflected signal, from which micro-movement can be tracked and the speed of the object can be estimated. The phase shift follows:

$$\Delta\Phi = \frac{4\pi\Delta d}{\lambda} \tag{2.12}$$

where Δd is the distance the object moved and λ is the wavelength of the transmitting signal (The wavelength corresponding to the center frequency is used for simplicity). And we can obtain Δd with:

$$\Delta d = \frac{\lambda\Delta\Phi}{4\pi} \tag{2.13}$$

In this way we convert the phase change across chirps to the micro-movement. Note that the Range FFT is able to maintain the phase information of different frequency components, and we use the phase change of the corresponding peak in the range profile across chirps to track certain target.

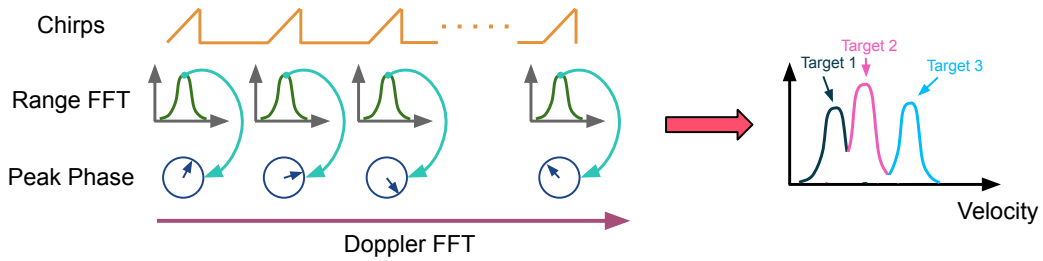


Figure 2.4: Doppler FFT is applied on the peak phase vector across chirps. Targets at the same range with different radial velocities can be differentiated with Doppler FFT.

With the interval between chirps T_c, we can estimate the speed of the object detected as

$$v = \frac{\lambda\omega}{4\pi T_c} \tag{2.14}$$

where ω is the phase variation across chirps. ω is often estimated with the DFT applied on the phase vector of the corresponding peak in the range profile across chirps, which is called 'Doppler FFT', as illustrated in Fig. 2.4. If two or more targets with different speeds appear at the same range, then multiple peaks would appear in the result of Doppler FFT, and thus objects with same range and different radial velocity can be resolved. The velocity resolution of the radar is inversely proportional to the frame time (T_f) and is given by

$$v_{res} = \frac{\lambda}{2T_f} \tag{2.15}$$

As the phase is periodic, to avoid ambiguity, we often set an lower and upper limit on the speed that can be correctly detected:

$$v_{max} = \frac{\lambda}{4T_c} \tag{2.16}$$

Note that only the radial component of the speed can be detected. Tangential velocity does not introduce any phase shift of the receiving signal.

Often the range FFT and Doppler FFT are applied on the raw IF signal together to produce a 'Range-Doppler Heatmap', where each peak corresponds to a detected object at certain range with a certain radial speed.

2.1.5 Angle of Arrival Estimation

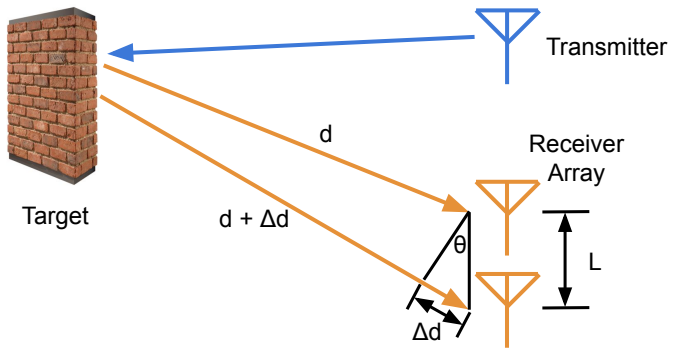


Figure 2.5: The slight difference in the distances from the target to the receivers in the receiver antenna array results in a phase shift, which can be used to estimate the angle of arrival.

Angle of arrival can be estimated with a receiver antenna array (≥ 2 receivers). The signal reflected from the obstacle has to travel a longer distance to reach the antenna which is further away from the obstacle, which leads to a time difference between the two receivers receiving the related signal, reflected by the phase difference, as shown in Fig. 2.5. This can be represented with

$$\omega = \frac{2\pi l \sin(\theta)}{\lambda} \tag{2.17}$$

where l is the distance between consecutive elements in the receiver array, θ is the angle of arrival and λ is the wavelength corresponding to the center frequency of the chirp. The angle of arrival can be calculated as

$$\theta = \sin^{-1}(\frac{\lambda\omega}{2\pi l}) \tag{2.18}$$

As ω is periodic, with $l = \lambda/2$, the above equation can give maximum unambiguous field of view of $[-\pi/2, \pi/2]$.

Similar to range and velocity, angle information can also be extracted with Fourier Transform on the phase vector of different elements across the linear receiver array, called 'Angle FFT'. As a result, the angle resolution is inversely proportional to the number of elements in the receiver array, more specifically,

$$\theta_{res} = \frac{\lambda}{N_r l \cos\theta} \tag{2.19}$$

where N_r represents the number of receivers in the array. Note that the angle resolution changes with the AoA θ. The best AoA resolution can be achieved at $\theta = 0$, i.e., if the object is directly in front of the sensor.

It is often expensive to increase the number of receiver antennas, as each additional receiver antenna requires a set of corresponding components in the hardware chain, including Mixer, LPF, ADC, etc. To lower the cost, current single-chip mmWave radar often features MIMO (Multiple-Input, Multiple-Output) antenna array for better resolution [35], i.e., instead of just increasing the number of receiving antennas, the number of transmitting antennas is also increased. With proper antenna location arrangement, each Transmitter-Receiver pair, also known

as a 'Virtual Antenna', can be used as one element in the receiver array for AoA estimation, equivalent to a receiver antenna in SIMO (Single-Input, Multiple-Output) configurations.

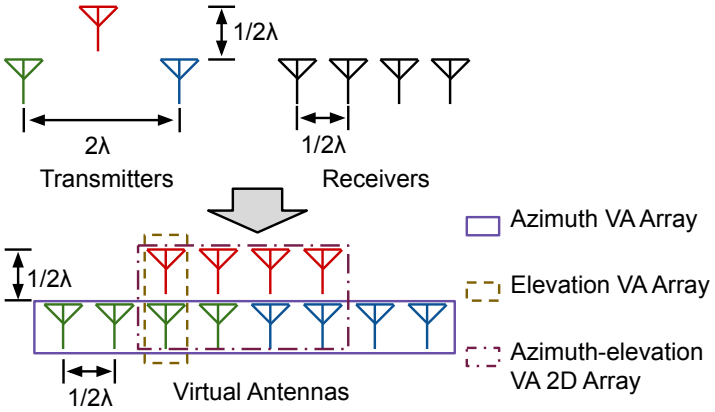


Figure 2.6: TI IWR1443/IWR6843 MIMO layout.

The mmWave sensor from Texas Instruments we used in this thesis (IWR1443 or IWR6843) features a 3Tx-4Rx MIMO antenna array as shown in Fig. 2.6 [35]. It has 8 virtual antennas for azimuth AoA estimation, and has 2 virtual antennas for elevation AoA estimation. It is possible to jointly estimate azimuth and elevation AoA, with a 2x4 2D virtual antenna array.

Receivers in the MIMO array must be able to separate the signal from different transmitters, and the most common way to do this is to use Time Division Multiplexing (TDM) techniques, i.e., to use the transmitters one by one, one at a time. As a result, we will also need to compensate for the phase change due to the movement of the object across different transmitters in a chirp loop, before using the phase information of the virtual antenna array for AoA estimation.

2.1.6 Point Cloud Generation Pipeline

With the above mentioned processing for each frame, mmWave radar is able to produce a point cloud, often represented in 4D (x, y, z, velocity) or 5D (x, y, z, velocity, intensity). Note the velocity here means radial velocity. The pipeline for generating the point cloud is shown in Fig. 2.7.

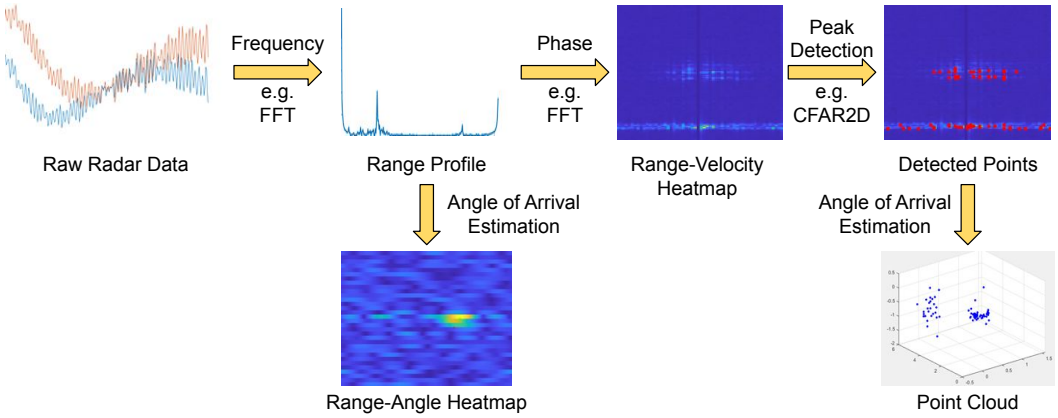


Figure 2.7: Point Cloud generation pipeline. Peak detection algorithms like CFAR2D is applied on Range-Velocity Heatmap. For each detected peak, elevation and azimuth angle are estimated and converted to a point in the point cloud.

The sampled raw IF signal is first transformed into the frequency domain for each chirp with range FFT, then velocity FFT is applied to generate range-velocity heatmap. Each peak in the range-velocity heatmap corresponds to a target in that certain range and with the certain radial speed, and can be detected with 2D peak detection algorithms, e.g., Constant False Alarm Rate 2D (CFAR2D). For each detected peak, the azimuth and elevation AoA is calculated and the peaks are mapped to point cloud. Furthermore, if AoA estimation is directly applied on range profile across VA array, we can get another type of heatmap, namely Range-Angle Heatmap. This type of heatmap provides location information of targets, while excluding the velocity information.

mmWave radar can have different level of data representations throughout the point cloud generation process, from raw radar data at the lowest level, to the point cloud at the highest level. The higher the representation level, the more straight forward the data representation would be (e.g. point cloud is much more easier to understand than range profile). However, there's possible information loss during the process. As a result, downstream applications do not always take point cloud as input. Some works prefer heatmaps [42, 43], range profile [12], Doppler information [44], or even raw IF data [45] as input. We provide a discussion on combining deep learning techniques with different radar data representations in Chapter 7.

In this thesis, we also make use of different level of radar representations as input to deep neural networks for downstream applications. In Chapter 3 we use voxelized point cloud as features for person identification. We use the phase variation signal as input to deep neural network for heart rate estimation in Chapter 4. Range-angle heatmaps are used as input for drone localization introduced in Chapter 5, and finally we propose learnable pre-processing, directly using raw radar signal as input to improve typical hybrid methods for identification tasks in Chapter 6. We now turn to an overview of basic neural network architectures and methods.

2.2 Neural Networks

Deep Neural Networks have been successfully applied on different tasks, e.g., image understanding [46, 47], speech recognition [48], machine translation [49], localization [50, 51], etc., and show exceptional performance. Not only does deep learning improve the task performance compared to conventional shallow learning models, but it also greatly simplifies the algorithm design, for it can automatically extract latent features with little hand-crafting. Deep Neural Networks can be built with different types of layers as building blocks, and trained with back-propagation [52] and optimization methods like stochastic gradient descent (SGD) [53]. Deep neural networks are widely applied in the research in this thesis. In this section we introduce some commonly seen layers that are related to our research, including fully-connected layers, convolutional layers, and recurrent layers. We also introduce how neural networks solve classification and regression problems, corresponding to our research on mmWave radar recognition and tracking, respectively. Finally we introduce complex-weight neural networks, which we use to directly take raw radar signal as input, and can be trained to adaptively extract features from raw radar data.

2.2.1 Fully Connected Layers

Fully Connected Layers are the most basic building blocks of deep neural networks. It works as a linear transformation to the input tensor, plus possible activation to

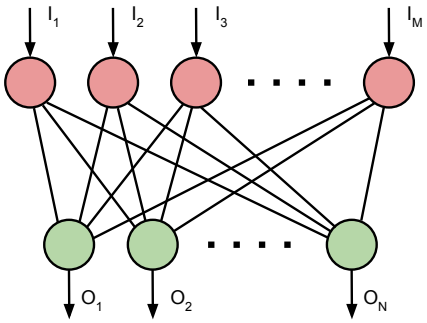


Figure 2.8: Fully Connected Layer

provide non-linearity. Each output node is directly connected all the input nodes, as shown in Fig. 2.8. The value of the j^{th} output node O_j can be written as

$$O_j = \sum_{i=1}^{n} w_{ij} * I_i + b_j \tag{2.20}$$

where I_i is the value of the input node i, w_{ij} is the weight from input node i to output node j, and b_j is the bias of output node j. Fully connected layers used in the hidden layers of the neural network, as well as the output layer for classification and regression problems. If used as hidden layers, fully connected layers are often followed by activation functions to provide non-linearity to the model. If used as output layer for classification problems, the softmax activation (or log-softmax activation) is usually used as the activation function, and for regression problems there's often no activation.

2.2.2 Convolutional Layers

Convolutional layer has also been one of the most common building blocks of deep neural networks. Convolutional neural networks are very good at extracting features from grid-like data, such as images, and have shown exceptional performance on various tasks. Convolutional layers perform convolution operations on local regions of the input data with trainable kernel weights.

Convolutional layers naturally focuses on local features. However, in some cases, ignoring the longer dependencies can downgrade neural network performance. To capture longer dependencies we need larger receptive fields. The most straight

forward way to have larger receptive field is to use larger convolution kernels. A preferable way is to use stacked CNN layers with smaller kernels to replace a single CNN layer with larger kernels, as discussed in VGGNet [54], since the stack of several convolutional layers with smaller kernels is the same as imposing a regularisation on the larger convolutional filters. In this way the network is able to learn a more generalizable and expressive feature space. Max-pooling/Average-pooling layers can be used between convolutional layers to enlarge the receptive field as well. However, the design of pooling layers makes them not trainable, and have information lost. Dilated convolutional layers have same number of pixels in the receptive field, but the pixels are scattered so that the receptive field is much larger. However, we need to design the neural networks with dilated convolutional layers carefully to avoid 'the gridding effect', where the neural network loses some information because of the dilation.

Researchers have also been trying to explore advanced structures based on convolutional layers, such as Inception module, which forms the basic components of GoogleNet [55]. ResNet [56] uses residual connection to solve the gradient vanishing problem when training very deep neural networks and have shown impressive performance, and has been used in many later works for feature extraction [57, 58].

In this thesis we also use convolutional layers heavily to extract features from radar. Radar heatmaps are similar to images, and we can use convolutional layers to extract features from radar heatmaps in the same way as extracting features from images, for drone localization as introduced in Chapter 5. Besides, in Chapter 4 we used 1D convolutional layers to extract frequency information from phase variation signal for heart rate estimation.

2.2.3 Recurrent Layers

Recurrent layers are used to extract temporal dependencies in sequential data, and have been widely applied in problems like voice recognition and machine translation. The structure of standard RNN layer is shown in Fig. 2.9. Standard RNN layer keeps a hidden state h. For each timestamp, the value of the hidden layer is decided

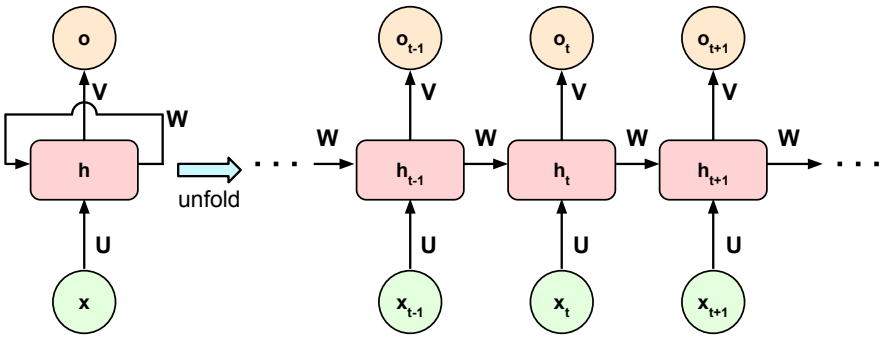


Figure 2.9: A Basic Recurrent Layer.

not only by the input of the current timestamp, but also by the value of the hidden layer in the previous time-stamp, which is given by

$$h_t = f(U \cdot x_t + b_{Uh} + W \cdot h_{t-1} + b_{Wh}) \tag{2.21}$$

where h_t is the hidden state at time t, and x_t is the input at time t. U and W are the weights of the recurrent cell, and b_{Uh} and b_{Wh} are the bias vectors of the weights U and W respectively. f is the activation function of the hidden state. The output of time t can be represented as

$$o_t = g(V \cdot h_t + b_{Vt}) \tag{2.22}$$

where V is the weight multiplied to the hidden state, b_{Vt} is the bias term and g is the activation function of the output.

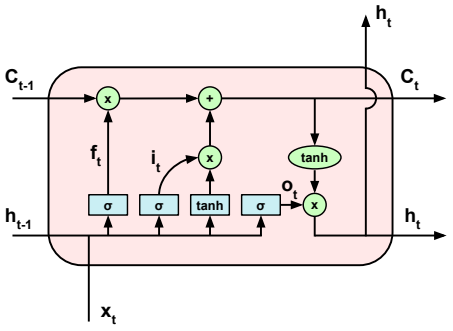


Figure 2.10: Long Short Term Memory Cell Structure.

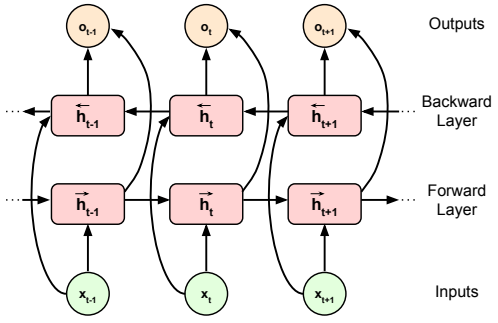


Figure 2.11: Bi-directional Recurrent Layer Structure.

However, in practice, standard RNN layers has difficulties learning the dependencies of long sequences because of the gradient vanishing and gradient exploding

problems [59]. Long Short Term Memory (LSTM) structure is introduced to tackle this problem [59], and is much more widely used for its exceptional performances. The structure of LSTM cell is shown in Fig. 2.10. The key idea behind the design of LSTM structure is that it introduces cell state in addition to hidden state, which is protected and controlled by three gates to make it relatively stable for remembering long-term dependencies. The equations of LSTM can be expressed as

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ C_t &= f_t * C_{t-1} + i_t * tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * tanh(C_t) \end{aligned} \tag{2.23}$$

In some cases extracting the temporal dependencies in both directions can lead to better performance, as sometimes the information from future inputs are also crucial for the output of the current timestamp. Bi-directional recurrent neural networks (Bi-RNN) are proposed to exploit the temporal dependencies on both directions. The structure of Bi-RNN can be viewed as two separate RNN layers with opposite direction, as shown in Fig. 2.11. The hidden state of Bi-RNN at time t is the concatenation of the two hidden states of the separate RNN layers, represented as

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{2.24}$$

Recurrent Neural Networks have been successfully applied to extract features from time-series data to solve various problems. In this thesis we also adopt recurrent layers, LSTM specifically, in our network design. We used bi-directional LSTM to extract temporal information in voxelized radar point cloud for moving human identification, which is discussed in Chapter 3. We also integrate LSTM layers into our neural networks for gesture/activity recognition and drone localization which are introduced in Chapter 6 and Chapter 5 respectively.

2.2.4 Classification and Regression

In machine learning, problems can be roughly divided into two types: classification and regression. Neural networks can solve classification and regression problems

typically with fully connected layers at the end of the network to estimate the predicted label or the regressed values from latent features extracted by previous layers. For classification tasks, the number of the neurons in the last layer (output layer) is the same as the number of classes. Softmax are used as activation functions and the neural network is trained with Cross-entropy loss. For regression tasks often there is no activation function applied on the output layer, and Mean Squared Loss (MSE) is mostly used as the loss function.

In this thesis we adopt deep neural networks for both classification and regression tasks. For human identification introduced in Chapter 3 and gesture/activity recognition in Chapter 6 we use neural networks for classification; and in Chapter 4 and Chapter 5 we use neural networks to regress heart rate and drone location, respectively.

2.2.5 Complex-weight Neural Networks

Real-valued neural networks are well-developed and widely adopted in the recent decade. However, real-valued networks have difficulties in processing complex data. To use complex-values as input to real-valued neural networks, the complex input has to be converted to real values in some ways, such as taking the absolute value, processing real and imaginary parts separately, or use magnitude and phase representations. These methods either lose some information, or introduce extra difficulties in network design. Researchers have turned to complex-valued neural networks to process the complex data, among which the most representative one is radar data. Complex neural networks have been used for different radar signal processing tasks, such as extracting features from raw radar signal for drone classification [60], and use complex-valued SAR radar images for target recognition [61–64]. Even for real-valued inputs, using complex-weighted neural networks could also lead to potential performance improvements. For example, Trabelsi et al. demonstrated that on several computer vision tasks and music transcription tasks, complex-valued models can achieve equal or better performance than real-valued networks with the same structures [65].

For decades researchers have been looking for the complex counterparts of real-valued neural network components. For some layers it would be straight forward, such as fully-connected layers, as the operations involved (i.e., multiply and plus) can be applied to both real and complex values. However, for some operations, e.g., some activation functions, there's no direct counterpart in the complex space. As a result, different complex valued versions are proposed. For example, ReLU activation function does not directly have a counterpart in the complex space, so researchers proposed a variety of its correspondences, including ModReLU [66], $\mathbb{C}$ReLU [65] and zReLU [67]. In recent years, there has been an increasing amount of research on complex-valued versions of more sophisticated layers, such as complex-valued recurrent neural network [68, 69] and complex-valued Transformer [70], showing the vitality of this research area.

In this thesis we present end-to-end neural network for gesture/activity recognition, which directly takes complex raw radar signal as input. We adopt stacked complex linear layers to extract range, Doppler and angle of arrival information, which has shown better performance than conventional DFT pre-processing. This work is introduced in Chapter 6.

2.3 mmWave radar sensing methods for smart environments

As introduced in Chapter 1, the development of single-chip mmWave radar has opened up a wide design space for applications in different areas, and has attracted tremendous amount of research interests in recent years. In this section we provide an overview of the related literature, introducing the background information about conventional mmWave sensing methods as well as some latest advances.

2.3.1 Target Detection and Tracking

Conventional FMCW radar target tracking methods are mainly based on point clouds. Often the generated point cloud contains reflections from different targets as well as noise. Clustering algorithms are applied on the point cloud to segment

the point cloud into different parts. DBScan [71] is often chosen as the clustering method, because it performs the clustering based on the distances of point pairs, and does not require the number of clusters as a parameter. Also DBScan is able to ignore some noise by setting the minimum number of points in a cluster.

Tracking mainly involves associating clusters between frames and track filtering. Hungarian algorithm [72], as a well-know method to solve assignment problem, is widely applied to associate clusters between frames. Kalman Filter [73] can be used for correction and prediction of tracks. In complex scenarios, algorithms like Joint Probabilistic Data Association Filter (JPDAF) [74] and Multi-hypothesis Tracking (MHT) [75] are popular multiple target tracking algorithms, which are able to robustly track multiple targets in noisy environments.

In Chapter 3, we adopted Hungarian algorithm and Kalman Filter for people tracking to extract the point clouds for identification.

2.3.2 Micro Displacement Sensing

mmWave FMCW radar can track displacements at micron level, as introduced in Section 2.1. To track the phase change, the first step is to extract the phase information of the target. To do this, we typically perform a range FFT on the raw IF signal first. Fourier Transform keeps the phase information, i.e., the phase of the peak which represents a certain target is the same as the phase of the corresponding frequency component in the IF signal. As a result, we extract the phase variation signal from the corresponding peak in the range profile across chirps. Note that the angle of the complex value is $[-\pi, \pi]$ in a circular way, and we need to apply phase unwrapping to change it to continuous phase variation to avoid '2π jumps'.

Remote vital sign (heart rate and breath rate) monitoring can be one of the most important applications for mmWave radar micro-displacement sensing. Human heart beat typically causes chest displacement of 300-800 micrometers [76], and normal breathing around 4 millimeters [77]. Such displacements can be captured with mmWave radar. mmWave radar vital sign monitoring has the benefit of device-free, non-contact, and privacy preserving features, which has great potential in future

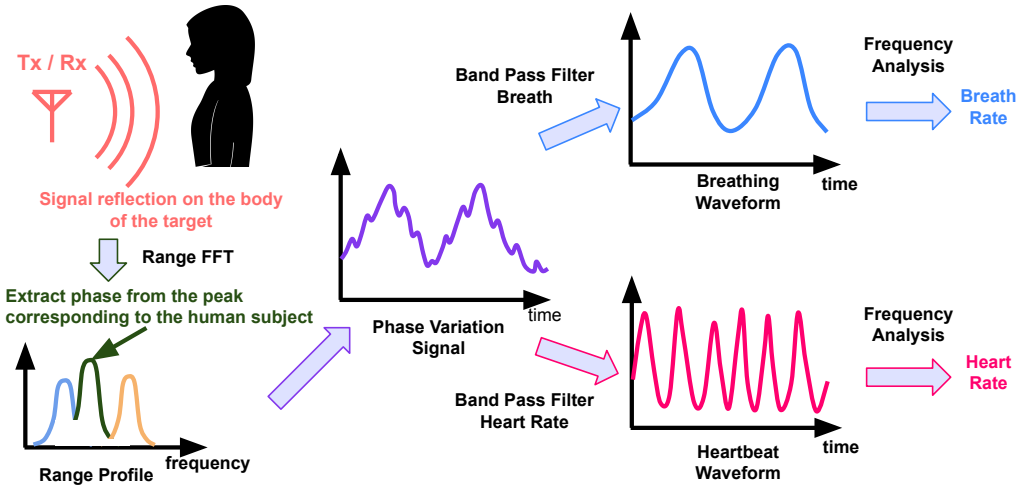


Figure 2.12: Typical mmWave FMCW Radar Vital Sign Monitoring.

smart homes, and is widely studied in recent years [13–15]. Typical method for remote vital sign monitoring is shown in Fig. 2.12. The combination with deep learning techniques further enabled even more applications, like inferring emotions [78], and estimating seismocardiogram [79]. However, remote vital sign monitoring with FMCW radar still has a number of limitations. One limitation is that most pipelines require the user to be in a static pose, like sitting or sleeping in the measurement area of the radar. Though there has been some initial explorations [80, 81], remotely measuring moving people's vital sign with FMCW radar still remains a problem to be solved. Another limitation is the range and angular resolution. If two people are close to each other then it is very hard to separate the radar reflections. However, this might be solved with multiple radars observing at different angles, or install the radar on some moving platforms to focus on a certain person [82].

Micro-displacement sensing can be useful in many other areas besides vital sign monitoring. For example, Jiang et al. proposed mmVib [16] for monitoring machine vibrations in industrial environments. The micro displacement can also be intentionally modified and coded to carry information to be sensed with mmWave radar. In [83], Tonolini and Adib proposed transmitting information across the water-air boundary by using FMCW radar to sense the displacement of water surface caused by the underwater acoustic signals emitted by the acoustic transmitter.

2.3.3 Human Pose Estimation and Identification

Though FMCW radar is not able to produce a fine-grained image like RGB or depth camera, the heatmaps and point cloud could still provide some information about the shape and size of the target. Adib et al. proposed using antenna array of 'T' shape to scan the area of interest and estimate human figure through wall, which can be used to identify the subject based on body shape and size [84]. Zhao et al. proposed learning human pose based on horizontal and vertical heatmaps, from the supervision of RGB video [85]. Sengupta et al. projected the generated point cloud onto X-Y and X-Z planes as inputs to pose estimation network, supervised by Kinect RGBD camera [86]. Xue et. al. further proposed estimating the human body shape in addition to pose using deep learning based on radar point clouds [87]. With pose estimation, mmWave radar could be used for at-home entertainment like motion sensing games, replacing the previous device based methods like marker-based tracking, or privacy intruding sensors like Kinect RGB-D cameras.

Different from [84] where the subject identity is inferred purely based on body shape, most of the FMCW radar human identification are based on the subject moving patterns. Human identification methods can be mainly divided into two types: Doppler signature based [88] and point cloud based [89–91]. Doppler signature is caused by the reflection on human limbs of the moving subject, while point cloud contains both gait and body shape information. Doppler based identification methods typically requires the user to walk in a designated line in order to be recognized, as the radar can only sense radial speed, and changing the walking orientation results in different Doppler effect, which cannot be recognized by the classifier (unless also collected for training). As a result, such system is suitable for places like corridor, but does not perform well in large open space. In this thesis we propose the first mmWave radar point cloud based human identification method [89], which allows the subject to walk freely inside the identification area to be identified, as introduced in Chapter 3.

2.3.4 Gesture and Activity Recognition

The use of FMCW radar on gesture and activity recognition has also been widely explored in recent years. As mentioned in the Chapter 1, Google ATAP and Infineon developed Soli, which is now installed in Google Pixel 4 mobile phone for device-free mid-air gesture control. Soli is smaller than a coin, which can fit perfectly into consumer level devices, like smart phones, smart TVs and smart speakers. Small-shaped mmWave radars has great potential to enable future mid-air human computer interaction techniques, and researchers have been exploring possible ways of realizing it.

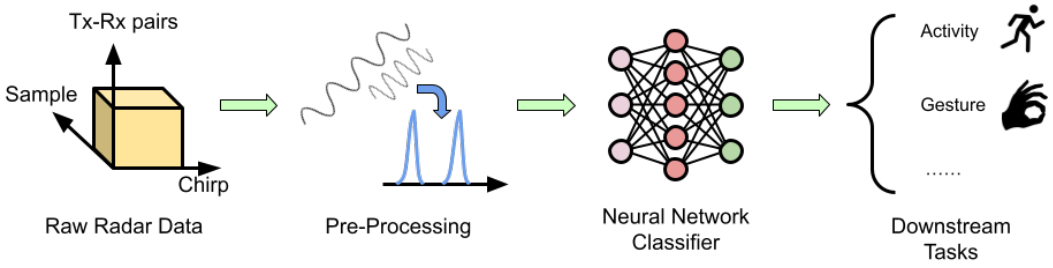


Figure 2.13: DFT Pre-processing + Neural Network Classifier Hybrid Method for Gesture/Activity Recognition.

Most existing mmWave FMCW radar gesture/activity recognition methods are based on the pipeline shown in Fig. 2.13. For consecutive frames, the radar is able to generate a data cube, with three axis as sample, chirp, and Tx-Rx pairs, respectively. Pre-processing is applied on radar data cube for each frame, to extract the frequency information along each axis in the data cube, and estimate range, doppler and angle of arrival. Different types of heatmaps can be generated according to downstream classifier requirements. Pre-processing is mostly done with Fast Fourier Transform (FFT), because of its high efficiency and the mature hardware level implementation to speed up embedded signal processing. Many existing methods sends the heatmaps directly to deep neural network classifiers [43, 92]. In other methods, hand-crafted features are extracted from the pre-processed data, and used as input to shallow [30] or deep [93] classifiers.

Besides gesture and activity recognition, such pipeline can also be used for similar motion recognition applications, e.g., fall detection at home for elder people [94], in privacy-sensitive places like bathroom. It can sense elder people falling down when showering, and call for help in time, which is very useful for elderly care.

2.3.5 Summary of Recent Breakthroughs in mmWave FMCW Radar Sensing

mmWave radar sensing solutions are also explored in a wide variety of other research topics. To help the readers get a broad overview of the advancements in mmWave radar sensing technologies, we use Table 2.1 to summarize the breakthroughs in related areas in recent years, including the time when the techniques appeared (officially published), the problems they solve, and brief descriptions of the the methods they use.

In this table we mainly select the high-impact works that: 1) uses mmWave FMCW radars; 2) targets Smart Environment Applications (HCI, human identification and authentication, health, robotics, etc.); 3) have high impact (i.e., published in relatively top conferences and journals). Papers that are not yet peer-reviewed (e.g., pre-prints) are not included here. Also, demo, poster and workshop papers are not included in the list. The works are organized by their official publication time (month/year). Note that the target tasks are classified on a broad scale, e.g., hand gesture recognition [30, 95–97], arm gesture recognition [20, 97–99], hand hygiene technique monitoring [100], America Sign Language (ASL) recognition [101] and mid-air musical instrument interface [102] are all classified as gesture recognition.

The method description mostly involves two parts, especially for those deep learning based methods. The first part is the data representation they use in the proposed system, and the second part is category of the specific deep neural network used. Neural network which has complex structures and cannot be classified into a specific type (e.g., CNN) are simply noted as Deep Neural Networks (DNN).

We aim to provide a high level summary of recent works in mmWave sensing here, and in Chapter 3 to Chapter 5, more comprehensive reviews and discussions of related work around specific topics will be provided, respectively.

2.3.6 Discussion

In previous parts we discuss conventional methods and recent advances in mmWave radar sensing. Despite the fact that in recent years, researchers have made huge amount of progresses using single-chip mmWave radars for industrial applications, robotics and human-computer interaction, there are still many issues and topics waiting to be explored.

With the development of single-chip MIMO mmWave radar, we can get sparse radar point clouds easily in real time, thanks to the increasing power of on-board hardware processing and on chip FFT accelerators. Using radar point cloud for human identification takes both gait and body shape into consideration, overcoming the drawbacks of Doppler-based identification methods mentioned above. Besides, with point cloud, we can track the position of the subject and identify the subject simultaneously. However, identifying human subjects with radar point cloud was an under-developed area (at the beginning of the research in this thesis), because the inherent sparsity of radar point cloud makes it quite hard for target identification. In Chapter 3 we introduce our proposed 'mID', a system to track and identify the human subject while he/she is walking freely in the sensing area.

Conventional signal processing methods on mmWave radar, especially the DFT based methods introduced in Section 2.1, have been well-developed in the last decades. However, their performance on tracking tasks is not always robust, and sometimes strongly affected by outliers. There have been a lot of works on deep learning based target tracking with RGB images in Computer Vision, but using deep neural networks on radar tracking is an underdeveloped area. We present 'mBeats' and 'mDrone' in Chapter 4 and Chapter 5, respectively. mBeats uses a convolutional neural network to estimate the heart rate based on the micro-displacement of the subject's skin and provide the estimation uncertainty, and

Name	Time+	Target Task	Main Method
Soli [30]	07/2016	Gesture Recognition	Manual Feature + RF
Deep Soli [95]	10/2016	Gesture Recognition	Heatmap + DNN
mID* [89]	05/2019	Human Identification	Point Cloud + LSTM + RF
Waveear [18]	06/2019	Voice Sensing	Frequency Spectrum + DNN
FerroTag [103]	11/2019	Object Recognition	Manual Feature + RF
ThuMouse [96]	01/2020	Gesture Recognition	Point Cloud + CNN-LSTM
SuperRF [104]	02/2020	Imaging	Heatmap + DNN
mmGait [90]	04/2020	Human Identification	Point Cloud + DNN
RF-SCG [79]	04/2020	Health	Phase Variation + CNN
ThermoWave [105]	04/2020	Temperature Monitoring	Manual Feature + GAN
mmVib [16]	04/2020	Vibration Sensing	Signal Processing
mm-Pose [86]	05/2020	Human Pose Estimation	Point Cloud + CNN
WaveSpy [106]	05/2020	Object Recognition	Manual Feature + SVM + DNN
mBeats* [82]	06/2020	Health	Phase Variation + CNN
milliMap* [12]	06/2020	Mapping	Point Cloud + cGAN
Osprey [107]	06/2020	Imaging	inverse-SAR
Guan et. al. [108]	06/2020	Imaging	Heatmap + cGAN
MU-ID [109]	07/2020	Human Identification	Heatmap + CNN
Wang et. al. [98]	08/2020	Gesture Recognition	Heatmap + GAN
ExASL [101]	08/2020	Gesture Recognition	Point Cloud + DNN
Pegoraro et. al. [88]	09/2020	Human Identification	Heatmap + DNN + KF
milliEgo* [110]	11/2020	Ego-motion Estimation	Point Cloud + DNN
VocalPrint [17]	11/2020	Voice Sensing	Manual Feature + shallow ML
RFWash [100]	11/2020	Gesture Recognition	Heatmap + DNN
Pointillism [111]	11/2020	Object Detection	Point Cloud + DNN
mHomeGes [97]	12/2020	Gesture Recognition	Point Cloud + CNN
MilliPoint [21]	12/2020	Point Cloud Generation	SAR
Smith et. al. [112]	01/2021	Gesture Recognition	Heatmap + CNN
milli-RIO [113]	02/2021	Ego-motion Estimation	Point Cloud + RNN + KF
Dong et. al. [19]	03/2021	Human Authentication	Manual Feature + CNN
Pantomime [20]	03/2021	Gesture Recognition	Point Cloud + DNN
Smith et. al. [102]	05/2021	Gesture Recognition	Heatmap + CNN
Pegoraro et. al. [91]	05/2021	Human Identification	Point Cloud + DNN + KF
GWaltz [114]	05/2021	Vibration Sensing	Heatmap + Signal Processing
GaitCube [115]	05/2021	Human Identification	Heatmap + DNN
Vid2Doppler [44]	05/2021	Activity Recognition	video processing + DNN
mmMesh [87]	06/2021	Human Pose Estimation	Point Cloud + DNN
mDrone* [116]	06/2021	Tracking	Heatmap + CNN-LSTM
Park et. al. [117]	06/2021	Ego-motion Estimation	Point Cloud + SLAM
m-Activity [118]	06/2021	Activity Recognition	Point Cloud + DNN
M-Gesture [99]	07/2021	Gesture Recognition	Point Cloud + DNN
mmFlow [119]	07/2021	Health	Phase Variation + DNN
XGest [120]	09/2021	Gesture Recognition	Heatmap + DNN
SquiggleMilli [121]	09/2021	Imaging	SAR + cGAN + DNN
CardiacWave [122]	09/2021	Health	Heatmap + DNN
FG-LiquID [123]	09/2021	Object Recognition	Range Profile + DNN
Gong et. al. [80]	09/2021	Health	Range Profile + DNN

+ Official Publication Time.
* Works I contributed to.

Table 2.1: A brief summary on breakthroughs in mmWave FMCW radar sensing for smart environments in recent years. RF: Random Forest; SAR: Synthetic-Aperture Radar; KF: Kalman Filter; cGAN: conditional Generative Adversarial Network.

mDrone tracks macro-movement of an unmodified flying drone in 3D space at around 10cm of error. Both techniques significantly outperform their competing conventional signal processing baselines.

Deep learning radar gesture/activity recognition works are mostly based on DFT pre-processing of the radar IF signal, for extracting range, Doppler and angle of arrival information as input to downstream deep learning classifiers. This seems a very natural way and is adopted by a wide variety of works on different applications. Most existing works on mmWave radar gesture/activity recognition focus on the design of downstream classifiers [42, 92, 93, 97, 124–132]. However, if we go one step back, we may wish to get rid of the conventional frame and extract patterns directly from raw data. Deep neural networks have been proven to be extremely good at extracting latent features, and we can actually rely on deep neural networks to extract the useful information directly from raw radar data and build end-to-end deep neural networks for gesture/activity recognition. In Chapter 6 we present using end-to-end deep learning method for gesture and activity recognition on FMCW radar, which can be used in different pipelines to improve model performance. To the best of our knowledge, this is the first work to use stacked complex linear layers to replace conventional DFT pre-processing on FMCW radar recognition tasks.

Detailed system design and evaluation of our proposed novel deep learning methods to address the above mentioned issues and challenges are introduced in the following chapters.

2.4 Other Smart Environment Sensing Solutions

In this section we briefly introduce previous proposed smart environment sensing solutions based on other sensor modalities, including camera, acoustic, and other RF sensing modalities such as WiFi CSI and CW radar. We will only provide a high-level overview here as these topics are less related.

2.4.1 Camera based Solutions

Thanks to the fast development of the computer vision communities, vision based techniques are quite mature today. There are many types of cameras, including RGB cameras, RGB-D cameras, IR cameras, etc. With rich visual information, cameras are extremely good at human identification through face recognition [133], re-identification [134–136] and gait recognition [137]. Camera based identification systems have already been widely deployed for a variety of applications such as identity check and looking for missing children. Cameras can also enable different types of interaction techniques, such as 3D human skeleton estimation [138, 139] and gesture recognition [140–142]. Cameras are of the most important sensors for robotics applications, including visual SLAM [143, 144] and re-localization [51, 145, 146]. Also, 3D pose (location and orientation) of objects can be estimated with multiple RGB/IR cameras, which can be used for AR/VR headset/controller localization (e.g., PSVR[1]) and precise motion capturing (e.g., Vicon[2]).

Vision based techniques are quite mature, and has been studied for decades. Despite the cheap price of cameras and the well-developed computer vision algorithms, researchers are still looking for other sensors to replace cameras. This is because cameras capture visual information, which can be privacy intruding, especially if used at home. Besides, vision based sensors are sensitive to lighting conditions, which impacts their performance.

2.4.2 Acoustic based Solutions

Voice user interfaces have been implemented in smart speakers and smart phones, and is able to understand the user's intentions through speech recognition and natural language processing techniques and give corresponding reactions. Apart from recognizing the user's words, researchers have also developed algorithms to extract identity-related features for speaker identification [147] and authentication [148].

[1] https://www.playstation.com/en-us/ps-vr/
[2] https://www.vicon.com/

Besides passively receiving and analyzing acoustic signals with microphone or microphone array, we can also actively transmit acoustic signals by co-located speakers and analyze the reflections to estimate the distance, Doppler, and angle-of-arrival of the targets, which is also known as sonar. With pulse acoustic signals, the sensor is able to tell the distance through Time-of-flight (ToF), which is often seen in ultrasonic transceivers. Many of the early robot SLAM systems are built upon ultrasonic transceivers for sensing the environment [149]. With the speaker continuously transmitting acoustic signal at a certain frequency, the co-located microphone can sense and analyze the frequency of the reflected signal to estimate the Doppler information of the target. The transmitted acoustic signal can also be modulated as sawtooth chirps, and the sensor would work pretty much like the mmWave radar we use in this thesis, which further enables range estimation. The abilities to sense the distance, Doppler, and AoA of target objects enables a series of downstream applications, such as gesture recognition [150, 151], gait identification [152, 153], mobile health [154, 155], etc. Active acoustic sensing can also be done in a way where the speaker and the microphone are separate. For example, in [156], the authors proposed detecting driver phone use by sending acoustic beeps from car speakers, which are received by the mobile phone microphone and the phone's distance from the car's center can be estimated based on time difference of arrival (TDoA).

Acoustic solutions also raise privacy issues as microphones are used, similar to cameras. As a result, people are concerned about privacy leakage when using voice interfaces like smart speakers at home. Besides, typical smart speakers have microphone arrays, which could raise further privacy concerns. If compromised, the smart speaker can be used to send modulated inaudible signals (e.g., FMCW ultrasonic signals) whose reflections can be captured by the microphone array to sense the surrounding environments. Furthermore, the performance of acoustic based sensors can be impacted by environmental noise.

2.4.3 RF based Solutions

Apart from mmWave FMCW radar, researchers have also used other RF sensing techniques for smart environment applications. Commonly seen solutions include using WiFi Channel State Information (CSI), Continuous Wave (CW) radar, and FMCW radar working on other frequencies.

WiFi can be used not only for data transmission, but also for sensing the environment with CSI. One big advantage of using WiFi CSI as sensing modality is that we do not need additional hardware, as WiFi network cards are available in almost every electronic device. WiFi routers, together with the network cards in PCs, laptops and mobile phones forms a local wireless network, and the information about the physical environment is contained in the receiving CSI. Different from radar sensing, the transmitter and receiver is separate in WiFi CSI sensing (e.g., WiFi router as transmitter and laptop as receiver), and the CSI contains multi-path reflections on the receiver end. Researchers have been designing different algorithms to extract information from the WiFi CSI in previous years for smart environment applications, like pose estimation [157], gesture recognition [158, 159], activity recognition [160–162], human identification [163, 164], vital sign monitoring [165, 166], etc., in a device-free manner. However, as multi-path information is contained in WiFi CSI, it could be very challenging to achieve high accuracy in dynamic environments (i.e., other people/objects are moving in the scene at the same time) [167]. Besides, most motion recognition methods rely on machine learning algorithms, which brings difficulties when trying to generalize to other environments [167].

CW radar transmits signal at a certain frequency continuously, and measures the Doppler information of the target object by analyzing the frequency variation of the receiving signal. Since CW radar do not require frequency modulation, it is cheaper than FMCW radar. With CW radar, the Doppler information can be used to sense and recognize human motions, and support applications like vital sign monitoring [168], gesture recognition [169, 170], activity recognition [171] and gait recognition [172]. However, as CW radar does not provide information regarding target distance, so it does not support applications like static object localization and recognition.

FMCW radars can also use electromagnetic waves at other frequencies. Generally speaking, at lower frequencies, electromagnetic waves have better penetration abilities, with potential less reflections on objects. In order to support smart environment applications like human motion sensing, we would want electromagnetic waves to reflect on human body rather than penetrate, so electromagnetic waves at very low frequencies are not applicable. Also low center frequencies limit the accuracy of micro-displacement and speed measurements, as we can tell from Eq. 2.15. On the other hand, electromagnetic waves at high frequencies also have limitations. With higher frequencies, the electromagnetic wave attenuates faster, which limits its sensing range, as shown in Eq. 2.5. Besides, increasing the frequency would result in weaker penetration ability, which also loses the advantage in certain applications, such as sensing in smoke-filled environments. mmWave is found to be one of the most suitable electromagnetic wave to be used for smart environment sensing, as the sensing range is abundant, the Doppler resolution is relatively high, and the penetration and reflection properties supports a wide range of applications.

Through introducing smart environment solutions based on other sensor modalities, we could see that the mmWave FMCW radar we use in this thesis has its own advantages compared to other sensors. mmWave FMCW radars have far less privacy concerns compared to vision based and acoustic based techniques, and are more robust against lighting conditions and noise. It is also less environment dependent than WiFi CSI, and the environment reflections can be removed through the 'clutter removal' algorithm we introduced previously. With FMCW modulation, it can measure distances to targets in the scene, which provides more information than CW radars to support richer downstream applications. And also, wavelength of millimeter level provide a good trade-off between reflection, penetration and sensing range, and is very suitable for smart environment applications. In the next chapters, we introduce our contributions towards answering the research questions raised in Chapter 1.

3

People Tracking and Identification with mmWave Radar Point Cloud

Contents

3.1 Introduction 49
3.2 System Requirements 51
3.3 System Design 52
3.3.1 mmWave Radar Point Cloud Generation 53
3.3.2 Point Cloud Clustering 54
3.3.3 Moving Object Tracking 55
3.3.4 User Identification 58
3.3.5 Intruder Detection 59
3.4 Implementation 61
3.4.1 Testbed Setup 61
3.4.2 Dataset Collection 62
3.4.3 mID Configuration 63
3.5 Evaluation 65
3.5.1 Sensitivity Analysis 65
3.5.2 User Tracking 66
3.5.3 Identification Evaluation 67
3.5.4 Intruder Detection Evaluation 72
3.6 Discussion on Scalability and Robustness 73
3.7 Related Work 75
3.7.1 Device-Free Gait Recognition 75
3.7.2 Other RF-based Human-centric Applications 76
3.8 Chapter Summary 76

mmWave radar point clouds are widely used for object tracking and counting, however, the inherent sparsity of the point cloud brings difficulties in target recognition. Previous radar human identification methods are mainly based on Doppler signatures caused by the reflection from the moving human subject. As a result, this is sometimes termed 'gait recognition' in the literature. We conjecture that point cloud based human identification could have better performance than Doppler based methods, as the radar point cloud carries richer information including both the body shape and the gait. In this chapter we introduce 'mID', a system to track and identify human subjects with mmWave radar. The proposed system consists of four modules which work in a sequential manner, namely: point cloud generation, point cloud clustering, tracking, and identification. The proposed method is evaluated on our own collected dataset and demonstrates good performance. In addition we also show how intruders can be detected, which is important for smart home systems.

Portions of this work is included in the following publications:

- **Peijun Zhao**, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, Andrew Markham. "mID: Tracking and Identifying People with Millimeter Wave Radar." In *The 15th International Conference on Distributed Computing in Sensor Systems (DCOSS'2019)* (Best Paper Nominee).

- **Peijun Zhao**, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, Andrew Markham. "Human Tracking and Identification through a Millimeter Wave Radar." In *Ad Hoc Networks*.

- **Peijun Zhao**, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, Andrew Markham. "Poster: Deep Gait Recognition via Millimeter Wave." In *2019 European Conference on Wireless Sensor Networks (EWSN'2019)*. (Poster)

3.1 Introduction

Knowing 'who is where' is a key requirement for emerging applications and services in smart spaces, such as personalized heating and cooling, security management, efficiency monitoring, natural light adjustment, and background music selection [173]. For these and other pervasive services to be truly seamless, tracking and identification need to be performed with high accuracy and without active human effort.

Currently, most identification methods in smart spaces are device-based. Via a carried token, such as ID/swipe cards, active badge, smartphone, or smartwatch, these methods identify users by the unique identifier of their personal devices. However, an implicit assumption made with these techniques is that the user and their identifying device are inseparable, which is not always the case.

In order to cope with more general scenarios, device-free methods have been proposed and are increasingly being adopted for human identification. Vision based techniques (e.g., cameras) are widely used methods in this category and have good performance when given a clear, frontal view of the face. However, cameras are intrusive and have a low user acceptance in domestic and commercial settings [174]. In contrast, radio frequency based methods are less intrusive and have also been utilized for device-free identification. For instance, it has been found that the variations in ambient WiFi signals can be used to recognize people while they walk [164, 175]. Unfortunately, such methods require a separate transmitter and receiver, and are limited to cases when users walk between the transmitter and receiver. The mmWave radar is a transceiver, so only requires a single device for tracking and identification. Point cloud based sensors, such as LiDAR and depth cameras are also able to identify and track people [176]. However, LiDAR is too expensive for home use while depth cameras only have a limited tracking range and accuracy. As optical based sensors, they have similar user acceptance concerns as with conventional cameras.

In this paper, we introduce mID, a system that identifies people by their unique characteristics as sensed by a mmWave radar. mmWave radar provides highly precise ranging by analyzing the reflection from obstacles in the environment, such

as humans. It has a number of interesting properties as we mentioned in Chapter 1. For example, a mmWave radar can be concealed behind furniture, as it is able to penetrate thin layers of different kinds of material [177], unlike optical based sensors. This property makes mmWave radar significantly more unobtrusive by being able to be installed inside furniture or walls. The unobtrusive nature of mmWave radar also means domestic users are more likely to accept it, similarly to how Amazon Echo has been more widely accepted by users than web based cameras [178].

Exploiting these characteristics, we developed our human tracking and identification pipeline based on a commercial-off-the-shelf mmWave Radar. Our device is based on a single chip solution and operates in the $77 - 81$ GHz band. To the best of our knowledge, this is the first work to use the point cloud generated by a mmWave radar to track and identify people while they are walking.

The contributions of this work are as follows:

- We design and implement a human tracking and identification system using mmWave radar, which is capable of providing highly accurate tracking and identification.
- The proposed system is the first to identify people from mmWave radar point clouds, using deep recurrent neural networks.
- We train the deep recurrent neural network with Cross-entropy loss and Center loss, and use Random Forest Classifier for intruder detection.
- We evaluate the tracking and identification ability of our system, demonstrating median tracking accuracy of $0.16m$ and identification accuracy of 89% for 12 people with intruder detection accuracy of 73%. By increasing observation time from 2 s to 7 s and adopt a majority voting method, identification accuracy could rise to 99%.

The roadmap of this chapter is: we introduce the system requirements in Section 3.2; Section 3.3 describes mID, which is our proposed mmWave gait identification system; Section 3.4 introduces the implementation details of mID and the setup

of our testbed; Section 3.5 evaluates the system and Section 3.6 discusses the scalability and robustness, as well as limitations of mID. Related work is given in Section 3.7 and Section 3.8 summarizes this chapter.

3.2 System Requirements

In this work, we set out to design a multi-person tracking and identification system for smart homes, to enable a higher level of autonomy in future installations. The sensing range needs to be up to 5 meters, which could cover a relatively large room. To be minimally intrusive, we require the tracking and identification to be done in an uncooperative way, i.e., users do not need to wear or carry any devices/markers to be tracked or identified.

We require that the system be able to track the human subject while he/she is walking normally inside the monitoring area. Different users could have different walking pace, but typically the system should be able to track the user with $\leq 1.5m/s$ of walking speed. To get a good knowledge of the user's location, the system should be able to track the user with $\leq 20cm$ of error. In addition, as users could enter or exit the monitoring zone, or be temporarily blocked by obstacles, we need the system to be able to handle these situations.

For recognition, we want the system to be able to have a relatively good accuracy ($\geq 85\%$) on a database of around 10 people, based on a short period of observation (2 seconds). Furthermore, we also want the system to be able to recognize intruders, rather than wrongly classifying such sample as one of the users in the database.

To summarize, the requirements of the system design is as follows:

- R1: The radar configuration needs to cover a range of $5m$.
- R2: The system needs to be able to track human subject moving with $\leq 1.5m/s$, and has an error of $\leq 20cm$.
- R3: The system needs to be able to identify the human subject with more than 85% of accuracy from a database of 10 people.

To fulfill the above-mentioned requirements, we introduce our proposed system, mID, as detailed in the next section.

3.3 System Design

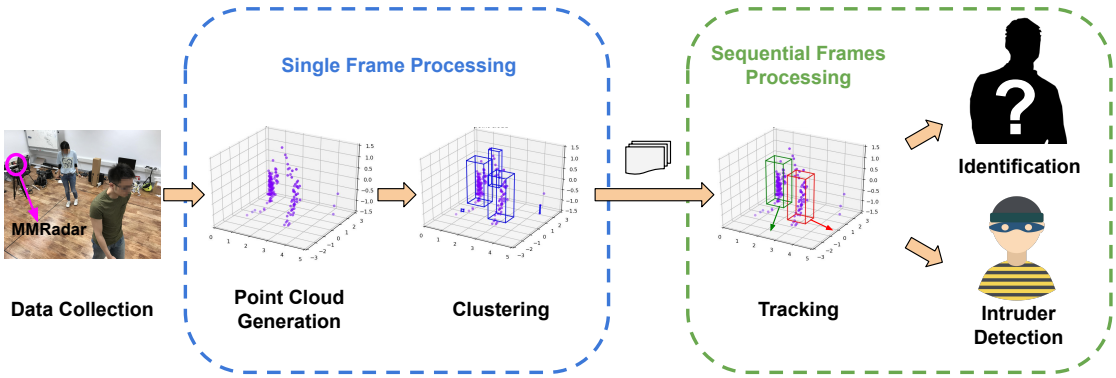


Figure 3.1: System Overview

mID is a tracking and identification system that exploits the unique properties of mmWave radar. It operates by transmitting an RF signal and recording its reflections off objects. By analyzing the generated point cloud, it then infers the people's trajectories and identifies them from a database of known users. The mID system consists of four modules that operate in a pipeline fashion, as shown in Fig. 3.1:

1. *Point Cloud Generation.* In this module, a FMCW radar transmits millimeter waves and records the reflections from the scene. Reflection from clutters is removed after performing range FFT on raw data, and then point clouds are generated with Doppler FFT and Angle of arrival estimation. Because of the sparsity, we combine the point clouds of 3 consecutive frames before further processing.

2. *Point Cloud Clustering.* In this module, potential human objects are detected by merging individual points into clusters with DBScan clustering algorithm.

3. *Tracking.* In this module, mID associates the same human object in consecutive frames and uses a multiple object tracking algorithm to maintain the trajectories.

4. *Identification and Intruder Detection.* In this module, a deep recurrent neural network is used to recognize user identity from the sequential data extracted along the trajectory. The classifier is trained with co-supervision of softmax-loss and center-loss. The distances to the centers of the latent features are used to decide whether the input sample belongs to one of the human subjects in the training set, or it is a sample from an unknown person, with a random forest classifier.

The point cloud generation is done with on-board processing using the methods introduced in Chapter 2. Our contributions in this work involves the remaining modules, which we will provide detailed descriptions in the remaining of this section.

3.3.1 mmWave Radar Point Cloud Generation

In this work we use Texas Instruments IWR1443[1], which features a 3Tx-4Rx antenna array. The antenna arrangement allows both azimuth and elevation angle of arrival estimation, as described in Chapter 2. As a result, the sensor is able to generate point clouds in 3D space.

The radar point cloud generation pipeline follows conventional radar signal processing principles. The point cloud generation in this work is mainly done with on-board processing, consisting of 5 steps: range FFT, clutter removal, Doppler FFT, peak detection, and AoA estimation.

Clutter removal is performed by subtracting a mean for each range bin per antenna across the chirps in a frame. With this step in the processing pipeline, the millimeter wave radar is able to generate a point cloud which does not contain static obstacles. This step is important for two reasons: 1) we are interested in identifying moving people in the scene, removing the static clutter could help us to localize the human subject more easily; 2) static clutter includes the reflections from the environment, and removing such information could make the proposed system work in an environment independent way. However, this does not guarantee that the point cloud does not contain noise from the background. When the human

[1] `http://www.ti.com/tool/IWR1443BOOST`

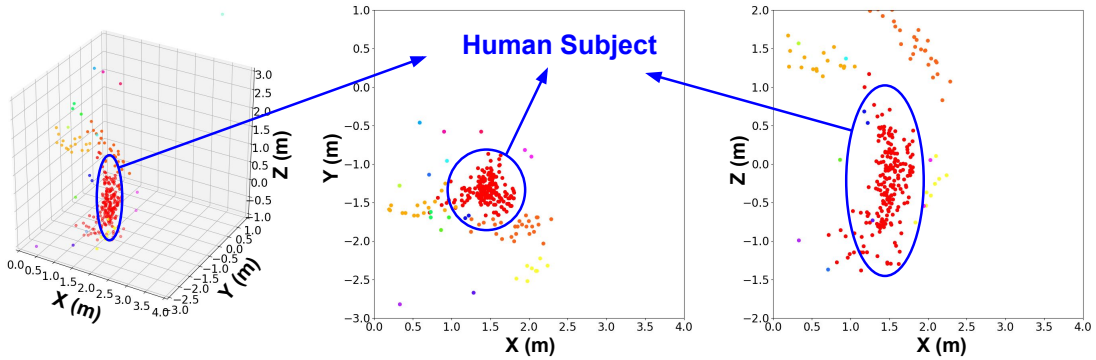


Figure 3.2: 2D Projections of a 3D Point Cloud (see Section 3.3.2). Red points corresponding to a human object are compact in the horizontal x-y plane but dispersed in the vertical z-axis.

subject moves, the previously blocked background would appear again, reflecting the radar signal which results in noise in the point cloud that cannot be fully canceled out through clutter removal.

3.3.2 Point Cloud Clustering

The generated sparse point clouds are dispersed and not informative enough to detect different objects. Moreover, although static objects are discarded through clutter removal, the remaining points are not necessarily all reflected by moving people, but could be spurious noise. As shown in Fig. 3.1, noise points can be significant and lead to confusion with points from nearby people. To determine which points in the scene are caused by reflections from people, mID first merges points into clusters using DBScan [71], a density-aware clustering method that separates cloud points based on their pairwise distance in the 3D space. A major advantage is that it does not require the number of clusters to be specified *a priori*, as in our case people could walk in and fade out of the monitored scene at arbitrary times. Additionally, DBScan can automatically mark outliers to cope with noise.

However, in a real-world measurement study, we observed that points of the same person are coherent in the horizontal (x-y) plane, but more scattered and difficult to merge along the vertical (z) axis. Fig. 3.2 illustrates an example. We hence modify the Euclidean distance to place less weight on the contribution from the vertical

z-axis in clustering. The distance metrics we use in clustering is represented as:

$$D(p^i, p^j) = (p_x^i - p_x^j)^2 + (p_y^i - p_y^j)^2 + \alpha * (p_z^i - p_z^j)^2 \tag{3.1}$$

where p^i and p^j are two different points in the point cloud and the parameter α regulates the contribution of vertical distance.

3.3.3 Moving Object Tracking

To track and identify a person with a sequence of point clouds as input, we require an effective temporal association of detection in each frame as well as correction and prediction of tracks. Algorithm 1 illustrates the flow of our multi-object tracker. We essentially create and maintain tracks for each detected object, which is represented with T. T_i and P_i refers to the tracks and point cloud respectively at frame i, where $T_0 = \emptyset$. A new track is created for each object detection which is either from the first incoming frame or one which cannot be associated with an existing track. Inter-frame object association is based on the Hungarian Algorithm. If a track object is undetected for U continuous frames, we mark the track as inactive and exclude it from successive associations. As the point cloud is not stable, and we lost track of some objects at times because of occlusion or noise, we apply Kalman Filter on the tracks for prediction and correction. These two components are discussed in more detail below. An illustration of track management is shown in Fig. 3.3. Through the algorithm design introduced in this section, we are able to meet the system requirements of smart home user tracking.

3.3.3.1 Detection and Association

We use an effective combinatorial optimization algorithm, the Hungarian Algorithm [72], to create an association between each detected cluster and maintained tracks so that the combined distance loss is minimized. Here we are facing a many-to-many assignment problem where the cost matrix can be non-square because the number of active tracks K_1 and the number of detected clusters at the current timestamp K_2 can be different. Given K as the greater of K_1 and K_2, we essentially augment

Algorithm 1 Human Tracking Algorithm

Input: existing tracks T_i, point cloud P_{i+1}, association distance threshold L, maximum undetected frames U
Output: new tracks T_{i+1}

clusters $C \leftarrow$ DBSCAN(P_{i+1})
$D \leftarrow$ distance matrix for T_i and C
$D_e \leftarrow$ insert dummy entries to D
association $A \leftarrow$ HUNGARIAN(D_e)
$T_{i+1} \leftarrow \emptyset$
for association (t, c) in A **do**
    if t not dummy **then**
        if c not dummy **then**
            if $D_e[t][c] \leq L$ **then**
                t.UPDATETRACK(c)
                $t.missed_frames \leftarrow 0$
                $T_{i+1} \leftarrow T_{i+1} \cup t$
            else
                $t.missed_frames \leftarrow t.missed_frames + 1$
                if $t.missed_frames \leq U$ **then**
                    $T_{i+1} \leftarrow T_{i+1} \cup t$
                end if
                $t_{new} \leftarrow$ CREATETRACK(c)
                $T_{i+1} \leftarrow T_{i+1} \cup t_{new}$
            end if
        else
            $t.missed_frames \leftarrow t.missed_frames + 1$
            if $t.missed_frames \leq U$ **then**
                $T_{i+1} \leftarrow T_{i+1} \cup t$
            end if
        end if
    else
        if c not dummy **then**
            $t_{new} \leftarrow$ CREATETRACK(c)
            $T_{i+1} \leftarrow T_{i+1} \cup t_{new}$
        end if
    end if
end for
for track t in T_{i+1} **do**
    KALMANFILTER(t)
end for

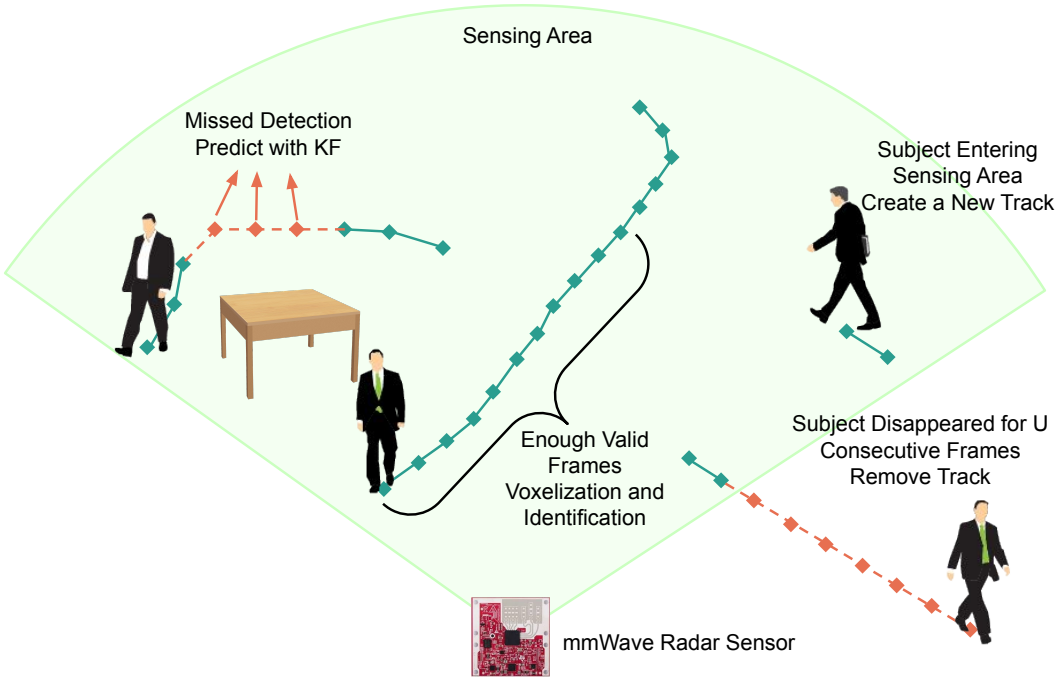


Figure 3.3: mID tracking workflow.

the true cost matrix D with dummy entries to construct a $K \times K$ matrix D_e where $D_e[i][j]$ represents the distance of positions (x,y) between track i and cluster j in current frame. If a cluster is mapped to an augmented track dimension, we create a new track for this detection. Also, if $M_{i,j}$ exceeds step size threshold L we regard the pair as not associated, even if they are associated by the algorithm, given the intuition that the human subject could not move too fast, so track i cannot be associated to cluster j if they are too far away. In this case, cluster j could be a new previously unseen person, or noise, ans we also create a new track for this cluster. Similarly, if a track is mapped to an augmented cluster dimension or a correspondence with cost larger than L, we treat the track object as undetected in the current frame. This method enables us to successfully maintain tracks of objects. If a certain track is undetected for U consecutive frames, we remove this track as we regard this track as noise, which typically lasts a few frames, or an human subject that has left the monitoring area.

3.3.3.2 Track prediction and correction

We use a Kalman Filter[73] to correct for sensor noise and to offer predictive guidance in scenarios where tracked objects are undetected due to occlusion or temporary loss from the sensing region. For each track we maintain a state which consists of location and velocity along the x and y axes. For each track the initial state consists of the first detection location, and the initial velocity is set to 0. At each successive time step, the Kalman Filter predicts the current state variables with the state mean and covariance from the previous frame, as well as transition matrix and covariance, which is known as 'Predict' step. Based on the predicted position and velocity from the transition model, together with the observation at current timestamp, it estimates new state for current frame as well as state covariance, which is known as 'Update' step. The Kalman Filter produces estimates that tend to be more accurate than those based purely on sensor measurement, especially in our case with unstable point clouds and occasional undetected track objects.

3.3.4 User Identification

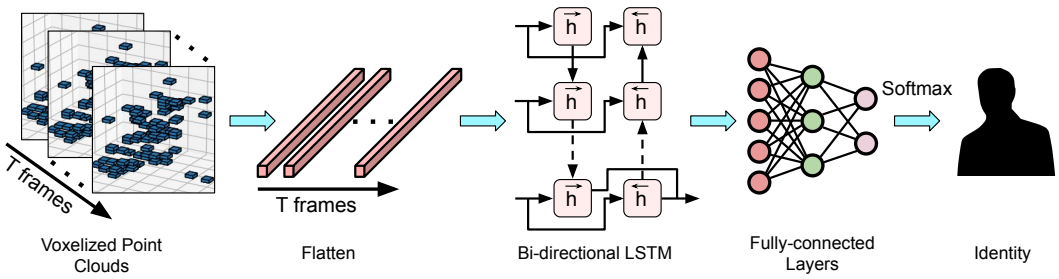


Figure 3.4: Classification Network Structure. Here T represents the number of data frames used for identification.

After the sequence of points corresponding to human objects are determined, we can use tracklets to recognize their identities. Specifically, from each frame in the trajectory, we use a fixed-size bounding box to enclose the points of potential human objects, and voxelize it to form an occupancy grid. Note that the occupancy grids inherently encapsulate body shape information. For instance, tall people tend to have higher center of mass. By feeding the sequential occupancy grids to a

neural network classifier classifier, the ID of a tracklet is recognized based on both movement characteristics, i.e., gait, and body shape information. The tracklet used in mID is segmented with a sliding window method. A window contains consecutive occupancy grids for 2 seconds, with a 75% overlap ratio with the previous window. Extracting useful features directly from the occupancy grids is difficult, as most feature engineering methods are not effective for point cloud classification tasks. The Long-short Term Memory (LSTM) network is an established recurrent neural network architecture suited for sequential data classification which is able to learn the features automatically through network training. We therefore propose using it as the identity classifier in mID. The 3-D data is first flattened and then each frame is converted into a feature vector. This is then passed into a bi-directional LSTM network followed by dense layers. Lastly, softmax activation is applied on the output of the last linear layer to infer the identity (see Fig. 3.4).

3.3.5 Intruder Detection

The output layer with softmax activation tend to be very confident in their predictions, even if it is a wrong prediction. This can also happen when the testing sample is actually not from any of the classes in the training set, i.e., the testing sample is from an out-of-set human subject. As the goal of person identification is to classify into a 1-of-N category, an out-of-set sample, e.g., an intruder, will most likely be simply classified as one of the insiders. In addition, the features learned via the cross-entropy loss tend to scatter across the feature space, and the latent features of a sample from an out-of-set intruder could not be clearly distinguished. To address this issue, we want the latent features from the same class to gather together in the feature space. This is achieved by training the classifier with a center loss in addition to the cross-entropy loss. This has been proven to have better performance and authentication ability with CNN networks for face recognition [179].

The core idea of the center loss is to minimize the intra-class distance of features extracted by the deep neural network from each sample, so that the features of the same class samples tend to gather together in the space [180]. For a sample

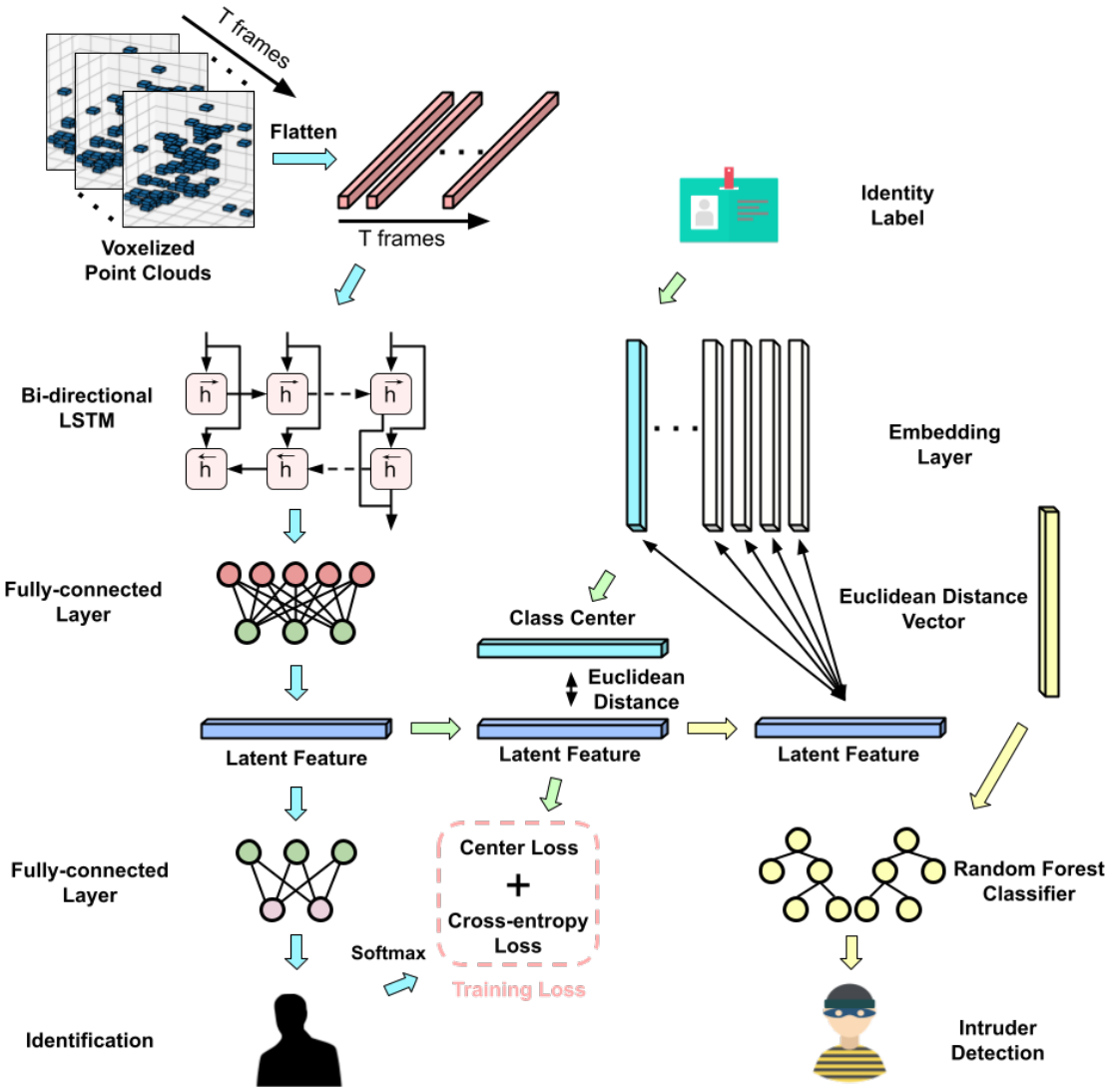


Figure 3.5: Identification and Intruder Detection module workflow. The blue arrow represents the information flow of the human identification part, the green arrow represents the information flow of the center loss part and the yellow arrow represents the information flow of intruder detection part. The network is trained with the combination of cross-entropy loss and center loss.

from an intruder, the features extracted is likely to be relatively further away from the feature centers of the classes in the training set, and thus we can tell that the sample is from an outsider, rather than having the sample wrongly identified as one of the users in the training set.

We define the training loss as the weighted sum of cross-entropy loss and center loss:

$$\mathcal{L} = \mathcal{L}_{cross-entropy} + \lambda \mathcal{L}_{center} \tag{3.2}$$

where we use a parameter λ to weigh the two losses. The center loss is defined as:

$$\mathcal{L}_{center} = \sum_{i=1}^{m} \|\mathbf{f}_i - \mathbf{c}_{y_i}\|_2^2 \tag{3.3}$$

where $\mathbf{f}_i$ represents the i^{th} feature and $\mathbf{c}_{y_i}$ represents the center of i^{th} feature of label y. The center loss is the squared Euclidean distance between the learned features of a sample and the center of the features in the class the sample belongs to.

We implemented this modified network as shown in Fig. 3.5. We use the output of the second last dense layer to represent the latent features. We use an Embedding layer to map the identity label to the center of the features. A custom layer is used to calculate the squared Euclidean distance, and the output of the layer is used as center loss.

Features from intruders have different spatial distributions compared to insiders, which leads to different distances to the feature centers of each class. We use the distances to different class centers as features for intruder detection with a Random Forest Classifier, which has been proven to work very well as a shallow classifier, and outputs a binary prediction of whether the sample comes from an intruder or not.

3.4 Implementation

3.4.1 Testbed Setup

As mentioned before, mID is developed on top of a commercial, off-the-shelf mmWave radar, IWR1443boost. The system is tested in a room with a Vicon optical tracking system, which is able to provide ground-truth position of each marker with an error within $1cm$. Ground truth identities of the training and testing samples are manually labeled. The system consists of two parts, the mmWave radar and the backend. The radar senses the environment and generate 3D point cloud, which is transferred

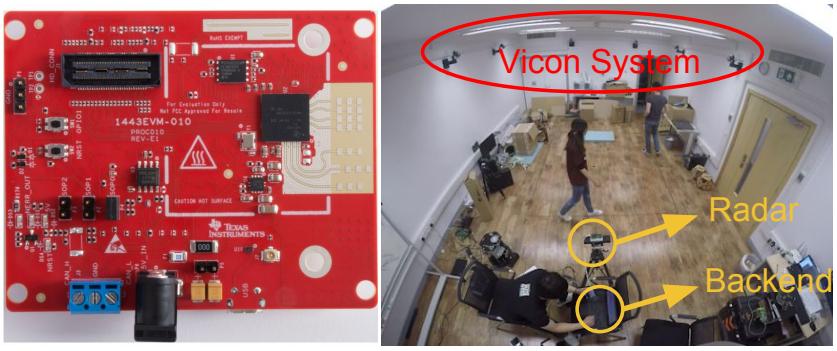


Figure 3.6: Experiment Settings. Vicon system is a high-precision tracking system used for acquiring ground truth trajectories.

to the backend computer for further processing, as shown in Fig. 3.6. We implement the deep neural network classifier using the Keras[2] library with Tensorflow[3] backend.

3.4.2 Dataset Collection

We collect data from 28 participants, with each of them walking randomly in the designated area for 10 minutes. The study has received ethical approval SSD/CUREC1A CS_C1A_18_024. The ages of the participants range from 18 to 35, and 13 of the participants are female. The heights of the participants fall within 155cm to 188cm, and the weights of the participants are from 48kg to 100kg. Data from 12 participants were used for identification evaluation in Section 3.5.3. The body shape of the selected 12 participants used for identification evaluation varies, which mimics domestic settings. The data from the 12 insiders is split into training and testing set with a ratio of 11:1. The data from the other 16 participants were used as intruders.

In intruder detection, data from 12 out of the 16 intruders is used as training data to train the Random Forest classifier, together with the training data from the insiders. Data from the remaining 4 intruders and the testing data from insiders are used as testing data for intruder detection evaluation, because we want the intruder testing data to be from those subjects that are completely unseen

[2] https://keras.io/
[3] https://www.tensorflow.org/

for the Random Forest classifier. This dataset breakdown remains the same for all evaluations in Section 3.5.

3.4.3 mID Configuration

3.4.3.1 Sensor Setup

The IWR1443Boost sensor is configured to use all three transmitter antennas and all four receiver antennas in order to generate 3D point cloud data. Start frequency and end frequency were set to $77GHz$ and $81GHz$ respectively, so the bandwidth B is $4GHz$. The Chirp Cycle Time T_c is set to $162.14\mu s$ and the Frequency Slope S is set to be $70GHz/ms$.

As the proposed system relies on radar point clouds for tracking and identification, ideally we would want the radar to produce denser point clouds, which could contain richer information. However, with a limited number of transmitters and receivers for angle-of-arrival estimation, the TI mmWave radar mainly relies on the Range and Doppler dimensions to resolve targets, so we want the Range and Doppler resolution to be small.

As introduced in Chapter 2, the Range resolution is inversely proportional to chirp bandwidth. As a result, we utilize the full frequency band (4GHz) of the radar, which gives us a Range resolution of around $4cm$. Under this configuration, we could have a maximum unambiguous range of around $5m$, which satisfies our system requirement R1 in terms of tracking area. Although we could track over a longer range by employing a smaller frequency slope, it is not necessary for the proposed system, and as a side effect, this will limit the range resolution.

We want a finer Doppler resolution for a denser point cloud. Due to hardware limitations, there exists a fundamental trade-off between Doppler resolution, maximum Doppler and frame rate. For our configuration, we can have a Doppler resolution of $0.26m/s$ and a maximum unambiguous radial velocity of around $2m/s$, with frame rate reaching $30fps$. This satisfies our system requirement R2. If we require an even finer Doppler resolution, we would need a longer frame time, which will result in lower frame rate.

3.4.3.2 Tracking Implementation

DBScan is implemented with sklearn[4] library. DBScan has two parameters, namely 'Eps' which indicates the maximum distance of two points in the same cluster and 'MinPts' which indicates the minimum point number in a cluster to cope with noise points. α was set to 0.25 in the customized distance function. A smaller distance threshold would split the human point cluster into two, while a larger one might also merge noise into the human point cluster. In practice, we choose 0.05 as the distance threshold and 20 as the minimum points of a cluster. Hungarian algorithm is implemented based on scipy[5] library, with the distance threshold set to $0.1m$. If this threshold is set too large, the human cluster may be wrongly associated to noise clusters; while if the threshold is too small, sometimes the clusters of the same human subject between across frames cannot be associated successfully, as the single frame localization accuracy is limited. In practice, we find that setting the threshold to $0.1m$ works well in our proposed system. Kalman Filter is implemented based on pykalman[6] library.

3.4.3.3 Classifier Training

Through multiple trials we work out a set of parameters that have the best performance. Each frame of the input data is first flattened to a vector of dimension 16000. A bi-directional LSTM with 128 hidden units is used. We set the dropout ratio to 0.5 for the dense layers, because we find that the network is easy to get overfitted. We and used the Adam optimizer with learning rate of 0.0003. We used a balanced dataset where training/test sample ratio is set to 11:1. To decrease overfitting, we further augment the training data to 8 times the original size, by shifting the data in X and Y axis respectively for 1 voxel, and rotating each frame by 90°, 180° and 270°. The model is trained for 30 epochs. Without or with less data augmentation, the neural network may get overfitted to the training data, as the dataset only contains around 10 minutes of training data for each person.

[4] `https://scikit-learn.org/stable/`
[5] `https://www.scipy.org/`
[6] `https://github.com/pykalman/pykalman`

Based on extensive experiments, we find in practice, a tensor of length 64 is the best size for the dense feature layer. If the feature layer is too large, the model can easily become overfitted, while if the feature layer is too small, it lacks representation ability which will also lead to performance degradation. We set the weight of cross-entropy loss and center loss to 1:0.5, which has the best performance in our case.

The Random Forest classifier is implemented based on sklearn library. The parameters of Random Forest classifier are decided by grid search with 5-fold cross validation.

3.5 Evaluation

3.5.1 Sensitivity Analysis

3.5.1.1 Non-Line-of-Sight Conditions

Material	Point Cloud Density Decrease (%)
Foam	0.02
Plastic	0.12
Wood	0.25

Table 3.1: Impact of different materials on point cloud density of mmWave radar. The differences in point cloud density are below 1% for all cases.

In the first experiment, we study the robustness of mmWave radar under occluded conditions. This property is important because optical imaging based tracking and identification methods, such as RGB and depth cameras, can only work under line-of-sight conditions. We evaluate the robustness of millimeter wave radar with three types of obstructions: foam, plastic and wood. We used a sheet of each material in turn with a thickness of approximately $3mm$ and a size of around $30cm \times 30cm$. The obstacles are placed $1cm$ away from the sensor so that the signal cannot be transmitted in a line-of-sight condition. We let a user walk back and forth in front of the millimeter wave radar while collecting sensor readings. mID uses the generated 3D point cloud for tracking and identification, so we compare the percentage of change in point cloud density for mID. The testing result is summarized in Table 3.1. This shows that mmWave is very robust against non-line-of-sight interference, with

less than 1% change in point-cloud density, which means that there is almost no impact on the radar sensing ability. Robustness to thin obstructions could enable mID to work under furniture or concealed within a picture frame, etc., making it psychologically less obtrusive, which could yield higher user acceptance.

3.5.1.2 Impact of weighing the vertical axis in DBScan

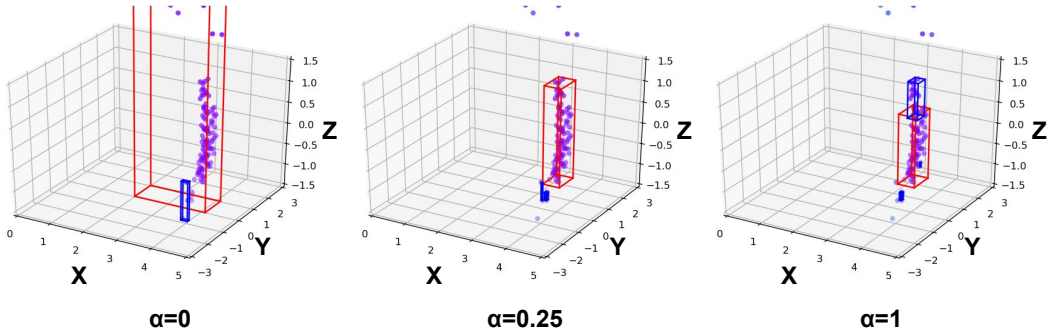


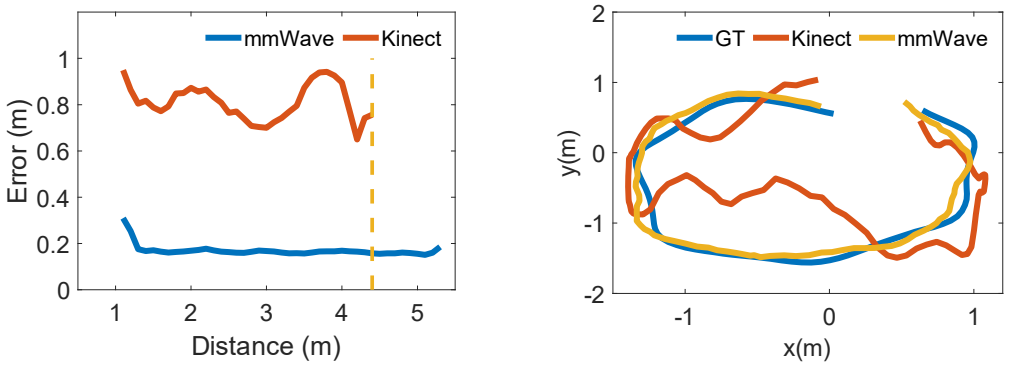
Figure 3.7: Clustering results with different α. A small α leads to loose clusters containing many noisy points. A large α splits a human object into two clusters. Setting $\alpha = 0.25$ gives the best empirical performance, as shown in the middle one.

As introduced in Section 3, it is important to define the weighting parameter to make the DBScan algorithm work better. Therefore, we need to find a suitable value of α to obtain a good clustering result. In practice, we found that $\alpha = 0.25$ results in good clustering performance, as shown in Fig. 3.7. In contrast, when $\alpha = 0$, (i.e., points are effectively projected onto the $x - y$ plane), outliers are merged into the cluster. When $\alpha = 1$, (standard Euclidean distance) points corresponding to a person are split into two clusters.

3.5.2 User Tracking

To evaluate the tracking accuracy of mmWave radar, we compared it to a Kinect v2, an RGB-D camera which is widely used in homes for gaming. We installed the mmWave radar and Kinect v2 co-located in a room. The Vicon system was set to track a marker placed on the top of a hat worn by the participant. Time was synchronized through a NTP server and the coordinate transformation matrices of the systems were calibrated. A state-of-the-art open source tracking algorithm[7]

[7]based on `https://github.com/mcgi5sr2/kinect2_tracker`



(a) Tracking Error **(b)** Inferred Trajectory

Figure 3.8: Comparison of tracking performance between mmWave radar and Kinect v2. (a) Kinect v2 can only track people within $4.5m$. (b) Inferred trajectory. (GT: Ground Truth by Vicon)

is used for Kinect v2 people tracking. The tracking error and a comparison of the trajectories is shown in Fig. 3.8, which shows the median tracking error of mID is $\sim 0.16m$ whereas the Kinect is $0.9m$. Besides a significantly smaller error compared to Kinect v2, the tracking range of mID is also larger than Kinect v2. mID can track people at a distance of more than $5.5m$, whereas Kinect has a tracking limitation of $4.5m$. This could be further extended, possibly at the cost of a reduction in identification accuracy. This shows that the radar based technique is able to achieve highly accurate tracking over a larger tracking area, satisfying our system requirement R2 and making it suitable for increasing levels of automation.

3.5.3 Identification Evaluation

In this part we evaluate the identification accuracy of our proposed system. The recognition accuracy is based on single-person scenario, where we use our proposed neural network classifier introduced in Section 3.3 with parameters specified in Section 3.4. We also provide an ablation study of the network size and structure.

3.5.3.1 Identification Performance

Overall, mID is able to reach an accuracy of 89% for 12 people, which satisfies our system requirement R3. The confusion matrix is shown in Fig. 3.9. Note that this

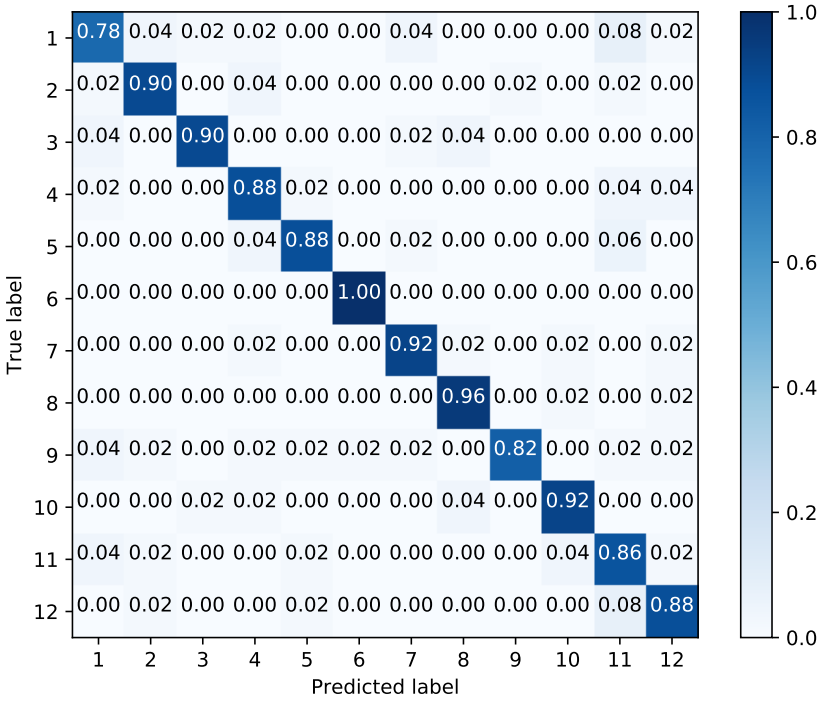


Figure 3.9: Confusion Matrix of 12 Users.

performance is non-trivial considering that the original point clouds are very sparse, and demonstrates the neural network's capability for feature extraction.

From the confusion matrix we can also see that, the identification accuracy varies across different human subjects. For example, subject 6 has almost both 100% for precision and recall, while subject 1 only has an accuracy of 0.78, with 22% of the samples wrongly identified as other subjects. This is because some people are inherently easier to be identified as they have significantly different height, body shape or walking patterns compared to other human subjects in the dataset.

3.5.3.2 Impact of number of people

In this experiment, we further explore identification accuracy with varying group sizes. Intuitively, a smaller group size should make the problem easier as there are fewer candidates to choose from. Fig. 3.10 shows the performance trend of mID when varying the number of participants from 4 to 12 with a step of 2. As we can see, mID works quite well in the cases where there are no more than 6 people in the dataset, with 95% accuracy. According to U.S. Census Bureau, the average

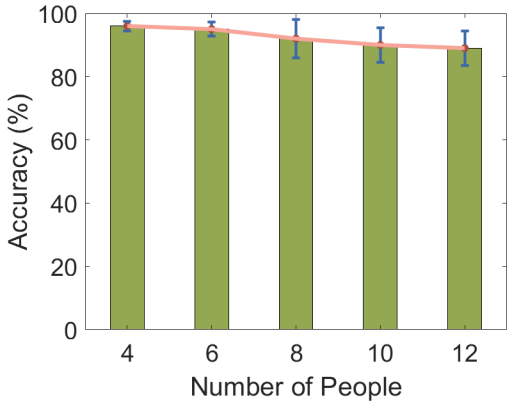
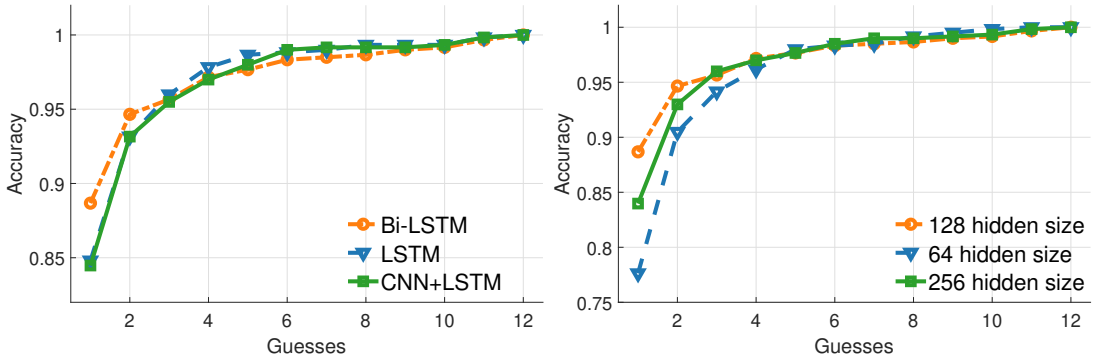


Figure 3.10: Impact of number of people. mID works reliably with different numbers of participants and performs best when there are no more than 6 people in the dataset, e.g., in home scenarios.

number of people in a household in the United States is 2.58 [181]. As a result, the number of people in domestic settings is generally less than 6, and we envision that mID could greatly benefit many applications in smart homes of the future.

3.5.3.3 Neural Network Ablation Study



(a) Different network architectures

(b) Different bi-directional LSTM sizes (per direction)

Figure 3.11: Ablation study of neural network architecture and size.

Neural Network Architecture Comparison: To explore the best neural network architecture for the identification problem, we compare 3 different architectures that are based on LSTM. We use variants of LSTM as it has been shown to be performant for end-to-end learning of time sequence data. Each model is trained under the same settings as described in Section 3.4. The LSTM layer used in all

the 3 models has the same hidden size of 128. As a result, for bi-directional LSTM we use in this work, there are 256 hidden units in total. The CNN used in the CNN + bi-directional LSTM model has two convolution layers, with a max pooling layer after each convolution layer. The CNN is time distributed, which means the data of each frame is first sent into a two-layer 3D CNN for feature extraction, then the flattened sequence data is sent into bi-directional LSTM for classification.

The accumulative identification accuracy of the different architectures is shown in Fig. 3.11 (a). The accuracy differences between architectures become negligible after 3 guesses. However, bi-directional LSTM significantly outperforms the other two architectures within fewer guesses, and we find that it is able to converge in less epochs compared to single direction LSTM. This is presumably because bi-directional LSTM is able to model the rich temporal correlations in a long sequence of frames from both ends. Such information gain improves the identity inference performance. It is straight forward to consider adding CNN for feature extraction before LSTM, since the input occupancy grid is image-like type of data. However, in practice we find that with CNN added, the model gets more overfitted and thus the identification accuracy actually downgrades. However, while we use the flattened occupancy grid directly as features in our case for better performance in our case, we believe that CNN+LSTM is still an architecture worth considering.

Impact of Network Size: Beyond architecture, it is also important to consider the impact of the network size. If the network is too small, it may lack representation ability, while networks that are too large can suffer from over-fitting problems. We evaluated the Bi-directional LSTM network with 3 different sizes and the comparison of the classification performance is shown in Fig. 3.11 (b). Results show that the identification accuracy is not very robust to the size of the LSTM layer. The Bi-directional LSTM with 128 hidden units per direction significantly outperforms the same network architecture with a larger or smaller size, especially in the first two guesses. This suggests that a Bi-directional LSTM with size of 128 hidden units per direction is a good fit under the current configuration.

3.5.3.4 Impact of Observation Time

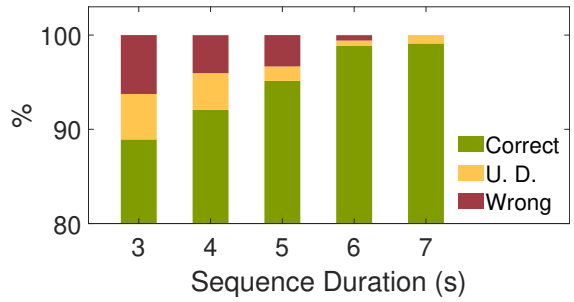


Figure 3.12: Impact of Observation Time on identification performance. The total observation period is cut into testing samples and a majority voting method is adopted. The green color represents percentage of correct prediction, and the red color represents wrong prediction. Yellow represents Unable to Decide (U.D.), which happens when the top two predictions get the same number of votes.

Intuitively, the longer that the sensor senses the walking pattern of a person, the more chance that the person could be identified correctly. Given a period of observations, we split it into testing samples using sliding window with length of 2 seconds and 75% overlap between consecutive samples, the same way as we did in the data preparation for identification evaluation. Given an observation duration of k seconds, the total number of testing sequences n would be equal to:

$$n = \frac{k - t_{sample}}{0.25 \times t_{sample}} + 1 \tag{3.4}$$

where t_{sample} equals 2 seconds in our case. The final prediction for the longer sequence is simply decided with a majority voting policy on the prediction result of the testing samples.

We evaluated the identification accuracy against different sequence duration with the same dataset that contains 12 human subjects. The results are shown in Fig. 3.12. The percentage of correct prediction reaches 95% when the observation time is longer than 5 seconds, and further reaches around 99% at 6 seconds. Results show that with longer observation time, mID can have a excellent identification performance, even with more than 10 people in the dataset. With the system deployed at large, open spaces, we are likely to have more chance to get longer observation periods, which could benefit the identification accuracy.

3.5.4 Intruder Detection Evaluation

Insiders	Precision	Recall	F1	Accuracy
4	0.83	0.95	0.89	0.87
6	0.77	0.88	0.82	0.81
8	0.75	0.88	0.81	0.80
10	0.73	0.88	0.80	0.78
12	0.79	0.70	0.74	0.73

Table 3.2: Precision, Recall, F1 score and Accuracy of Intruder Detection. Here the hypothesis is the person is an intruder. Precision is the ratio of real intruders to all samples classified as intruders. Recall is the percentage of intruders identified. Accuracy is the chance of correctly classifying a sample.

In this section, we examine the performance of the intruder detection classifier. We collected additional data from 16 people, who are not involved in the classifier training process. This 16 people act as out-of-set intruders. We trained the classifier network using center loss and cross-entropy loss to force feature distances of the same class to be small, with the data from users in the insider dataset. We then used 12 out of the 16 intruders, together with the training data of the insiders, to train the binary random forest classifier. The data from the remaining 4 intruders and the testing data from insiders are used as testing data, with a ratio of 1:1. We balanced both the training set and the testing set, i.e., using same number of samples from insiders and intruders, for a better understanding of the intruder detection ability of the proposed model.

Results are shown in Table 3.2. We can see that when there are fewer users in the dataset, the performance of intruder detection model increases greatly. We find that intruder samples scatter widely across the high dimensional feature space, leading to a higher chance of confusion when the number of users (classes) is high. From the results above, we can see that for intruder detection, mID also works best in places where the number of users is limited, such as smart homes. Note that with access to a larger training set of intruders, e.g., hundreds of users, it could be possible to improve these ability of intruder detection further.

3.6 Discussion on Scalability and Robustness

Despite the fact that our proposed system works well for tracking and identification, we elaborate the limitations in this section, and provide a discussion on scalability and robustness.

Scalability

1. *Large Number of Non-concurrent Users.* We have demonstrated the reliable performance of mID when working with a relatively small group of users (12). However, tracking and identifying a large number of non-concurrent people remains an open problem for two reasons. First, the point cloud generated by mmWave is sparse, and sometimes the sparsity could significantly disturb human detection and tracking. Secondly, body shape and human gaits used in this work are weak biometrics, and could become ambiguous with the increasing number of users in the database.

2. *Large Number of Concurrent Users.* When multiple people appear in the scene at the same time, the tracking and identification ability of the system are both affected. Firstly, The IWR1443Boost we use in this work features a 3 transmitter 4 receiver MIMO antenna array. The best azimuth angular resolution of mID processing pipeline based on this device is 15°, so if two people are too close together, it can be hard for them to be separated in the generated point cloud. Also, due to hardware memory and SNR requirements, the total points of each frame is limited. Multiple people in the scene could share the point cloud, making it even more sparse, and extremely hard to identify people with the model trained on the dataset of single person walking. This work mainly aims to prove the feasibility of identifying people with mmWave radar, and identifying multiple people at the same time still requires further exploration.

3. *Large Buildings* mID uses millimeter wave radar, and the wavelength of the signal cannot penetrate thick walls. As a result, when using the system in

large buildings with many rooms, we need devices deployed in each room, and make sure that there's no interference between the deployed sensors.

Robustness

1. *Monitoring Range.* In our experiment setting, we set the maximum unambiguous range of mID to around $5m$, which is roughly 3/4 the size of monitored room (see Section 6.8). Although theoretically the range of the mmWave radar can be more than $20m$, this increased range comes at the cost of reduced spatial resolution and diminished signal-to-noise ratio (SNR), and the identification accuracy is very likely to get worse.

2. *Flat and Planar Surfaces.* As we found in the experiments, the reflection profile of mmWave can be affected by flat and planar surfaces, such as windows or mirrors, due to strong multi-path effect. As a result, noisy 'mirror' human objects appear occasionally when encountering these surfaces. Our experimental site is mainly with walls, which are typically not strong reflectors due to their rough surfaces. However, it is worthwhile to consider the impact of disturbing surfaces in real world deployment, such as large glass walls and windows.

3. *Clothes and Injuries* Clothes made of different materials could have different reflection properties. And whether clothes can affect the identification accuracy remains an open question currently. Injuries could affect the identification accuracy as well, for it is likely to change the walking pattern of the human subject.

4. *Room Clutter and Obstacles* As mentioned in Section 3.3, clutter removal is applied when generating the point cloud, so the reflection from static objects like room clutters would mostly be canceled out in the point cloud and does not affect the system. However, there may be obstacles in the room like furniture that can block the signal and make the system temporarily lost tracking of the subject. In this work, we use a Kalman Filter to provide a location

prediction of the missed frames. There are many advanced techniques and algorithms for robust multi-object tracking that are also worth trying to solve this problem, including Multi-hypothesis Tracking [182], Joint Probabilistic Data Association [74], Maximum Joint Likelihood Tracking and Association [183], etc.

3.7 Related Work

3.7.1 Device-Free Gait Recognition

As device-based methods always suffer from 'forget to carry' and 'forget to charge' problems, there has been a lot of works on device-free human identification. Middleton et al. built a floor sensor based gait recognition system and achieved 80% accuracy over 15 people [184], at high deployment cost. Vision-based methods are one of the most established techniques. Besides face recognition, they can work on gait recognition as well. Some works use silhouette analysis approaches like [185], others use model-based methods like in [186]. Besides monocular vision, multiple cameras, stereo cameras and depth cameras are also utilized in gait recognition tasks [187]. However, as long as video data is used, even without facial images, there will always be a risk that users' privacy would be compromised if data leakage occurs.

Some researchers tend to RF sensing for device-free human identification solutions. It has been shown that the Channel State Information (CSI) of WiFi signals, for both $2.4GHz$ and $5GHz$ bands, captures human gait information to a certain extent [163, 175]. These gait recognition systems are relatively easy to deploy compared to some specific sensors such as floor sensor, as WiFi devices are commonly seen in daily lives. However, such methods are generally scene-dependent and cannot handle environmental dynamics very well. Also the WiFi CSI information does not capture body shape and size information very well, which loses some very useful information for human identification.

A full comparison of different device-free identification methods are provided in Table 3.3.

	Floor Sensor	RGB	Depth	WiFi CSI	mmWave
ID Accuracy	Moderate	Very high	High	Moderate	Moderate
Tracking	Yes	Yes	Yes	Yes	Yes
Environment Independence	No	Yes	Yes	No	Yes
Privacy Concerns	None	High	Medium	Low	Low
Ease of Deployment	Very Difficult	Easy	Easy	Difficult	Moderate

Table 3.3: Comparison of different device-free identification methods and their relative merits.

3.7.2 Other RF-based Human-centric Applications

RF sensing has been widely used in human-centric applications, such as fall detection [188], occupancy counting [189], breathing monitoring [190], etc. Google has recently published a mid-air gesture interface based on mmWave radar, which recognizes hand gesture in high fidelity [30]. Zhao et al. have recently proposed a human pose reconstruction approach with a customized RF radar. By using the pose extracted from the collocated cameras as supervision signals, their reconstruction network could learn to estimate human skeleton, in both 2D and 3D scenes [85, 191]. Please refer back to Chapter 2 for a detailed review of other mmWave radar based human-centric applications.

3.8 Chapter Summary

In this chapter, we propose mID, an accurate human tracking and identification system for smart spaces based on a commercial-off-the-shelf millimeter wave FMCW radar. We first obtain sparse point clouds, and then we extract the point clouds representing human objects and associate them to their historical trajectories. Based on the tracklets, a recurrent neural network is used to recognize their identities, and we trained the network with co-supervision of cross-entropy loss and center loss followed by a Random Forest classifier for intruder detection. Experimental results show that mID achieves an overall recognition accuracy of 89%, and intruder detection accuracy of 73% among 12 people, which increases when there are fewer people in the dataset. mID also bears approximately $0.16m$ positioning error. We envision mID as a promising step towards smart home human identification and

tracking, for the sparse point clouds adopted in mID are not as privacy sensitive as vision based techniques. Besides, mID can be concealed inside furniture or walls thanks to the penetration abilities of mmWave, which can be highly psychologically unobtrusive and gain acceptance by smart home users.

This chapter proves the possibility of target recognition from the sparse mmWave radar point cloud with the help of deep neural networks, addressing the first research question we raised in Chapter 1. To the best of our knowledge, this is the first exploration of using mmWave radar point cloud for people identification, which is followed by later works focusing on simultaneously recognizing multiple human subjects [90, 91] and on using similar pipeline for activity recognition [192].

4

Sensing Micro Displacements

Contents

4.1 Introduction 79
4.2 Related Work 82
4.2.1 Electrocardiogram-based Techniques 82
4.2.2 Vision-based Techniques 82
4.2.3 RF-based Techniques 83
4.3 System Requirements 84
4.4 Proposed System 85
4.4.1 User Tracking Module 85
4.4.2 mmWave Servoing Module 86
4.4.3 Heart Rate Estimation Module 88
4.5 Implementation 91
4.5.1 Data Collection 91
4.5.2 Network Training 92
4.6 Evaluation 93
4.6.1 Setup 93
4.6.2 Performance Comparison 94
4.6.3 Uncertainty Evaluation 96
4.6.4 Impact of the mmWave Servoing Module 98
4.7 Discussion 98
4.8 Chapter Summary 99

In this chapter we look at research question 2, which is to robustly extract micro-displacement patterns and estimate the prediction uncertainty. We propose 'mBeats': sensing human heart rate with a robot mounted mmWave radar. Vital sign

monitoring is one of the key applications for mmWave radar in smart environment, and this work mainly aims to explore monitoring the user's heart rate in an unrestricted way, i.e., no active user cooperation is required. mmWave radar heart rate sensing accuracy is impacted by the distance and orientation of the human subject, so we mount the mmWave radar on a moving platform for the optimal sensing position and angle. We use deep neural networks to extract the heart rate information from the phase variation signal, which demonstrates significant improvement compared to conventional methods. Furthermore, our proposed system can also predict the uncertainty of the estimation, which is very important for health care applications.

Portions of this work is included in the paper:

- **Peijun Zhao**, Chris Xiaoxuan Lu, Bing Wang, Changhao Chen, Linhai Xie, Mengyu Wang, Niki Trigoni, Andrew Markham. "Heart Rate Sensing with a Robot Mounted mmMave Radar." In *2020 IEEE International Conference on Robotics and Automation (ICRA'2020).*

4.1 Introduction

Heart rate is a key indicator for assessing health, stress and fitness. In particular, periodic (e.g., hourly) monitoring of elderly patients, those in palliative care, or patients undergoing post-operative rehabilitation provides a quantifiable metric of whether intervention from a health professional is required or not [193]. This is becoming more and more important as there is an increasing shift towards caring for patients within their homes, rather than in hospitals, as it is more cost-effective [194].

Significant strides have been made in ambulatory heart rate monitoring, e.g., with wearable devices operating either with electrocardiogram (ECG) or photoplethysmography (PPG) sensors. In particular, smart-watches and fitness bands provide a 'wear and forget' capability and have made continuous heart rate monitoring inexpensive and wide-spread. Although wearable devices are an excellent solution for the general public, for the aforementioned problems they suffer from 'forget

to wear' and 'forget to charge' problems. This leads to low compliance, making them unsuitable for monitoring patients with physical or mental health issues. Alternatives include sensor-equipped beds [195] but these are limited to monitoring patients in a single location.

These limitations have lead to the development of non-contact heart rate monitoring solutions, including those based on imaging and radio-frequency (RF) techniques. With respect to the former, a camera (either in the visible or infrared spectrum) is used to detect subtle variations in blood vessel dilation in the face [196]. These methods require a direct line-of-sight to the patient, and do not work with occlusions, e.g., clothing. They also raise issues about privacy due to the use of cameras. Besides, camera based methods are very sensitive to lighting conditions. RF based techniques operate by inferring the micro-displacement of the heart through subtle changes in the reflected radio signal. Most work to date has considered using a static device, e.g., placed on a wall, to obtain these measurements [78, 190]. However, these types of sensors are very sensitive to the orientation and position of the target user [197].

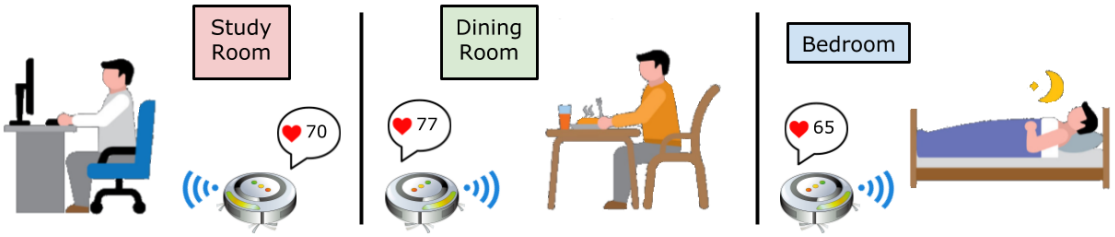


Figure 4.1: mBeats is a robot mounted mmWave radar system. It is able to provide periodic heart rate measurements of a user, without interfering in a user's daily activities, and without being constrained to only operate at a certain location.

An ideal system, as shown in Fig. 4.1, would be able to provide periodic heart rate measurements of a user, without requiring anything to be worn, and without being constrained to only operate at a certain location. Motivated by the increasing adoption of domestic service robots, e.g., robotic vacuum cleaners, we posit a scenario where these could serve a dual purpose as mobile heart rate scanners. We in particular exploit recent developments in the miniaturization of single-chip, low-cost mmWave radar which is capable of accurately measuring micron-level

displacements. We believe it is quite likely that low-cost robotic platforms of the future will be equipped with mmWave radar for obstacle avoidance [198]. In this way, a domestic service robot could periodically scan a patient's heart rate before continuing with its normal operation.

In this paper, we present mBeats, a novel approach towards providing non-contact heart rate measurements throughout the home, without requiring any fixed infrastructure or compliance with charging and wearing a device. A particular challenge is that service robots are typically low in height and close to the floor and thus unable to sense the chest displacement due to heart beat. Instead, we demonstrate that it is possible, by using mmWave radar, to measure the heart rate in a user's lower leg, operating through clothing. This is a difficult signal processing challenge as the radar return is weaker, and conventional methods often produce sub-optimal heart rate estimation results. To solve it we rely on advances in deep learning to extract accurate heart rate measurements. Unlike competing approaches, not only do we provide a measurement, we also provide a metric of uncertainty. This is critical for providing trustworthy measurements without creating false alarms, or worse, not reflecting an emergency condition.

The main contributions of this work are:

- To the best of our knowledge, this is the first work to implement robot based heart rate measurement with mmWave radar.
- We propose a feedback control approach to actively steer the mmWave radar for optimal signal detection.
- We use a deep neural network to process the radar displacement measurements to accurately measure the heart rate in the lower leg. This network can provide estimation uncertainty positively correlated to prediction errors.
- We release the implementation of mBeats to the community, including code and datasets.

The remaining sections of this chapter are organized as follows: We give an introduction of related works in Section 4.2. We define the system requirements in Section 4.3. Our proposed system is introduced in Section 4.4 with implementation details in Section 4.5. We demonstrate the experiment settings, results and discussions in Section 4.6. Potential future works are discussed in Section 4.7 and Section 4.8 finally concludes the chapter.

4.2 Related Work

Contactless Heart Rate measuring has been studied a lot in recent decade. Mainstream methods include three categories: Electrocardiogram-based, Vision-based and RF-based. In this section, a brief review and discussion of these methods is provided.

4.2.1 Electrocardiogram-based Techniques

Non-contact Electrocardiogram measuring uses capacitive electrodes instead of the conventional adhesive electrodes, and thus do not have be in contact with the user's skin directly. These kind of sensors have been embedded in a range of different objects, e.g., bed [199], wheelchair [200], driver's seats [201], etc. Although far less intrusive than conventional electrode-based ECG, it requires the user to be very close (a few cm) to the sensor, limiting its applicability.

4.2.2 Vision-based Techniques

Vision-based Heart Rate Measuring has been widely researched. In most of these methods, a Region of Interest is first detected and tracked, over various parts of the body [196, 202, 203]. As the heart beats, blood flow causes subtle color changes on human skin and this information can be captured with an RGB camera. Infrared cameras have also been used for heart rate detection [202, 204, 205], and can even extract heart beat information from pupillary fluctuations [206]. Vision-based techniques can work well for tasks like sleep monitoring [207], or tele-medicine with a webcam [208], and puts least burden on the user. However, they can only work

under line-of-sight conditions and raise privacy issues. RGB based systems also are typically restricted to well illuminated conditions.

4.2.3 RF-based Techniques

RF-based heart rate measuring techniques are primarily based on Radar and WiFi. Various types of radar like UWB Impulse Doppler Radar [209], Continuous Wave Radar [210], and Frequency-Modulated Continuous Wave Radar [190], with different operating frequencies and output powers have been used for this application. These techniques measure the micro-displacement of a user's skin. Radar signals can penetrate many kinds of materials so it is possible to perform non-line-of-sight measuring of vital signs. However, signal strength impacts the operation range, with consequent trade-off between transmitter power and measurement distance [211].

WiFi-based techniques are widely used for RF-based heart rate measuring, as most of these techniques can be implemented with commercial-off-the-shelf WiFi devices, like routers and laptops. Micro-displacement of the user's skin is estimated by capturing the channel state information (CSI) [165, 212]. However, the accuracy of WiFi-based techniques is greatly affected by the user's location and body orientation [197].

The aforementioned techniques are mainly based on conventional signal processing algorithms, like peak detection, Fast-Fourier Transform, Auto-correlation, etc. These algorithms have long been used to extract periodic information from time series signals.

Our approach broadly falls into the category of RF based monitoring, however, it has three prominent advantages over other methods. The first is that by mounting the system on a low-height mobile robot, our approach is location independent. Our system works in a 'best-effort' way, taking measurements while the user is sitting or lying at home, and does not require active user cooperation. Secondly, as our experimental results show, compared to the conventional methods which are based on signal processing algorithms and heuristics, our system reaches the highest accuracy. Lastly, our system is able to provide the estimated uncertainty

in addition to heart rate. By setting an uncertainty threshold we are able to keep only those heart rate estimations that we are confident about, which is very important for health care applications.

4.3 System Requirements

As introduced in Section 4.1, the proposed system targets daily heart rate monitoring in smart homes. To avoid the drawbacks of wearable devices (forget to wear and forget to charge) and cooperative measuring (causing burden to users), this system aims to realize daily heart rate measuring without active user cooperation. As users can move around the smart space, we propose using a mobile platform (e.g., a cleaning robot) to sense the skin displacement from the user's lower leg skin while the user is sitting or lying, as typical domestic mobile robots are short and close to the ground.

As the skin displacement at lower leg due to heart beat is much smaller than the chest skin displacement, the proposed system needs to be able to measure displacements of down to 50 microns. To achieve clinical validity, the proposed system needs to estimate the heart rate with an average error of $\leq 10\%$.

As the system works in a best effort way, erroneous heart rate estimation could happen due to noise or user movement - thus we also need the system to have the ability of rejecting invalid measurements. To conclude, the requirements of the proposed system are:

- R1: The proposed system should be able to measure the user's heart rate with accuracy of over 90%.

- R2: The proposed system should be able to estimate the uncertainty of the heart rate measurement.

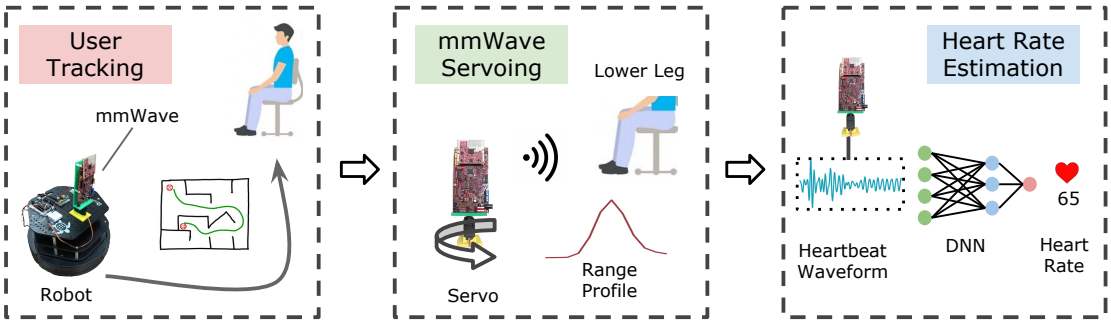


Figure 4.2: mBeats consists of three modules operating in a pipelined manner. (i) *user tracking module* actively tracks a target with the mmWave radar and instructs the mobile robot to move towards the user's proximity; (ii) *mmWave servoing* module adaptively rotates the mmWave radar that optimises the sensor angle for best heart rate observation; and (iii) *heart rate estimation* module senses the micro displacement of user's skin and estimates his/her heart rate with an uncertainty.

4.4 Proposed System

mBeats comprises of a robotic platform equipped with a mechanically steerable mmWave radar module. Without loss of generality, in our work we use a TI IWR6843 single chip mmWave radar, which operates by sending out a FMCW chirp over a 4 GHz bandwidth, starting from 60 GHz. By measuring the time for the signal to return to a co-located antenna and correlating with the transmitted signal, it is possible to measure the range to reflective objects with high accuracy (cm level), as we have introduced in Chapter 2. By computing the phase difference between successive scans, it is then possible to measure relative displacements with near micron-level accuracy [213].

The three modules in our active sensing and signal processing pipeline are shown in Fig. 4.2, including (i) user tracking, (ii) mmWave servoing and (iii) heart rate and uncertainty estimation.

4.4.1 User Tracking Module

The user tracking module is largely based on Chapter 3 that utilizes the mmWave point cloud to pinpoint the user in the field of view. Given the estimated user's location, a robot then approaches the user at a certain measurement distance (0.5m in this work), and performs subsequent sensing actions. A smaller distance can

increase the target signal strength, but it may disturb the user, which violates the system requirement. On the other hand, if the distance is too large, the signal-to-noise ratio might be too low to extract meaningful information.

As this module is not our primary focus here, we assume in the remainder of the chapter that the robot can localize and approach the target user.

4.4.2 mmWave Servoing Module

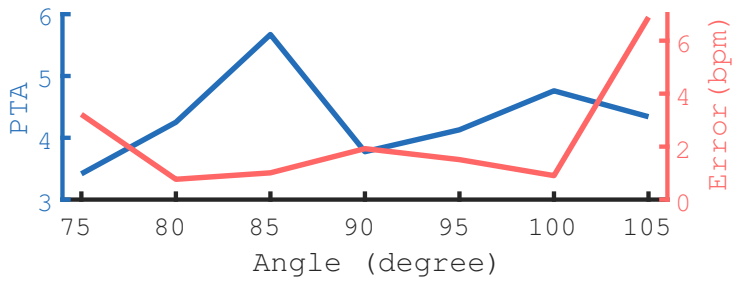


Figure 4.3: Impact of observation angles on heart rate estimation. We place the robot approximately 0.5m away from a user's lower leg and rotate the sensor from 75°to 105°at a stride of 5°and each angle for approximately 40 seconds. The error is obtained from a fixed estimation method [214]. PTA: Peak-to-average ratio.

Unlike a planar surface, directly facing the user does not guarantee the best reflection signal, as the tibula and fibula bones in the calf block the signal. Furthermore, due to multi-path effects caused by surrounding objects, a small deviation from the optimal direction may lead to a significantly degraded performance. As such, given a good measurement distance, the heart rate estimation accuracy of a mmWave radar still largely depends on observation angles (see Fig. 4.3). Generally speaking, we have a more accurate heart rate estimation with higher Peak-to-average (PTA) ratio, because of the better signal quality.

The goal of this module is therefore to optimise the sensor angle for peak signal reflection from the patient's lower leg/calf, i.e., we are trying to rotate the sensor for an optimal PTA in the Range Profile. Although it would be possible to rotate the robot itself, depending on the capabilities of current service robots, this is cumbersome and would be imprecise to reach the optimal angle. Hence we control the orientation of sensor with a *servo motor* directly according to the characteristics of the reflected mmWave signal.

This is considered an optimization problem. However, we cannot compute the gradient at each point, so most of the typical methods to solve optimization problem cannot be applied here. Instead, we use a simple Proportional-Derivative (PD) controller to guide sensor rotation. Formally, we take the desired change of the measurement angle $\Delta\theta$ as the control variable u_t at time t. Next, based on the PTA value v_t provided from the sensor which indicates the reflected signal strength, we define the observation variable as

$$y_t = -\text{sgn}(u_{t-1})(v_t - v_{t-1}), \tag{4.1}$$

where $\text{sgn}(x) = \dfrac{|x|}{x}$ if $x \neq 0$ else $\text{sgn}(x) = 0$.

This observation function uses the direction of the prior control variable u_{t-1} and the change of the received signal strength as the feedback for our control system. Then, given a zero reference $r_t = 0$, the error variable is obtained as:

$$e_t = r_t - y_t = \text{sgn}(u_{t-1})(v_t - v_{t-1}), \tag{4.2}$$

with the intuition that we vary the orientation of the sensor towards the prior direction when observing an increased signal strength and vice-versa. The feedback control function can be designed as:

$$u_t = K_p \cdot e_t + K_d \cdot (e_t - e_{t-1}) \tag{4.3}$$

where K_p and K_d are empirically set to 3.6 and 1.8. We set $u_0 = 1°$ to give an initial perturbation. To obtain the optimal PD controller parameters, we first tune the K_p parameter. If K_p is too small, the servoing module would be too slow, and the 'steady-state' error would be larger. As a result, we want a larger $K - p$ for faster convergence. However, if the K_p parameter is too large, the system easily becomes unstable. In this case, we add K_d to help stabilize the system. After multiple trials, we find that $K_p = 3.6$ and $K_d = 1.8$ is a pair of reasonable parameter.

As the heart rate measurement requires a static measuring environment, we disable the PD controller when the error variable e_t has settled. If the subject's movement is detected (the user changes his/her pose or moves his/her leg), we start the process again to find the optimal orientation.

4.4.3 Heart Rate Estimation Module

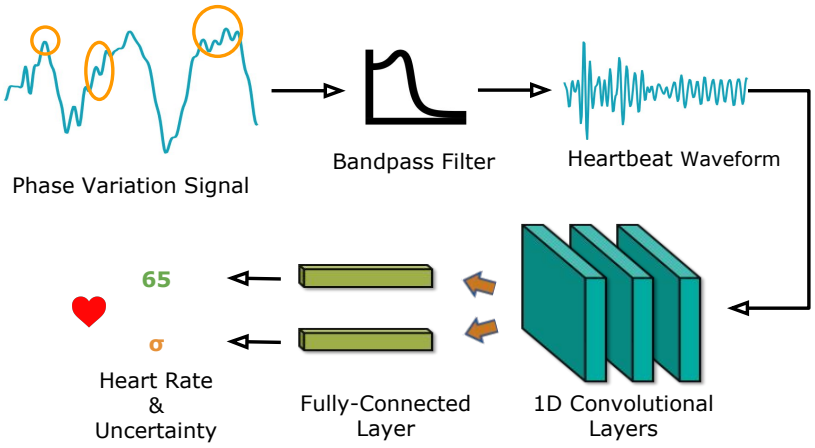


Figure 4.4: Heart Rate Estimation. A bandpass filter is used to extract heartbeat waveforms (circled in orange) from phase variation signals. A convolutional neural network takes inputs as the extracted waveforms and predicts both heart rate value and uncertainty.

4.4.3.1 Heartbeat Waveform Extraction

After rotating the sensor to the optimal orientation, the mmWave radar starts to sense the micro-displacement of the user's skin by measuring the phase variation respective to the corresponding peak in the range profile. Nevertheless, in addition to heartbeat waveforms, the phase variation also contains signals caused by other body movements that may lead to erroneous estimation. Considering the fact that heartbeat frequency lies in the band between $0.8 \sim 4$Hz [190], a biquad cascade IIR filter [215] is used in this module to extract heartbeat waveforms. For the band pass filter, a larger band may introduce more frequency components to the extracted waveforms, which could harm the performance of heart rate estimation algorithms, especially the conventional signal processing ones. However, if the band is too narrow, some valid signal components might also be filtered out, leading to incorrect estimations. The extracted waveforms are then used as inputs for next module.

4.4.3.2 Neural Network Predictor

Unlike prior works where the sensor is facing a user's chest, our mmWave radar collects signals from lower legs. For each heart beat, the chest usually has a vibration

amplitude of 400 micron, while the displacement of the skin on lower leg only has an amplitude of around 80 micron according to our measurements. Consequently, the collected signals are much weaker and our extracted heartbeat waveforms have less clear patterns than the signal collected from the reflection on the chest. Directly using these waveforms to predict heart rate, e.g., through peak extraction is difficult and error-prone. Deep neural networks have recently emerged as a powerful data-driven technique, effective at learning latent features in data. We therefore formulate heart rate estimation as a regression problem, and use a convolution neural network as the predictor. Although RNN networks, e.g., Long-Short Term Memory networks (LSTM), are widely used for modeling temporal signals, such as reconstructing pedestrian trajectories from IMU data [216], their recursive computation incurs substantial latency [180] that cannot provide timely predictions on our resource-constrained platforms. Given this real-time consideration, our network adopts three lightweight 1-D convolutional layers, followed by two fully connected layers to produce heart rate and uncertainty estimation. In particular, the number of kernels, kernel sizes and strides of all convolution layers are set to 64, 5 and 1, respectively. By taking inputs as the heartbeat waveforms with a window size of 10 seconds (200 frames), the predictor is able to provide both heart rate value and uncertainty estimation. Fig. 4.4 illustrates this process.

Neural Network Predictor transforms the signal input $\mathbf{x} \in \mathbb{R}^{200*1}$ (heartbeat waveform) into predicted heart rate $\hat{\mathbf{y}} = f^{\mathbf{W}}(\mathbf{x}) \in \mathbb{R}^1$ with model parameters $\mathbf{W}$ (i.e., weights and bias in neural networks). Typically, the optimal parameters are recovered by minimizing the mean squared error (MSE) loss between the prediction and ground truth $\mathbf{y}$, represented as:

$$loss(\mathbf{x}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 \tag{4.4}$$

The output $\hat{\mathbf{y}}$ only reflects the predicted heart rate value, given a certain set of input heartbeat waveform. In order to capture the output uncertainty as well, we extend this framework into a Bayesian model, which is introduced in the following part.

4.4.3.3 Uncertainty Estimation

Although heart rate regression with a Deep Neural Network is relatively straightforward, uncertainty estimation is non-trivial. The uncertainty reflects to what extent the predicted heart rates can be trusted. This is extremely critical to health-related problems, as wrong values will lead to serious consequences, e.g., misdiagnosis. Intuitively, the uncertainties in our problem are originated from inaccurate mmWave measurements, due to sensor biases and noises, environmental dynamics, multipath reflection and non-optimal reflection plane. We hence quantify the uncertainty of our model based on the aleatoric uncertainty which is widely used to capture inherent sensor observation noise [217, 218].

In order to estimate the uncertainty of predicted heart rates, we reformulate Equation 4.4 into a Bayesian model by defining the likelihood between the prediction $f^{\mathbf{W}}(\mathbf{x})$ and ground truth $\mathbf{y}$ as a conditional probability following a Gaussian distribution:

$$p(\mathbf{y}|f^{\mathbf{W}}(\mathbf{x})) = \frac{1}{\sqrt{2\pi\boldsymbol{\sigma}^2}} \exp\left(-\frac{(\mathbf{y} - f^{\mathbf{W}}(\mathbf{x}))^2}{2\boldsymbol{\sigma}^2}\right), \tag{4.5}$$

where $\boldsymbol{\sigma}^2$ denotes the prediction variance. To maximize the likelihood $p(\mathbf{y}|f^{\mathbf{W}}(\mathbf{x}))$, we need to determine an optimal set of parameters $\mathbf{W}^*$, which can be achieved by minimizing the negative logarithm likelihood $\log p(\mathbf{y}|f^{\mathbf{W}}(\mathbf{x}))$:

$$\begin{aligned} \mathbf{W}^* &= \arg\max_{\mathbf{W}} p(\mathbf{y}|f^{\mathbf{W}}(\mathbf{x})) \\ &= \arg\min_{\mathbf{W}} -\log p(\mathbf{y}|f^{\mathbf{W}}(\mathbf{x})) \\ &= \arg\min_{\mathbf{W}} \frac{1}{2\boldsymbol{\sigma}^2}\|\mathbf{y} - f^{\mathbf{W}}(\mathbf{x})\|_2^2 + \frac{1}{2}\log\boldsymbol{\sigma}^2 \end{aligned} \tag{4.6}$$

Thus, we can define the loss function of the model as

$$loss(\mathbf{x}) = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2}{2\boldsymbol{\sigma}^2} + \frac{1}{2}\log\boldsymbol{\sigma}^2 \tag{4.7}$$

where the model predicts a mean $\hat{\mathbf{y}}$ and variance $\boldsymbol{\sigma}^2$. From this loss function, we can see that poor predictions will encourage the network to decrease the residual term, by increasing uncertainty $\boldsymbol{\sigma}^2$. On the contrary, the term $\log\boldsymbol{\sigma}^2$ acts to prevent the

unbounded growth of the uncertainty term. In practice, we aim to learn $\mathbf{s} = \log \boldsymbol{\sigma}^2$ as it is more numerically stable [219] in this way:

$$loss(\mathbf{x}) = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2}{2\exp(\mathbf{s})} + \frac{1}{2}\mathbf{s}. \tag{4.8}$$

4.5 Implementation

In this section we introduce the hardware and software implementation, the task definition and details of the neural network training method.

4.5.1 Data Collection

4.5.1.1 Collection System

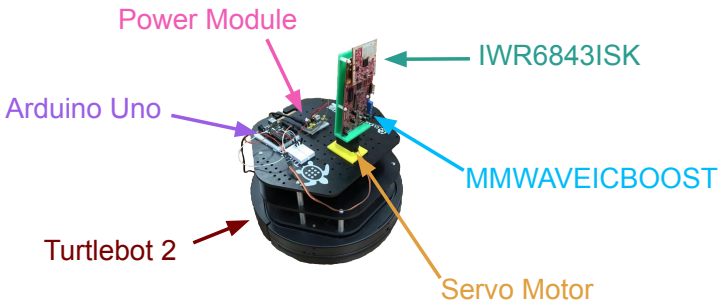


Figure 4.5: Experiment Platform. The mmWave sensor is installed on a Turtlebot 2, and an Arduino Uno serves to receive control signal from PC and rotates the Servo Motor.

We installed the mmWave radar sensor on a Turtlebot 2, as shown in Fig. 4.5. A commercial servo motor is installed on the robot platform as well and we use it to precisely control the radar rotation in 1° increments. The servo control unit is implemented by Arduino Uno. Concerning the mmWave radar, we adopt the IWR6843ISK [220], which is an emerging low-cost single-chip sensor, carried by MMWAVEICBOOST [221]. Unlike the mID we discussed in the previous Chapter, for this system we do not require the range resolution to be very large, as the micro-displacement is tracked through phase change of the received signal. Besides, with a larger range bin (smaller bandwidth), the target is less likely to move to another range bin, and we have a smaller chance of losing track of the target. As a

result, radar configuration with a relatively small bandwidth would best suit the proposed system. Both mmWave data and timestamps are logged. Polar H10, an accurate commercial chest strap heart rate monitor is used in our experiment for ground truth labelling. An Android application is developed to record Polar H10 heart rate data and timestamps via Bluetooth. The data from the radar and the data from the Polar H10 device are synchronized to the same NTP server.

4.5.1.2 Selected Poses



Figure 4.6: The 8 different poses used in evaluation.

According to the American Time Use Survey realeased in June, 2019 [222], people spend 12.61 hours on average either sitting or lying down/sleeping at home each day, accounting for over 50% of time in a day and over 80% of time at home. We thus select four different sitting poses and four different lying poses at home in our data collection (see Fig. 4.6). We collect $\sim$ 180 minutes of data from two subjects in these poses.

4.5.2 Network Training

To train the proposed network, we set the window size to 10s, which consists of 200 frames as the input data. Longer observation periods may produce better accuracy, but it would make the system track trends too slowly, while shorter observation periods are less reliable. The uncertainty component in our network is initialized

with zeros, while the remaining parts are randomly initialized. The network is implemented with PyTorch, applying the Adam optimizer with learning rate of 1×10^{-5}. The network is trained on a NVIDIA Titan X GPU with a batch size of 512 and a dropout rate of 0.2 for the linear layers.

4.6 Evaluation

In this section, we systematically evaluate mBeats. We start with the introduction of experiment setup and then compare our Neural Network based predictor with established baselines for heart rate estimation. Uncertainty estimation is examined, followed by studying the impacts of the servoing module.

4.6.1 Setup

4.6.1.1 Competing Approaches

Three commonly seen signal processing methods for extracting frequency information are used as the baseline approaches: Fast Fourier Transform (FFT) [214], Peak Count (PK) [223] and Auto-correlation (XCORR) [224].

4.6.1.2 Pre-processing

After obtaining the raw data stream, frames in the first 40 seconds and in the last 20 seconds are discarded due to system calibration. Additionally, data frames collected while the mmWave servoing module is dynamically searching for best orientation are also discarded. We take samples with a sliding window of size 200. Each sample contains the Heartbeat waveform and the synchronized ground truth.

4.6.1.3 Evaluation Protocol

Similar to [190], we use *accuracy* as the evaluation metric throughout our experiments. As mentioned in the system requirements, for clinical validity [225], a usable heart rate estimator should have at least 90% accuracy.

For training our Deep Learning based heart rate predictor, we split our data into training, validation and test sets. The three sets are prepared differently according to the experiment requirements, which will be detailed in subsequent sections.

4.6.2 Performance Comparison

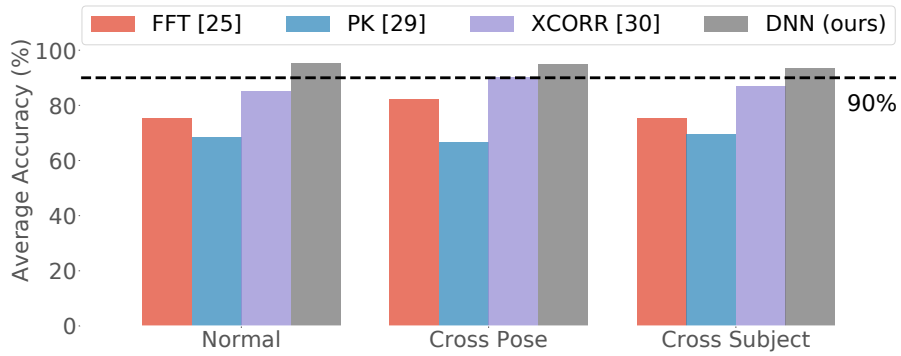


Figure 4.7: Overall performance comparison in different test categories. The 90% accuracy line is indicated for reference.

To validate the heart rate estimation accuracy and robustness of our DNN predictor, three experiments are carried out, including (i) holistic evaluation on all poses, (ii) cross pose evaluation, and (iii) cross subject evaluation. Fig. 4.7 summarizes the overall comparison results.

	FFT [214]	PK [223]	XCORR [224]	DNN (Ours)
Pose 1	77.02	65.84	91.95	**97.89**
Pose 2	89.94	65.88	88.49	**96.11**
Pose 3	52.43	75.70	48.27	**93.76**
Pose 4	83.18	73.41	92.67	**96.97**
Pose 5	93.46	64.31	94.10	**94.85**
Pose 6	74.59	67.81	85.89	**94.75**
Pose 7	64.35	64.22	91.06	**91.08**
Pose 8	74.57	70.60	87.24	**96.64**
Overall	76.19	68.47	84.96	**95.26**

Table 4.1: Holistic Evaluation Results.

4.6.2.1 Holistic Comparison

In this experiment, all data streams are equally divided into 5 continuous segments where segments 1,3,5 are for as training whilst segment 4 and segment 2 are used

for validation and testing respectively.

As shown Table 4.1, although the competing methods have previously been shown to be effective for chest-sensing scenario, they struggle to accurately estimate heart rate when it comes to lower-leg sensing. Because the sensed micro displacement of the user's skin is very weak from the calf, the competing approaches cannot reliably estimate the heart rate and be consistently accurate across different poses. For instance, as the best-performing baseline, XCORR achieves a 94.10% accuracy on Pose 5, but its accuracy on Pose 3 is as low as 48.27%. The low robustness of XCORR makes it unsuitable to real domestic scenarios where diverse daily poses are common. When considering the 90% accuracy standard to measure method efficacy, our proposed DNN estimator is the only approach that meets this under all 8 poses, with a 5.26% safety margin overall, which satisfies our system requirement R1. In contrast, FFT method only attains this accuracy in one pose, while PK fails to achieve this standard under all poses.

	FFT [214]	PK [223]	XCORR [224]	DNN (Ours)
Pose 2	91.90	63.69	91.30	**96.08**
Pose 6	72.61	69.36	88.87	**93.94**

Table 4.2: Cross Pose Test Result

4.6.2.2 Cross Pose Evaluation

We next evaluate the generalization ability of our DNN estimator across different poses. To this end, we split the dataset into 8 segments corresponding to the different poses. The data with Pose 1,3,5,7 are used as the training data, Pose 4 and 8 for model validation purpose, and Pose 2 and 6 as testing data.

As we can see in Table 4.2, our DNN approach has a strong generalization ability w.r.t. different user poses, giving 96.08% and 93.94% accuracy on the unseen data from Pose 2 and 6. The accuracy drop from holistic testing to cross pose is only 0.25%, indicating the feasibility for real world deployment. Notably, FFT, PK and XCORR are signal processing based methods and in principle insensitive to cross poses, and their accuracy is still inferior to our DNN estimator.

	FFT [214]	PK [223]	XCORR [224]	DNN (Ours)
Subject 1	75.38	72.99	83.61	**93.85**
Subject 2	75.13	65.73	90.36	**93.03**

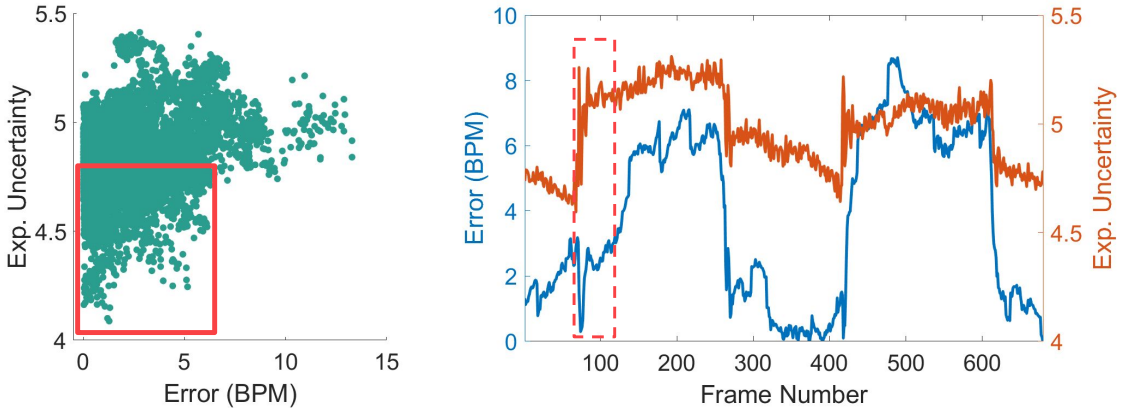
Table 4.3: Cross Subject Test Result

4.6.2.3 Cross Subject Evaluation

We further evaluate our method on cross subject testing, where the DNN estimator is tested on the data collected from subjects outside the training set. To do this, we train our model on the data from one subject, and test the model with the data from the other subject, and vice versa.

Although different people have different heartbeat patterns [226, 227], as we can see in Table 4.3, our DNN estimator still yields a testing accuracy of 93.85% and 93.03% on two different subjects respectively. Such high accuracy significantly outperforms competing approaches by around 7%. This generalisation ability is of paramount importance for real world deployment, as a crowd-sourcing approach can be used to train the DNN predictor before shipping to customers.

4.6.3 Uncertainty Evaluation



(a) Uncertainty v.s. Prediction Error. **(b)** A sample of uncertainty estimation.

Figure 4.8: Uncertainty Estimation. Here we use the exponential of uncertainty estimation since the uncertainty is defined as the logarithm of the variance.

As described in Section 4.4.3.3, our method has a prominent advantage in that is capable of providing a confidence estimation alongside the heart rate prediction.

This is important for avoiding false alarms during healthcare monitoring where predictions with high uncertainty can be filtered out. The uncertainty estimation can also work as an indicator for the downstream filtering algorithm (e.g., providing information of the observation noise in Kalman Filter) for continuous heart rate monitoring. Besides, it also acts as a clue towards improving the network by learning with more samples in undertrained situations, important for life-long learning.

In Fig. 4.8(a) we show the relationship of the heart rate prediction error and the estimated uncertainty. Recall in Section. 4.4.3.3, we define the uncertainty term as the logarithm of the variance to benefit neural network training, so here we use the exponential of uncertainty. We can see that although the estimated uncertainty is not strictly positive-correlated with the heart rate prediction error, we can still use uncertainty estimation to identify accurate and valid heart rate predictions. For example, if we only wish to keep predictions with low errors, e.g., smaller than $6BPM$, we can empirically set a exponential uncertainty threshold of around 4.8, as shown with the red box in the figure. Note that, even though in the training, we encourage the network to produce a larger uncertainty for a larger predicted error, and a small uncertainty for a more accurate prediction, in practice, a large uncertainty estimation does not necessary mean that the prediction error is large, it rather indicates that we do not know if the prediction is accurate or not. We could choose the uncertainty threshold based on the requirement of target application, for example, in real world health monitoring systems we might give up samples with uncertainty above a certain threshold as we only wish to keep valid measurements in health care applications, which satisfies our requirement R2.

Fig. 4.8(b) further demonstrates an example of uncertainty estimation on the testing set. We can see that, when the estimated uncertainty is small, the heart rate prediction is relatively accurate. For most of the time in this period, larger uncertainty means that the prediction error is relatively large. However, as we discussed, a large uncertainty does not necessarily mean the prediction is bad, as we can see from the part in the dashed red box in the figure, the prediction

error is around 3BPM, which is actually not bad, but the exponential uncertainty could reach approximately 8.

4.6.4 Impact of the mmWave Servoing Module

	FFT [214]	PK [223]	XCORR [224]	DNN (Ours)
w.o. Servoing (PTA=2.11)	78.71	75.90	62.43	**90.71**
w. Servoing (PTA=3.27)	80.74	88.77	87.61	**92.20**

Table 4.4: The overall performance with and without the mmWave servoing module.

In the prior experiments, we used heartbeat waveforms collected with the optimal observing view. In the last experiment, we investigate the importance of incorporating the mmWave servoing module into the system loop, which is responsible for searching the optimal observing view.

We collected a 3-minute data stream twice with the same user pose and robot position. The only difference is whether the servo module is used or not. Note that the initial measuring view of the mmWave radar is manually set to around 75°, deviating from the optimal angle. To have a fair comparison, we fix this initialization setup for all baseline testing. From the experimental results in Table 4.4, we can see that with the mmWave Servoing module turned on, the PTA metric is improved from an average of 2.11 to an average of 3.27. This in turn enhances the overall measuring accuracy of all methods. This is consistent with our hypothesis that PTA is vital to measurement accuracy and consequently proves the necessity of utilizing a mmWave servoing module for direction optimization before measurements.

Moreover, our method, once again, shows superior performance on the task. Even if the mmWave Servoing module is disabled, our method is still the only method that produces a satisfactory accuracy of over 90%.

4.7 Discussion

Although we demonstrated that deep learning based method can give accurate and robust heart rate prediction under different scenarios, with the inherent limitation

of learning based methods, the DNN predictor can only give heart rate predictions that it has seen before in the training set. As a result, to make the predictor truly functional, especially being able to recognize the abnormal heart rates, the predictor needs to be trained on a large dataset with variety of users and situations.

A current limitation of radar based vital sign monitoring techniques is that it has difficulties measuring heart rate from a moving subject where locomotion signals swamp the weak heart rate signals, and the phase variation needs to be tracked across range bins. Though there are some initial explorations on this topic [80, 81], huge research opportunities still exist. Also in our experiments, we find that if the user is standing, the movement of muscles in order to keep balance would overwhelm the micro-displacement caused by heart beats. As a result, future work could be considering how to best measure heart rate from a moving or standing subject with a robot mounted mmWave radar.

In the mmWave servoing module of the mBeats design, we only rotate the sensor horizontally for best sensing angle. As mentioned before, some parts of the human body (e.g., chest skin) may have larger displacement caused by heart beats, so in future work we can also explore controlling the sensor with more degree of freedom and pointing the sensor to the best sensing orientations.

For uncertainty estimation, though we encourage the network to produce larger uncertainty estimations for worse predictions, in practice many accurate predictions are also accompanied with large uncertainty estimation. Whether there exists some network structures that can produce uncertainty estimation better correlated to the heart rate predictions is a question to be answered. Besides, how to choose threshold under different situations is also worth exploring in future work.

4.8 Chapter Summary

In this chapter, we propose mBeats, a robot mounted mmWave radar system that provides periodic heart rate measurements under diverse poses without intruding on a user's daily activities. mBeats demonstrate, through the use of a novel deep-learning approach, how accurate heart rate measurements could be obtained from

the lower-leg, with corresponding uncertainty estimations. Through heart rate estimation in this work, we investigate using learning based method for micro-displacement sensing, addressing the second research question raised in Chapter 1. Results show that deep learning based method can not only produce a more accurate and robust prediction of the micro-displacement pattern than competing signal processing approaches, but can also provide an estimation of the prediction uncertainty, which is very useful especially in human vital sign monitoring.

5

Target Tracking with Radar Heatmaps

Contents

5.1 Introduction 102
5.2 Related Work 105
5.2.1 Cooperative Localization 105
5.2.2 Non-cooperative Localization 106
5.3 Conventional Drone Tracking Pipeline 107
5.3.1 FMCW Radar Background 107
5.3.2 Tracking the Drone with Radar Point Cloud 108
5.3.3 Tracking the Drone with Range-Angle Heatmaps 109
5.3.4 Problems with Conventional Drone Tracking Pipeline . 110
5.4 System Requirements 111
5.5 Proposed Method 111
5.5.1 Data Preprocessing 112
5.5.2 Network Structure 112
5.6 Implementation 113
5.6.1 Data Collection 113
5.6.2 Algorithm Implementation 115
5.7 Evaluation 115
5.7.1 Localization Evaluation 115
5.7.2 Running Time Comparison 118
5.7.3 Impact of Distance on Tracking Error 119
5.7.4 Impact of Training Data Size 120
5.7.5 Ablation Study: Impact of Network Structure 121
5.8 Discussion 122
5.9 Chapter Summary 123

Interacting with other intelligent systems also plays an important role to support better smart environment automation, and locating other smart agents, e.g., robots and drones, is one of the key enablers. Sometimes we need to locate an foreign drone, e.g., a delivery drone, and since we cannot modify the drone, marker based methods (e.g., Vicon) or transceiver based methods (e.g., UWB) will not work in this case. mmWave radar can be used to track unmodified drones, however, in practice the tracking performance with conventional signal processing methods is not as good as we would expect. This is because there are a number of factors that impact the tracking accuracy, as we mentioned in Chapter 1, e.g., multi-path reflection and noise. Besides, conventional methods treat the target as a single point, ignoring the shape, size, and orientation of the target which brings further error. In this chapter we propose tracking an unmodified drone in a data-driven way with deep neural networks and demonstrated the superiority of our proposed method on our own collected dataset, addressing research question 3.

Portions of this work is included in the paper:

- **Peijun Zhao**, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, Andrew Markham. "3-D Motion Capture of an Unmodified Drone with Single-chip Millimeter Wave Radar." In *2021 IEEE International Conference on Robotics and Automation (ICRA'2021)*.

5.1 Introduction

We envision that in the future compact robots will co-exist with humans in smart environments, and dynamic aerial robots (e.g., a small UAV or quadcopter drone) will be an important member of the robot team. Drones can help us in many ways in smart environments, such as surveillance, cleaning, inspection, and delivery. Motion capture of drones is a key enabling capability for navigation, path planning and autonomy for downstream applications. Motion capture in this context means the

estimation of precise 3D position in real-time, with an external sensor. Unlike on-board self-localization with techniques such as Visual-Inertial Odometry, localization with an external sensor can give an absolute 3D location, and does not suffer from accumulated drift. In outdoor areas with a clear sky view, the use of RTK GPS, often combined with secondary sensors such as IMU, can provide excellent absolute positioning accuracy.

In indoor environments or other GPS denied areas such as tunnels, alternative tracking approaches are typically employed. Broadly, these can be divided into optical based and RF based techniques. Vicon is one of the most widely used optical motion capture platforms [228]. These systems provide high precision 3D tracking of a reflective marker placed on a drone, provided that there is a good view from two or more spatially distributed cameras. These systems however are expensive ($10k upwards), time-consuming to set up and calibrate, and require modifying the drone to be able to track it. To prevent dead-zones, a large number of cameras (e.g., 10) need to be placed around the tracking volume and typically are mounted at height (e.g., 2-3 m) to provide a wide field of view. This clearly excludes Vicon style systems from viable and unobtrusive tracking technologies in domestic environments.

A large amount of research has considered the use of UWB ranging as a multi-lateration approach [229, 230]. These systems provide reasonable accuracy (10cm), but require the drone to carry a UWB transceiver to achieve pair-wise distance estimation. They also need a network of UWB transceivers distributed around the convex hull of the area to be tracked, leading to a time-consuming setup and calibration phase. It is typically costly to set up a UWB localization system (around $300 for an anchor and $30 for a tag).

In this chapter we present a new approach for drone motion capture that is able to operate from a single point (i.e., it does not require multiple anchors to be carefully arranged around the tracking volume). As a result, it is easy to setup and deploy in new environments. mDrone is able to track an unmodified drone in 3D with a level of precision comparable to UWB ($< 10cm$). Based on a single chip mmWave integrated radar transceiver, mDrone is low-cost (<$300

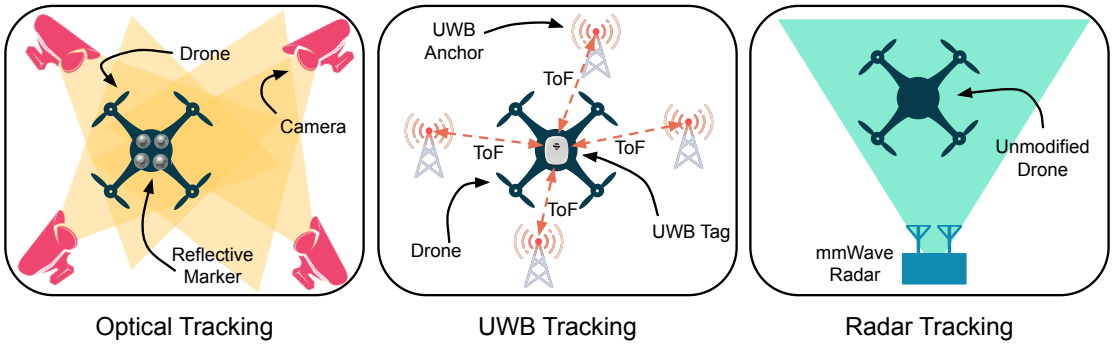


Figure 5.1: Compared to the optical tracking method and the UWB tracking method, our proposed system does not require any marker modification on the drone and only needs a low-cost mmWave radar.

for the dev kit) and low-power, and has the capability of operating in the dark and other visually challenging conditions.

Our primary innovation is the use of single chip mmWave radar which brings its own set of unique challenges. Although radar has a long history of being used to track aeroplanes at extremely long ranges (hundreds of kilometers), the majority of solutions are military/aviation grade, large, costly and power-hungry. Recent advances in microchip fabrication have led to the availability of multi-channel, wideband radar transceivers that have been successfully used in diverse applications such as SLAM [12, 110, 231, 232], industrial automation [16], and human-computer interactions [30, 96, 97, 192]. However, a number of challenges arise from the problem of drone tracking. Firstly, the drone is unmodified, i.e., it does not carry any markers to be tracked, e.g., a retro-reflector or a RF transceiver. This brings the challenge of detecting and distinguishing the drone from background reflections, and we demonstrate how to exploit the rotating propeller's Doppler effect as a natural strong reflector for the mmWave radar. Secondly, the shape and size of the drone also brings errors in the localization prediction, as the drone appears as a non-uniform blob in the radar returns. Thirdly, most of the existing signal processing techniques are computationally expensive, precluding the ability for real-time motion capture, and they produce location estimations with low accuracy due to noise and multi-path effects, as we pointed out in research challenges in Chapter 1. To address these problems, we introduce mDrone, a new deep learning

pipeline and demonstrate its performance and robustness in comparison to five conventional signal processing baselines. We believe that this low-cost motion capture platform will not only be useful as a low-cost research and development tool, but also enable a number of downstream applications in smart environments where mmWave radars become an indispensable component.

Our main contributions in this chapter include:

- We propose a deep learning method for the estimation of drone position based on mmWave radar.
- We collected 100 sequences of data and evaluated our proposed algorithm against multiple conventional signal processing algorithms. The dataset is released to the community together with the code.
- Our system achieves a mean error of less than 10cm in 3D whilst operating at 10Hz.

The remaining sections of this chapter are organized as follows: Section 5.2 talks about related works, including cooperative and non-cooperative drone localization with grounded sensors. In Section 5.3, we briefly review some related mmWave Radar background knowledge and discuss the conventional radar processing method for drone localization, which are used as competing baselines in later sections. We define the system requirements in Section 5.4. Section 5.5 introduces our proposed method. Section 5.6 shows the data acquisition and algorithm implementation details. We report the evaluation result in Section 5.7, followed by a discussion in Section 5.8. Section 5.9 finally concludes the chapter.

5.2 Related Work

5.2.1 Cooperative Localization

In cooperative tracking, modifications are made to the drone in order to track it. To achieve this, either optical markers or transceivers are installed on the drone itself. One of the most accurate 6-DOF cooperative tracking methods is based

on Optical Tracking systems (e.g., Vicon). Hemispherical markers are attached to the body of the drone to form a rigid and spatially unique pattern, which are modeled and tracked by the system [228, 233]. Optical tracking systems are expensive (tens of thousands of dollars to equip a room with an optical tracking system) and time consuming to install and calibrate. Santos et al. propose an optical tracking replacement system, for the estimation of 6-DOF drone pose, with data from multiple sensors, including RGB-D camera and on-board IMU, based on computer vision and sensor fusion approaches [234].

UWB localization systems based on RF ranging and multilateration are widely used for their lower cost, whilst still providing accurate positioning (errors typically around $10cm$ indoors, could be higher for outdoor scenarios) [229, 235, 236]. Ledergerber et al. designed a system using one-way UWB for drone self-localization [237], and Angle of Arrival estimation [238]. There are also many works that fuse UWB with other sensors for a better localization accuracy, such as UWB/IMU/Vision [239, 240], UWB/IMU [241], and UWB/Radar [242]. UWB techniques are also used for multi-UAV localization [243–245].

5.2.2 Non-cooperative Localization

Despite the fact that computer vision algorithms for object recognition and tracking with cameras are developing rapidly [246], cameras do not work robustly under extreme light conditions. Researchers have been exploring other sensors for passive drone localization, including the Light Detection And Ranging (LiDAR) sensor [247, 248]. Acoustic sensor arrays can also be used for drone localization [249], which are based on TDOA principles, but they are sensitive to ambient noise and have limited ranges [250].

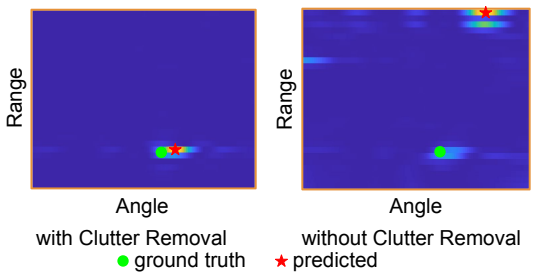
Different types of radars are also widely used for drone detection and localization [251–253], amongst which, FMCW radars have attracted a great deal of research interest [254]. In our work, we used a commercial-off-the-shelf mmWave FMCW radar, and focus on exploring the combination of signal processing and deep learning techniques to greatly improve short-range drone localization accuracy.

5.3 Conventional Drone Tracking Pipeline

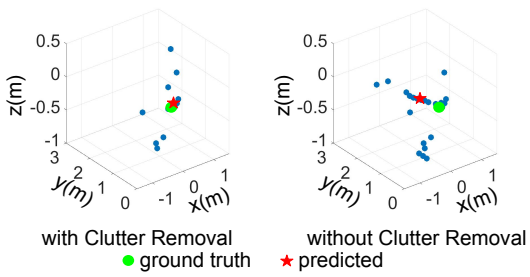
5.3.1 FMCW Radar Background

We first introduce the principles and limitations behind conventional tracking pipelines. As introduced in Chapter 2, FMCW radar detects objects with electromagnetic wave chirps whose carrier frequency increases linearly with time. To demodulate, the reflected signal is mixed with the transmitting signal and produces an Intermediate Frequency (IF) signal. The distance to an object can be calculated by extracting peak frequency components of the IF signal, typically with FFT. Also, the phase shift between chirps at a particular frequency component of the IF signal can be used to estimate the radial velocity of the object, which is caused by Doppler shift.

In this work, we consider the use of a Texas Instruments (TI) IWR6843 single chip radar, but the concepts presented here are generalizable to other platforms. The TI IWR6843 radar features a 3Tx/4Rx MIMO antenna array, which forms 12 virtual antennas (VA) based on the combination of Tx-Rx pairs [35]. The layout of the VA array is shown in Fig. 2.6 in Chapter 2. The phase differences between the antennas of the linear receiver arrays (horizontal and vertical) can be used to estimate the azimuth and elevation angle of arrival of different objects in the scene. For each frame, the radar is able to generate a data cube of complex numbers, with the axes corresponding to chirps, samples, and virtual antennas.



(a) Range-Azimuth Heatmap produced with MUSIC algorithm. Without clutter removal, the heatmap is noisier and gives incorrect results.



(b) Point Cloud. Without Clutter Removal, the reflection from static background might be mistakenly regarded as the drone location.

Figure 5.2: Effects of the Clutter Removal algorithm.

FMCW radars are particularly good at detecting moving objects as they can measure the Doppler of the targets. A UAV's rotating propellers can easily be detected and separated from the background static clutter, by a clutter removal algorithm that removes stationary objects from the range profile. As introduced in Chapter 2, this is done by subtracting the average of chirps in a frame from each chirps before doing other signal processing algorithms, as stationary objects introduces no phase shift. By removing the static clutter, the drone could be more easily detected and accurately located in radar heatmaps or point cloud, as shown in Fig. 5.2. In the remaining parts of this section we introduce conventional signal processing pipelines for drone tracking, and Clutter Removal technique is applied by default in the remaining narratives.

5.3.2 Tracking the Drone with Radar Point Cloud

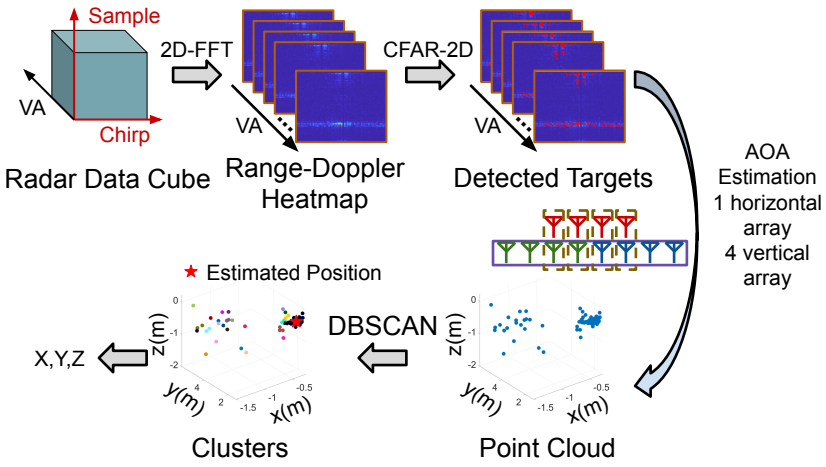


Figure 5.3: Drone tracking with radar Point Cloud.

The mmWave radar is able to generate a 3D point cloud of the scene based on the principles detailed in Chapter 2, which can be used to localize the drone. As input, the radar data cube is first processed along the chirp-sample dimension for 2D object detection, by forming a Range-Doppler heatmap. The 2-dimensional Constant False Alarm Rate (2D-CFAR) algorithm is used to extract peak points. The phase differences between the virtual antennas corresponding to a detected point are then used to estimate the 3D bearing angle of each point. These are

then combined to produce a sparse 3D point cloud. The full pipeline is shown in Fig. 5.3. Using the 3D point cloud, a clustering algorithm such as DBScan is used to segment the point cloud into clusters. As the background clutter has been removed, the largest cluster corresponds to the drone, under the assumption that the scene is relatively stationary. By taking the centroid of the largest cluster, we can estimate the position of the drone.

5.3.3 Tracking the Drone with Range-Angle Heatmaps

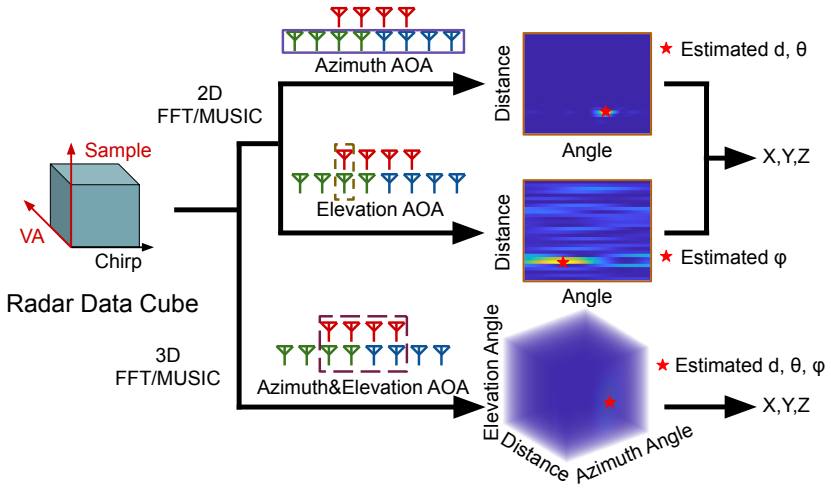


Figure 5.4: 2D (top) and 3D (bottom) FFT and MUSIC pipelines.

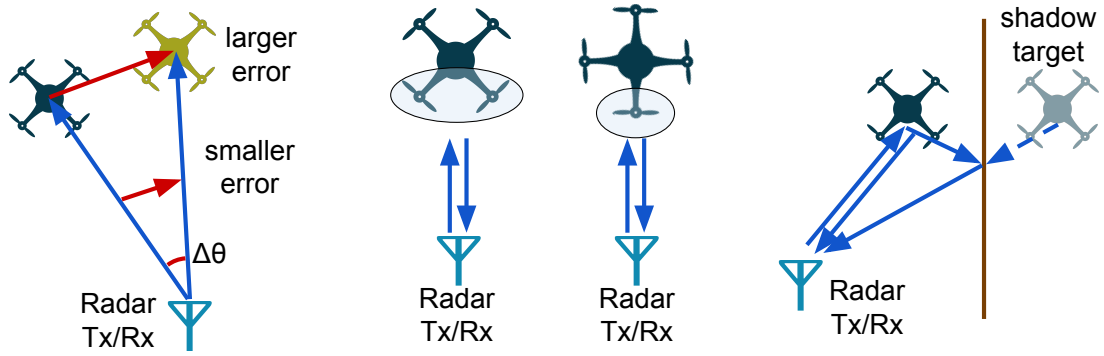
An alternative approach to estimate the location of the UAV can be performed by omitting the chirp axis of the radar cube, and only considering the Sample-VA plane. Conventional signal processing algorithms, such as the Fast Fourier Transform, or the super-resolution algorithms like MUSIC (MUltiple SIgnal Classification) algorithm [36], can be used to estimate the Range-Angle heatmaps, from which the distance as well as both the azimuth and elevation angle of arrival (AoA) of the object can be estimated. Given the distance and angle from multiple virtual antennas we are able to obtain the location of the drone.

As a refinement, we can perform 3D FFT or MUSIC directly on the cubic data formed by the Sample axis and the VA matrix shown in bottom pipeline of Fig. 5.4, and generate a 3D heatmap cube for a globally optimal estimation. The complexities

of these algorithms are however exponential in the number of dimensions. 2D FFT/MUSIC based pipelines run significantly faster than 3D FFT/MUSIC based pipelines, but are more sensitive to noise and hence lack robustness and accuracy.

5.3.4 Problems with Conventional Drone Tracking Pipeline

Although tracking a drone based on conventional signal processing algorithms is straightforward and relatively easy to implement, these methods suffer from a number of problems which lead to low accuracy in practice, as shown in Fig. 5.5.



(a) Distance **(b)** Orientation **(c)** Multi-path

Figure 5.5: Three factors that impact drone localization accuracy with conventional methods.

First of all, as mmWave radar is a polar coordinated sensor, a certain error in angle of arrival estimation would result in a larger localization error if the target is further away. Secondly, the drone is a complex object rather than a single point. Drone orientation might also impact the localization error. As the reflection mainly happens from the propellers, the drone could have different reflection pattern if oriented differently. This typically leads to a systematic localization error. Another very important factor that affects localization accuracy is multi-path. In indoor environments, when the drone is close to flat surfaces, such as walls or floor, there might be reflections on the planar surfaces. The mmWave radar typically has limited angle of arrival resolution, so it cannot distinguish the reflection that directly comes from the drone and the multi-path reflection from the 'shadow'. As a result, the reflection from the 'shadow' interferes the reflection from the actual target which leads to a large error in distance and angle estimation.

In addition, there are many empirical parameters in conventional algorithms and it can be difficult to fine-tune and assign optimal values to those parameters, especially for point cloud based pipeline. The localization results produced by conventional algorithms are not stable, and contain many outliers, because in some frames the noise or multi-path scattering overwhelms the primary signal and is wrongly identified as the target.

To overcome the above-mentioned drawbacks of conventional pipeline, we put forward a novel approach, combining signal processing with the state-of-the-art deep learning techniques, which is introduced in the next Section.

5.4 System Requirements

In this work we try to solve indoor drone localization in an uncooperative way, so the system should be able to track drones that do not carry any additional marker or transceiver. As we are targeting indoor scenarios, we want the system to be able to track the drone within a range of 5 meters.

To better support downstream applications, the proposed system should be able to estimate the 3D location of the drone with an error of around 10 cm, comparable to UWB localization systems. To avoid the drawbacks of conventional drone tracking pipelines with mmWave radar, as introduced in the previous sections, the proposed system should be able to provide stable estimations across frames, and at different distances.

The requirements of the system are listed as:

- R1: The system needs to track the unmodified drone with around $10cm$ of 3D localization error.
- R2: The system needs to work at $10Hz$.

5.5 Proposed Method

Our method for drone localization contains two parts. Firstly, the radar cube data is pre-processed into range-elevation and range-azimuth heatmaps. Secondly, a

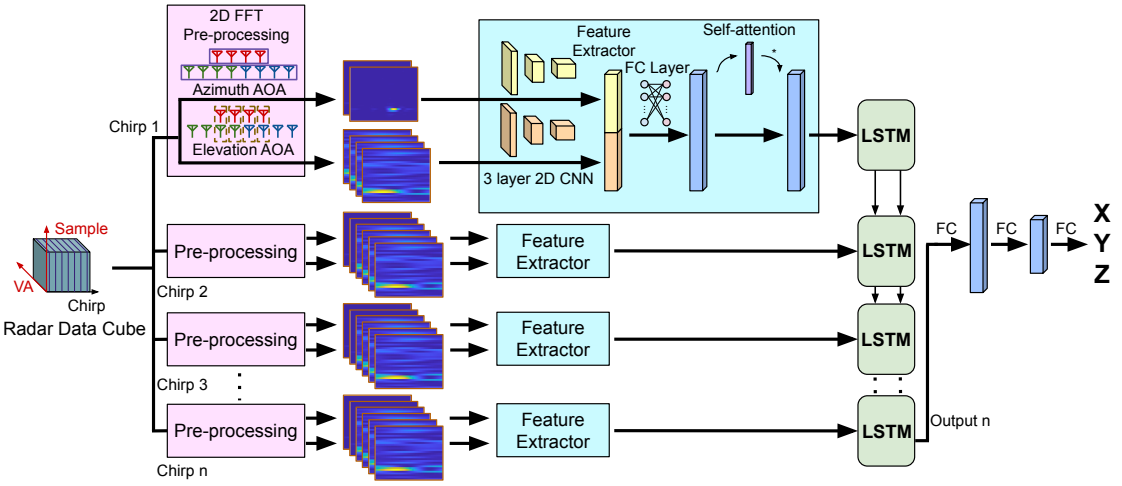


Figure 5.6: Proposed Method. For each chirp, the pre-processing module generates two range-azimuth heatmaps and four range-elevation heatmaps, which are sent into the Feature Extractor module respectively. The feature vectors of multiple chirps in a frame are input into a LSTM network and the output of the last timestamp is used to estimate the 3D location with fully-connected layers.

deep neural network, which consumes the heatmaps as input data and produces a 3D localization estimation. A demo video can be found at `https://youtu.be/CUXLhaMDBfQ`. Fig. 5.6 shows the pipeline of the proposed method and the two parts are discussed below.

5.5.1 Data Preprocessing

The data processing part is similar to a conventional 2D FFT pipeline, except for that it produces 6 heatmaps in total (2 azimuth heatmaps and 4 elevation heatmaps), rather than just one azimuth heatmap and one elevation heatmap for target localization as in the conventional 2D FFT pipeline. In addition, heatmaps of multiple chirps with the data cube are extracted and sent into the neural network.

5.5.2 Network Structure

The neural network we propose contains two parts. The first part is a feature extractor, which takes range-azimuth heatmaps and range-elevation heatmaps as input. Two 3-layer 2D CNNs are used to extract an azimuth feature vector and an elevation feature vector. The two feature vectors are concatenated together and fed into a fully-connected layer to form an overall chirp feature vector which combines

both azimuth and elevation information. A self-attention mechanism [49] is applied to the chirp feature, to assist in rejecting background noise by focusing on relevant features. The feature vectors of each chirp are sent into an LSTM network. Three fully connected layers are then used to estimate the final output from the hidden layer of LSTM at the last timestamp. Note that though our proposed method features a LSTM which takes a time-series of input, it uses the same amount of information (the data in a single frame) as the baseline methods, because the clutter removal requires the information from all chirps in a frame.

5.6 Implementation

5.6.1 Data Collection

5.6.1.1 Experiment Setup

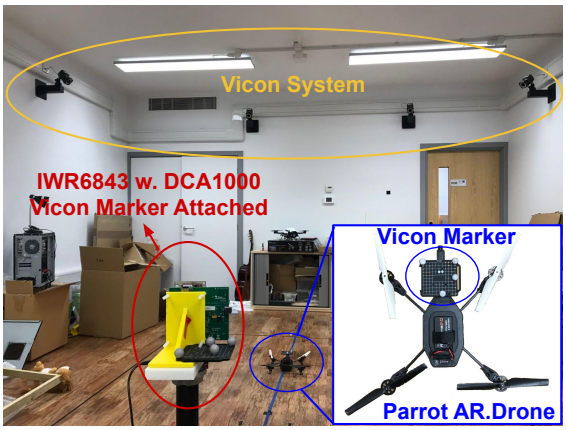


Figure 5.7: Experiment Settings

We collected 100 1-minute sequences of a drone flying in a room equipped with a Vicon system to provide accurate ground truth, as shown in Fig. 6.9. For each sequence, the data collection begins after the drone is launched into the air so the 100 sequences purely contain the data of the drone flying. The drone was manually controlled to fly randomly in the experiment space.

We use a TI IWR6843ISK carried by TI MMWAVEICBOOST board together with DCA1000EVM[1] for raw data streaming. The detailed radar configuration

[1] http://www.ti.com/tool/DCA1000EVM

Parameter	Value
Start Frequency (GHz)	60.25
Frequency Slope (MHz/μs)	72.999
Chirp Idle Time (μs)	7
Chirp Ramp Time (μs)	51.14
ADC Samples per Chirp	256
Sample Rate (ksps)	12000
Chirp Loops per Frame	128
Frame Period(s)	0.1

Table 5.1: mmWave Radar Configuration.

is shown in Table 5.1. The proposed system attempts to track the location of the drone as accurately as possible. As a result, we maximize the bandwidth in order to have a fine range resolution. In our configuration we have a bandwidth of 3.733GHz, which gives us a range resolution of around 4cm. We also want the angle-of-arrival estimation to be most accurate, so we utilize all the 3 transmitters and 4 receivers, for the best angle resolution. Using a smaller bandwidth or fewer antennas would result in lower resolution and larger error.

A desktop computer with Windows operating system is used for configuring the device and recording raw radar data, with mmWave Studio[2] software. A laptop with Ubuntu operating system and ROS is used for Vicon data recording, and the Vicon data is later parsed to extract ground truth.

5.6.1.2 Coordinate System Definition

The mmWave Radar is a right hand system, with y axis pointing forward, x axis pointing right, and z axis pointing upward.

5.6.1.3 Time Synchronization

Since the raw data streaming and Vicon data recording happen on two platforms respectively, we need to synchronize the timestamps of each frame in the mmWave data to the corresponding Vicon ground truth. The time of the two platforms are synchronized using NTP server. mmWave Studio is not able to provide timestamp

[2] `http://www.ti.com/tool/MMWAVE-STUDIO`

logging for each frame, so we record the time when the radar starts and extrapolate the timestamp of each frame based on frame period.

5.6.2 Algorithm Implementation

We have 5 baselines in total, as introduced in Section. 5.3, including Point Cloud based method, 2D FFT based method, 3D FFT based method, 2D MUSIC based method, and 3D MUSIC based method. All the conventional signal processing baselines are implemented in MATLAB. The size of the range FFT is 256, and the size of the Angle FFT is 180; in MUSIC, the range swept is 1m to 4m, with an interval of 0.1m, and the azimuth angle swept is 30° to 150°, with interval of 1°, and elevation angle $-15°$ to 15°, with interval of 1°. Sweeping with a shorter interval may increase the localization accuracy, but the complexity increases linearly. For Point Cloud based method, the localization performance depend largely on the algorithm parameters of point cloud generation and clustering. We test multiple set of parameters and choose the best one. For 2D-CFAR we empirically set the guard band size to 1, training band size to 2, and false alarm probability of 0.1. A larger CFAR false alarm probability could result in a denser point cloud, which contains more noise. Unlike the 'mID' system we discussed before, in this work we are only interested in the location of the target, rather than its shape, so we set a tight threshold to minimize the false positive detection. And for DBSCAN we set the maximum same cluster distance to 0.3m.

For our proposed method, the data pre-processing is done with MATLAB and the neural network is implemented in Python with PyTorch library.

5.7 Evaluation

5.7.1 Localization Evaluation

As our proposed methods uses deep learning techniques, we randomly picked 80 sequences as training sequences, 10 sequences as validation sequences and the remaining 10 sequences as testing sequences. All competing conventional method baselines are tested on the testing sequences. The deep neural network in our

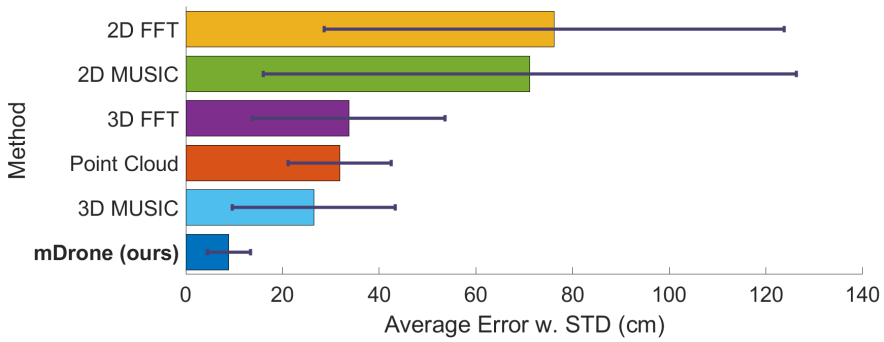


Figure 5.8: Localization Error with Standard Deviation (cm).

proposed pipeline is trained on the training sequences for 20 epochs and the model with weights that produce the best validation result are saved and tested on the testing sequences. Fig. 5.8 summarizes the result of the localization evaluation.

From the result we can see that our proposed method significantly outperforms the competing signal processing methods. The average localization error of our proposed method is $\approx$ 3 times smaller than the best competing baseline (i.e., 3D MUSIC), and satisfies the system requirements R1. Besides lower average error, our proposed method is also much more robust than competing methods, for it has a Standard Deviation less than half of the Point Cloud based method, which is the most stable baseline.

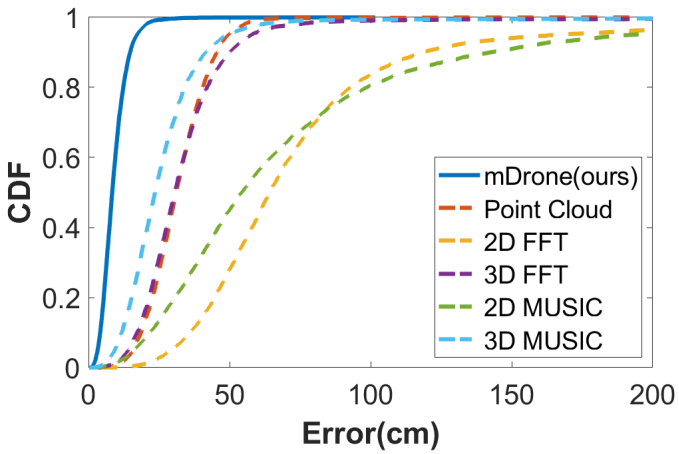
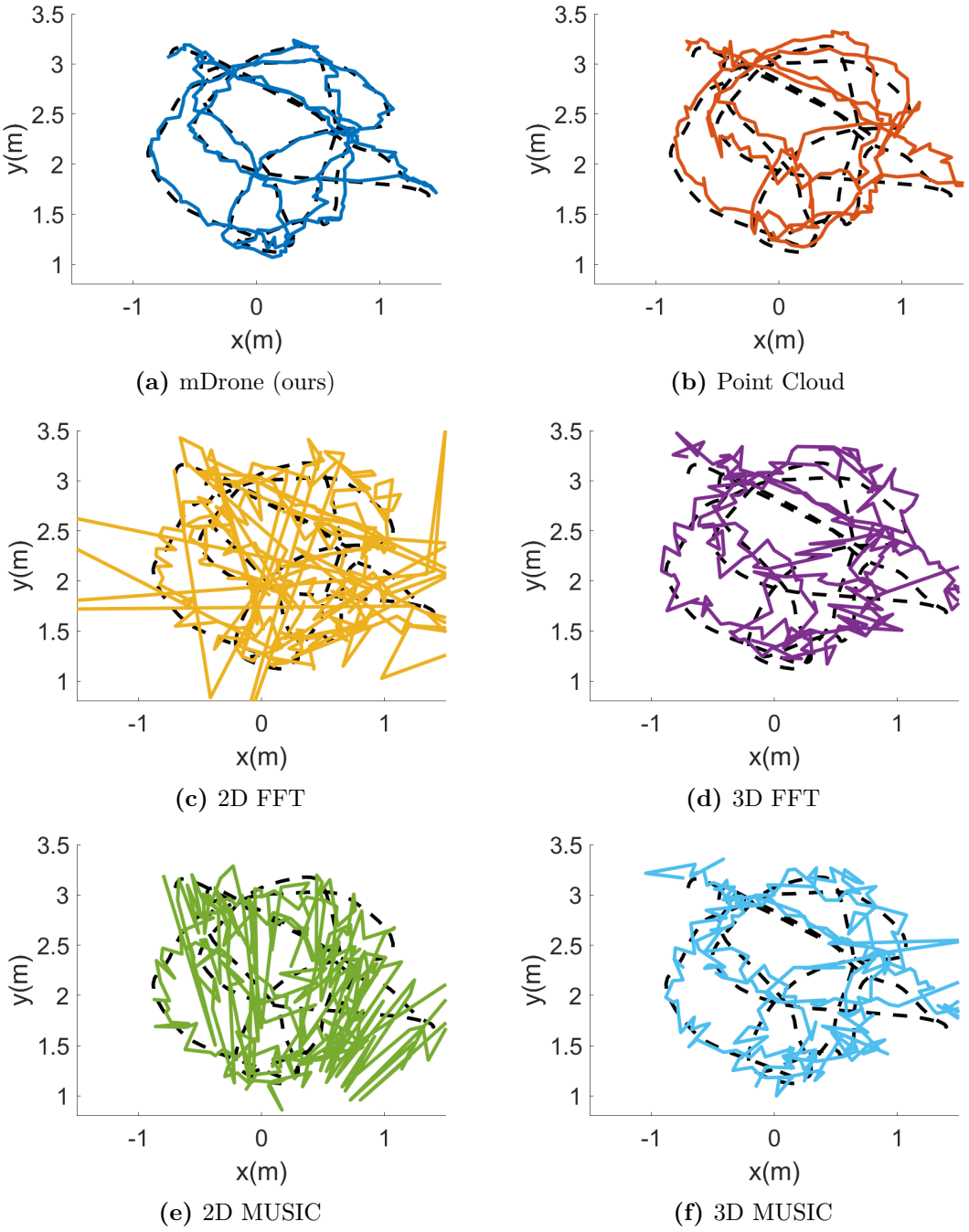


Figure 5.9: Cumulative Distribution Function (CDF) of the errors produced by different methods.

To further investigate the localization error of different methods, we plot the

Cumulative Distribution Function (CDF) of the errors of different methods, as shown in Fig. 5.9. The CDF plot shows that, for over 90% of time, our proposed learning based method produces an localization error less than 15cm, which is significantly lower than the next best competitor (3D-MUSIC: 46cm).



(a) mDrone (ours) **(b)** Point Cloud

(c) 2D FFT **(d)** 3D FFT

(e) 2D MUSIC **(f)** 3D MUSIC

Figure 5.10: Qualitative localization comparison on a sample testing sequence. Black dashed lines represent ground truth.

A sample comparison of different techniques is shown in Fig. 5.10. 2D FFT and 2D MUSIC are not stable as they search for a single global maximum on the 2D azimuth heatmap and often produce high errors, as a wrong prediction (especially wrong range prediction) in azimuth heatmap would result in a large 3D overall error. 3D FFT and 3D MUSIC search for the peak in a 3D heatmap and hence are more robust, with 3D MUSIC producing a better result thanks to the super-resolution feature, but the localization accuracy is still worse than mDrone. The Point Cloud based method is more stable than FFT and MUSIC based methods, with a relatively small standard deviation. This is because for FFT and MUSIC based methods, the location prediction is based on a single point in the 2D or 3D radar range-angle heatmap, while for Point Cloud based method, the location is predicted based on point clusters. As a result, Point Cloud based method is more robust to noise, but it is still not as good as our proposed mDrone. Also, as mentioned before, Point Cloud based method needs careful parameter tuning, which is time and labor consuming.

Applying filtering techniques like sliding window average or Kalman Filter/Smoother on the single frame localization results may further improve the stability and accuracy of mDrone. However, in this work we mainly focus on applying deep learning techniques to improve single frame localization accuracy, so utilizing the information across frames is out of scope and we will not discuss it here.

5.7.2 Running Time Comparison

Method	Time per Frame (ms)
Point Cloud	101
2D FFT	**5**
3D FFT	123
2D MUSIC	16477
2D MUSIC (parallel)	15426
3D MUSIC	282430
3D MUSIC (parallel)	220755
mDrone (Ours)	**43**

Table 5.2: Running Time Comparison.

We compare the average running time of the conventional baselines as well as our proposed method. The MUSIC pipelines are implemented with both single thread and 16 thread parallel for speeding up. The experiment is performed with a PC features a AMD Ryzen 3800X CPU with 64GB main memory and a NVIDIA RTX2080Ti graphics card. The results are shown in Table 5.2. mDrone is the second fastest among all the competing methods. The MUSIC algorithm based methods are shown to be extremely resource consuming as it takes more than 15 seconds for 2D music to process a single radar data cube, and 220 seconds for 3D music, even in parallel mode, which is practically unacceptable. The Point cloud method and 3D FFT method take around 100 ms for a radar data cube, and are more than 2 times slower than mDrone. The only method that is faster than ours, which is the 2D FFT method, produces much worse positioning results. mDrone takes around 43ms for a single frame on average. Note that according to the system requirement R2, the frame period is 100ms in this work, which means that with proper implementation, mDrone is capable of running in real time at 10Hz, which satisfies the requirement R2.

5.7.3 Impact of Distance on Tracking Error

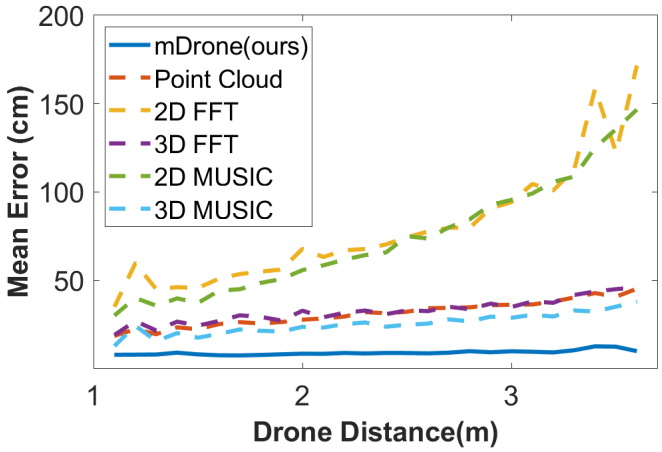


Figure 5.11: Mean Error v.s. Distance. Errors of conventional baselines increase with distance, while the error of mDrone is relatively constant.

We calculated the mean error at different distances for each method, with the step of 0.1m. Results are shown in Fig. 5.11. The errors of conventional

baseline methods, especially 2D FFT and 2D MUSIC, increase significantly with distance. This is because as we mentioned before, a certain angular error results in a larger overall position estimation error when the drone is further away, as mmWave radar is a polarized sensor. However, mDrone is able to achieve a relatively stable performance at different target distances, as the neural network takes the whole 2D range-angle heatmaps as inputs, rather than trying to find the peak point, which could produce more reliable predictions. This shows that learning based method performs better than conventional baselines for localization especially when the target is further away.

5.7.4 Impact of Training Data Size

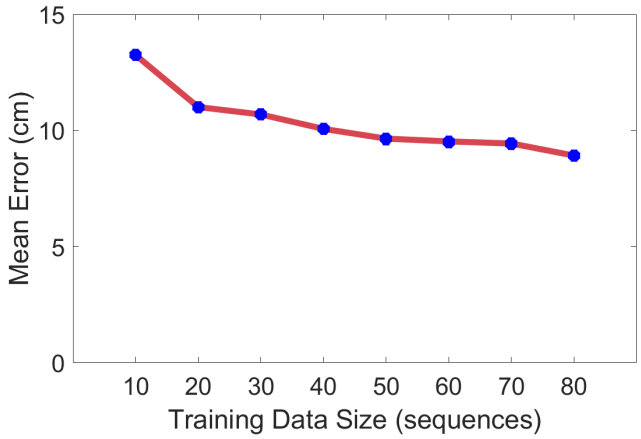


Figure 5.12: Error v.s. Training Size

Intuitively, the model trained with more training data could produce better localization results. To figure out how much training data is necessary, we trained the network with different amount of collected data. We evaluated the testing accuracy for different training data size, as shown in Fig. 5.12. We can see that even with only 10 training sequences (i.e., 10 minutes of data collection), the mean error produced by our method is less than 15cm, which is much better than competing signal processing pipelines. The error decreases as more sequences are used for training, and when we use more than 40 sequences (40 minutes of data collection), the mean error can be down to below 10cm.

5.7.5 Ablation Study: Impact of Network Structure

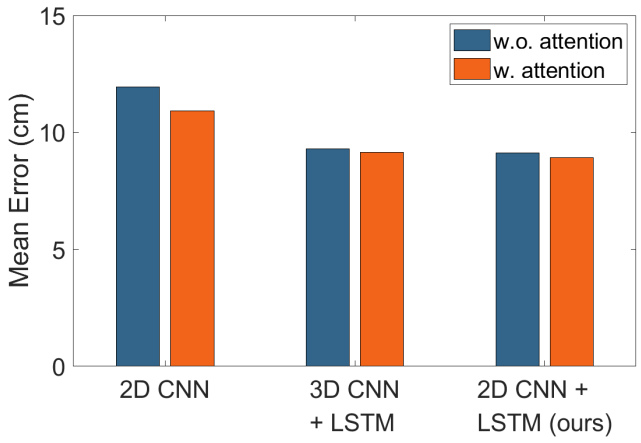


Figure 5.13: Comparison between Different Network Structures.

To show the functionality of different parts in our proposed neural network structure, we do the following three experiments:

- To prove that 2D CNN is a good choice for feature extraction, we replace the 2D CNN feature extractor with a 3D CNN feature extractor. (3D CNN + LSTM in Fig. 5.13)
- To show the effect of LSTM, we remove the LSTM and only use the features extracted by the feature extractor from the first chirp to estimate the localization result. (2D CNN in Fig. 5.13)
- To prove the effect of self-attention, we remove the self-attention module in each model. (Dark blue bars in Fig. 5.13)

Each of the above-mentioned models is trained with the same 80 training sequences for 20 epochs. The accuracy comparison is shown in Fig. 5.13. As we can see from the results, the neural network produces better results when using a LSTM layer after the feature extractor than directly using FC layers to produce the final output by around 20%. Using either 3D CNN or 2D CNN as feature extractor produces similar results, while a 2D CNN is slightly better than 3D CNN by around 2.5%. For all three structures, adding self-attention on the extracted

feature increases the model performance by around 1.5% to 8%. The results as a whole though demonstrate the utility of data driven techniques to track the moving drone, being relatively insensitive to the precise network structure.

5.8 Discussion

In the evaluation we show that our proposed learning based method has superior localization accuracy than conventional signal processing baselines, and is able to process one frame in less than 100ms, indicating that it could be used in real time in a 10Hz system with proper implementation. Utilizing the inter-frame information with filtering algorithms (e.g., Kalman Filter) could potentially improve the performance, and is worth exploring in future works.

Besides, as we discussed, the localization performance of mDrone is relatively stable at different target distances, which implies that the tracking distance could be potentially extended further. Due to venue restrictions, we only collected data of drones flying within 5 meters. However, we believe that the system can work in a much longer range as the mmWave Radar can detect objects more than $20m$ away, which could be potentially explored in future works.

Future works could also include studying the impact of drone speed on localization accuracy, the ability to generalize to different locations and completely different drone models (e.g., helicopter, hexacopter, mini-drones, etc., as different drones have different radar cross sections [255]), different sensor setups, as well as simultaneously tracking multiple drones.

Also, the learning based method proposed in this chapter requires a certain amount of training data with labeled ground truth, as discussed in Section 5.7. However in practice, we do not always have the opportunities to collect training data for certain environments. We believe that simulation could also be a very promising direction, which is already being explored for other applications, such as mmWave radar activity recognition [44]. If we could generate synthesis training data for certain environments, we do not need to rely on other sensors to collect training data, which makes our proposed method much more useful in real world scenarios.

5.9 Chapter Summary

In this chapter, we demonstrate that learning based methods can have much better tracking accuracy and robustness compared to the signal processing baselines, addressing research question 3 raised in Chapter 1. We propose mDrone, a method of non-cooperative drone localization with a grounded millimeter wave radar, as an alternative for optical tracking and UWB. mmWave radar is very suitable for localizing drones, as it is very sensitive to moving objects, so the drone can be easily detected by its rotating propellers even when hovering. mDrone consists of a two-step tracking pipeline and is compared to several conventional signal processing baselines. Results show that our method can achieve an average localization error of 8.92cm, which is more than 3 times better compared to the next best baseline method, without having to carefully tune the parameters as in conventional signal processing baselines. mDrone is also more stable than conventional signal processing methods, and is efficient enough to be used in real time at 10Hz.

6

End-to-end Learning with Learnable Pre-processing

Contents

6.1 Introduction 126
6.2 Related Work 129
6.2.1 mmWave Radar for Human-centered Applications . . . 129
6.2.2 End-to-end mmWave Radar Gesture/Activity Recognition Methods 131
6.2.3 Complex Neural Network on Radar 131
6.3 Range, Doppler, and Angle-of-arrival estimation with DFT 132
6.4 Gesture and activity recognition models 134
6.5 Requirements 135
6.6 Learnable pre-processing 136
6.6.1 Problems with Conventional DFT pre-processing 136
6.6.2 CubeLearn Module Architecture 138
6.6.3 CubeLearn Module Weight Initialization 140
6.6.4 Effect of CubeLearn Module 141
6.7 Experiment Setup 143
6.7.1 Experiment Design 143
6.7.2 Data Collection Setup 145
6.7.3 Neural Network Implementation 146
6.8 Evaluation 147
6.8.1 Hand Gesture Recognition Evaluation 148
6.8.2 Generalization to AGR and HAR Task 150
6.8.3 Convergence Analysis 152
6.9 Ablation Study 153
6.9.1 Impact of CubeLearn Module Learning Rate 153

6.9.2 Impact of Weight Initialization Method 155
6.9.3 Impact of Network Structure 158
6.10 Complexity and Running Time Analysis 163
6.11 Impact of Raw Data Size 165
6.12 Discussion and Future Work 166
6.12.1 Model Selection under Different Situations 166
6.12.2 Future Work 167
6.13 Chapter Summary 168

Despite the fact that the conventional DFT processing introduced in Chapter 2 is a widely adopted and efficient method for radar data pre-processing, it is fixed and does not have the ability to adapt to different tasks, which makes it sub-optimal in some cases. In this chapter we discuss how to use neural networks to directly and adaptively extract information from raw radar signals to increase the performance of human-centered recognition tasks, especially for lower-end radars. To address Research Question 4 raised in Chapter 1, we propose a novel learnable pre-processing module, called CubeLearn, using stacked complex linear layers to replace conventional DFT for FMCW radar recognition tasks. We evaluate our proposed method on three datasets (hand gesture recognition, arm gesture recognition and human activity recognition) that we collected, and demonstrated that the CubeLearn module could consistently improve the recognition accuracy with different pre-processing and classifier combinations, and on different tasks. We provide a comprehensive ablation study of module learning rate, module initialization and module structure. We also studied the complexity of our proposed method and the running time on both PC and edge devices.

Portions of this work is included in the paper:

- **Peijun Zhao**, Chris Xiaoxuan Lu, Bing Wang, Niki Trigoni, Andrew Markham. "CubeLearn: End-to-end Learning for Human Motion Recognition from Raw mmWave Radar Signals." (Under Review)

6.1 Introduction

mmWave FMCW radars have been used on cars for assisted driving for many years. With the recent development in low cost single-chip mmWave radar, (e.g., TI mmWave sensors [256] and Infineon mmWave sensors [257] including Google Soli [30]), more and more possibilities have been explored in indoor applications, including vital sign monitoring [190, 258], gesture recognition [20, 95, 97], fall detection [94, 259, 260], human activity recognition (HAR) [44, 127, 192], etc. As mmWave radars are less obtrusive than cameras, they have significant potential for domestic applications, where concerns about privacy dominate.

Typical methods for FMCW radar gesture or activity recognition follow a two-stage pipeline: radar signal pre-processing with conventional signal processing techniques, and a data-driven classifier for task-specific recognition purposes. In the classical radar processing chain, there are a number of different levels of data representations, ranging from Range Profile signatures to heatmaps through to point clouds, as introduced in Chapter 2. For gesture recognition and activity recognition tasks, heatmaps and point clouds are the most widely adopted data representations, with heatmaps more frequently used than point clouds [261, 262]. Besides the inherent sparsity of the point cloud, other reasons for not using the point cloud include the fact that point cloud generation relies on handcrafted parameters (e.g., for 2DCFAR), and for certain radar configurations, it is not possible to generate point cloud (e.g., Single-input Single-output (SISO) radar). Radar heatmaps are usually generated with Discrete Fourier Transform (DFT) which has correspondingly fewer heuristic parameters in comparison to point cloud generation. The DFT serves as a very efficient way to extract Range, Doppler and Angle-of-Aarrival (AoA) information from the raw radar signal. However, it also has limitations, including: 1) it uses preset, equally spaced transformation bases, which could be sub-optimal; 2) limited resolution; 3) heatmaps contain a large portion of redundant information, which wastes computational resources, and can even be harmful.

Some recent research in the area of data-driven radar super-resolution has shown that, with proper design, deep neural networks can extract richer information than

the DFT and can possibly improve radar resolution [39, 40]. In fact, researchers have been trying to use end-to-end deep neural networks for radar applications in recent years. The difficulty here lies in that the raw radar ADC samples are complex values, and both the magnitude and the phase are important. In previous work on end-to-end learning for radar applications, either only the real part is used [45, 263], which loses some information; or the real part and the imaginary part are treated as two channels [264], which loses the physical meaning, and using a specific network structure could make it less universal.

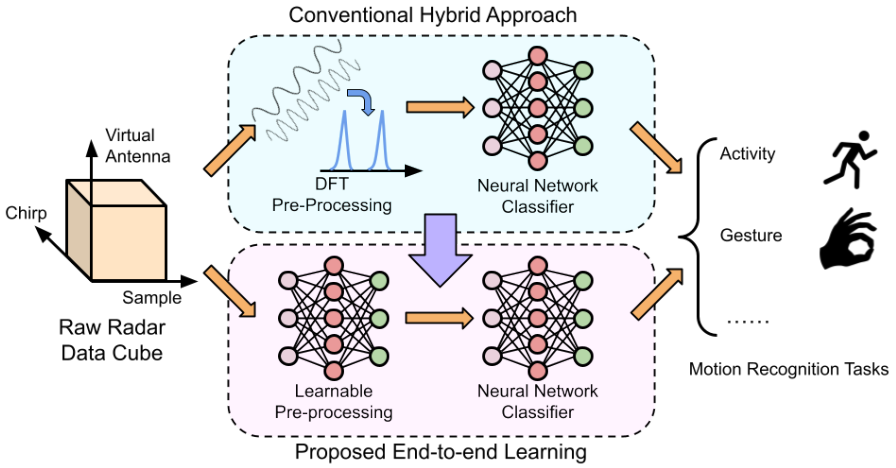


Figure 6.1: We propose learnable pre-processing module to replace conventional DFT and build end-to-end deep neural networks for mmWave FMCW radar interactive recognition tasks.

In this work, we explore replacing DFT pre-processing with a learnable pre-processing module called Cubelearn, as shown in Fig. 6.1, to improve the performance of heatmap-based indoor mmWave FMCW radar gesture and activity recognition pipelines. To the authors' best knowledge, this is the first work to use stacked complex linear layers to replace DFT pre-processing for FMCW radar pipelines, and the first in-depth evaluation of a purely end-to-end radar network. Our intuition is that by making the entire network learnable and data-driven, we can allow the network to better focus on important radar features, leading to higher overall accuracy. In particular, significant performance improvements are observed for those previously weaker models, indicating that our proposed learnable module could be especially helpful for low-end radars (e.g., SISO radars, or those with

limited computational capacity) to achieve recognition performance close to a more sophisticated radar. Besides, compared to previous end-to-end works where the proposed methods are designed for specific applications, our work proposes a more generic method for different radar recognition tasks.

Our contributions in this work are listed as follows:

- We propose CubeLearn, a learnable pre-processing module to replace the conventional DFT pre-processing, allowing the network to focus on task-specific radar signatures.
- Through extensive evaluation, we show that our proposed method can increase the accuracy of different gesture/activity recognition pipelines and accelerate the training process.
- We also provide a detailed, in-depth exploration of the impact of different pre-processing and classifier combinations, in terms of task accuracy and computational load on both PC and Raspberry Pi devices.
- We release the code of the proposed CubeLearn module and our own collected dataset to encourage further research on this topic.

The rest of this chapter is organized as follows: In Section 6.2 we provide an overview of the related works, including the existing hybrid pipelines for gesture and activity recognition tasks, as well as other end-to-end architectures. In Section 6.3, we briefly review the conventional DFT signal pre-processing of radar signals previously introduced in Chapter 2, as it serves as the basis for the discussions in later parts of this chapter. In Section 6.4 we introduce common pre-processing and neural network classifier pipelines for motion recognition tasks, including those that appeared in previous works. In Section 6.5 we discuss the design requirements of the network structure and present our proposed learnable pre-processing module in Section 6.6. Section 6.7 introduces the experiment settings and the evaluation results are discussed in Section 6.8. We provide a comprehensive ablation study on the learning rate of our proposed module, the weight initialization method and

the module structure in Section 6.9. In Section 6.10 we study the efficiency of the proposed method on PC and edge devices, and we evaluate the impact of input raw data size on the accuracy and running time on edge device in Section 6.11. We give a discussion in Section 6.12 about the model selection under different situations and potential future works. Section 6.13 finally summarizes this chapter.

6.2 Related Work

6.2.1 mmWave Radar for Human-centered Applications

With the development of small-sized single-chip mmWave radar, the research interests for mmWave radar based human-centered applications has grown exponentially in the HCI community. Because of the ability to sense micro-movements, mmWave radar has been used for vital sign monitoring [80, 190], which is further extended to seismocardiography estimation through deep learning [79]. It has also been widely studied to use mmWave radar for fall detection [94, 259, 260], which is useful if installed in privacy-sensitive rooms like bathrooms for the good privacy-preserving properties of mmWave radar. Qian et al. also brought synthetic aperture radar principles to single-chip mmWave radar for accurate 3D point cloud generation [21].

Pre-processing		**Feature Extraction**	**Classifier**	**Related Works**
Point Cloud		Convert to 'CPDP'	2DCNN	[97]
		Resampling	PointNet++-LSTM	[20]
		Voxelization	3DCNN-LSTM	[192]
1D Profile	R, A, D Profile	Trajectory Extraction + Kalman Filter	LSTM	[124]
2D Heatmap	D-T	-	2DCNN	[44, 125–127, 265]
	R-D	-	2DCNN	[128]
	R-D	-	2DCNN-LSTM	[92, 95]
	R-D	-	2DCNN-TCN	[266]
	R-D	Hand-crafted Feature Extraction	ANN	[93]
	R-D + R-A	-	2DCNN-LSTM	[42]
	R-T + D-T	-	2DCNN	[129]
	R-T + A-T	-	2DCNN	[130]
3D Heatmap	R-D-T	-	3DCNN	[43, 131]
	R-D-A	Calculate Weighted R-A Image and R-D Image	2DCNN + LSTM	[132]

Table 6.1: Deep Learning based mmWave FMCW radar gesture and activity recognition. Abbreviation: R–Range; D–Doppler; A–Angle; T–Time.

For mmWave radar sensing, the most studied areas in recent years could include human activity recognition and human gesture recognition, as they are very

important topics in human-computer interactions. In fact, they are similar tasks on the perspective of radar signal processing. Both activity and gesture recognition involves the classification of signal reflections on movements in a short period of time, and can be solved similarly. Various approaches have been used for activity and gesture classification, and most of them adopt a hybrid pipeline of conventional signal pre-processing and machine learning classifier. Some recent works on mmWave FMCW radar gesture and activity recognition are summarized in Table 6.1.

There are typically three ways for using point cloud as input for neural networks: projection, voxelization and PointNet-style architectures [267]. mmWave radar generates a point cloud with a various number of points inside for each frame, and some works project the 3D point cloud onto X-Y plane or X-Z plane, which can be further processed with mature computer vision techniques. Voxelization can turn the 3D point cloud into a fixed size input similar to images, and clustering and target localization are usually needed before voxelization to remove noise and specify the target area for voxelization. Voxelization is a very straight-forward method for converting a point cloud to an 3D grid representation, but it typically has a high computational cost and consumes a lot of memory especially for sparse point clouds like mmWave radar point cloud [90]. PointNet has been proved to have exceptional performance on extracting point cloud features, but it needs a fixed number of points as input for each frame, so a down-sample or up-sample is required for each frame.

Mainstream recognition methods are based on 1D/2D/3D heatmaps (sometimes 1D representations are called 'profiles'), and in most cases the heatmap data is directly used as neural network input. Heatmap data contains more information than point clouds, and are fixed in size, allowing them to be directly fed into the neural network. As introduced in Chapter 2, the conventional DFT pre-processing is typically used to extract Range, Doppler and AoA information from the raw radar data cube. There are many types of 2D and 3D heatmaps, such as Range-Doppler heatmaps used in [42, 92, 93, 95, 128, 266]. The frame dimension can be used as a separate dimension for the neural network to extract temporal information [42, 92, 95, 124, 132, 266], or used together with other dimensions

to be processed as part of the heatmap [43, 125, 126, 131, 268]. In our work we consider the heatmap based pipelines in previous works, together with other possible combinations, as our baselines.

6.2.2 End-to-end mmWave Radar Gesture/Activity Recognition Methods

Using end-to-end neural networks for radar applications is still rarely studied, as the main difficulty in designing an end-to-end structure is to handle the complex input. The magnitude and phase information are both important in the raw IF signal, as the magnitude contains information regarding target distance, and the phase information can be used for velocity estimation and AoA estimation. Some previous works tried different ways to handle the complex input data. Sakamoto et al. proposed converting the I/Q data received by CW radar receiver into an image for hand gesture information with convolutional neural networks [269]. Zhao et al. proposed treating the real and imaginary part as two separate channels and processed similar to images with real-valued convolutional layers for human activity classification [264].

There are also some works that use real-valued input. Stadelmayer and Santra proposed using a parametric convolutional neural network (2D Sinc Filter and 2D Wavelet Filter) [45] to extract Range and Doppler information from raw radar data. Ye et al. proposed using real-valued convolutional layers with Fourier initialization for human-activity classification [263, 270] on Continuous-wave radar data. However, their methods lose half of the information from the imaginary part.

To the authors best knowledge, this work is the first to use stacked complex linear layers to replace the conventional DFT pre-processing, and to build an end-to-end deep neural network together with downstream deep classifier, which directly takes raw radar complex data cube as input for common HCI recognition tasks.

6.2.3 Complex Neural Network on Radar

Most mature neural network structures are real-valued, i.e., they have real-valued weights, take real-valued input and produce real-valued output. However, real-valued

neural network are not very suited to handling complex inputs. Researchers have been looking at complex-valued neural networks for decades [271–273]. Complex-valued neural networks is an area that is still being actively researched, and is gaining more and more popularity, especially in the radar signal processing area. Radar signal is typically complex-valued, which cannot be directly used as input for real-valued neural networks. Recent research on applying complex-valued neural network on radar data has shown great potentials. Many researchers use complex-valued convolutional neural networks for Synthetic-aperture radar (SAR) image object detection and classification [61–64] as well as SAR image generation [274]. Cho et al. used complex-valued neural network with attention mechanism for ego-velocity estimation using an FMCW radar installed on a car [275]. Brooks et al. used complex-valued convolutional layer for drone classification [60]. As complex neural networks are still under research and development, there's reason to believe that there will be increasing research into applying complex neural network structures to process radar data.

6.3 Range, Doppler, and Angle-of-arrival estimation with DFT

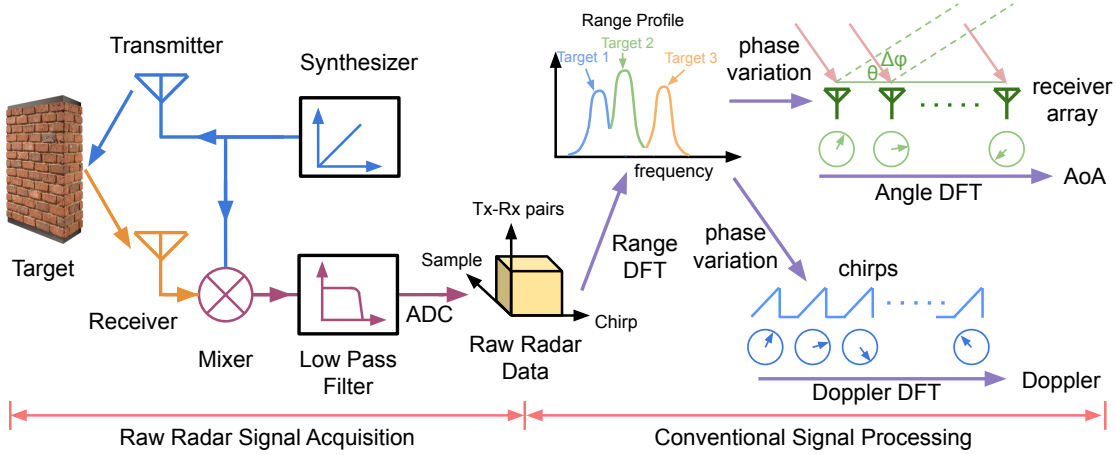


Figure 6.2: Basic mmWave FMCW Radar architecture and conventional DFT signal processing.

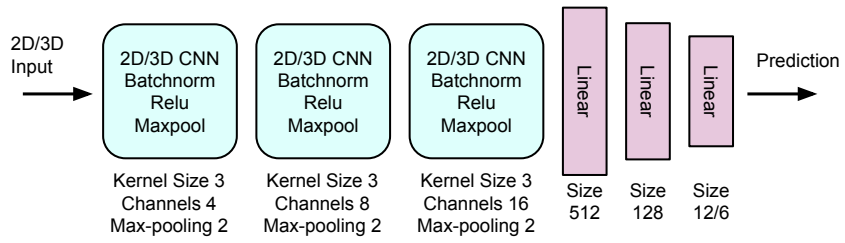
We introduced the background of basic principles and conventional signal processing of mmWave FMCW radar in Chapter 2, which can be summarized in Fig. 6.2. Recall that in a typical configuration, a frame consists of multiple chirps, which are short periods of signals whose frequency increases linearly with time. During each chirp, a number of analog-digital converter (ADC) samples are taken at each receiver. We use Texas Instruments IWR6843 mmWave radar in this work, which features a 3 transmitter, 4 receiver MIMO antenna array, but we note that our architecture is easily generalizable to different antenna configurations. For each chirp, typically a single transmitter is activated (with the TDM-MIMO [276] principle), and the received signals at each receiver antenna are mixed with the transmitting signal (which essentially calculates their frequency differences), followed by low-pass filters to produce an intermediate frequency (IF) signal, which is further sampled by ADC. For each frame, mmWave radar generates a raw data cube, with the three axes namely corresponding to: samples in a chirp, chirp loops, and Tx-Rx pairs. We can extract information regarding the distance to the targets, Doppler information, and AoA along the 3 dimensions of the raw radar data cube, respectively.

As the transmitting frequency increases linearly with time during a chirp, for a target at a certain distance, the frequency difference between the transmitting signal and the receiving signal will be a fixed value, which equals to the round trip time multiplied by the frequency slope of the chirp. As a result, for each target in the scene, there will be a corresponding frequency component in the IF signal, and we can estimate the distance to different targets by extracting different frequency components with DFT, which produces a 'Range Profile'. The phase of a peak corresponding to a certain target in the Range profile will change across chirps if the target is moving due to the slight round-trip distance change, and we can apply another DFT across chirps to extract the phase variation in order to infer the radial velocity of the object, which is often called 'Doppler DFT'. Furthermore, the AoA introduces phase difference at the antennas in the receiver array also due to the slight difference in the total length the signal travels. This can also be estimated with DFT, which is called 'Angle DFT'. Transforming the

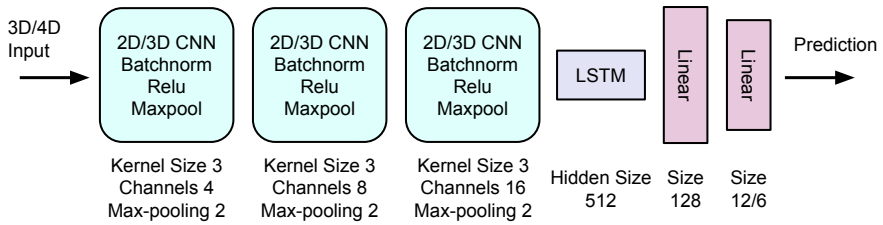
information from the sampled time/space domain to the frequency domain is the key for heatmap generation as introduced in the following section.

6.4 Gesture and activity recognition models

As discussed in the previous section, the mmWave radar is able to produce a raw data cube for each frame, with 3 axes as sample, chirp and virtual antenna, respectively, and range, Doppler and AoA information can be extracted from these three dimensions. To represent the information on these three dimensions. We often adopt heatmaps as a straightforward data representation. Peaks in the heatmaps correspond to detected targets. For example, a peak in the 2D Range-Doppler heatmap represents a target at a certain distance with a certain radial velocity. Since in this work we are studying gesture/activity recognition, which typically lasts several frames, we have an additional frame (time) dimension. As a result, with different combinations of information from three dimensions in the data cube, together with the frame dimension, we supply 2D, 3D or 4D data as input to the downstream neural network classifier.



(a) CNN classifier



(b) CNN-LSTM classifier

Figure 6.3: Two types of downstream classifier architectures we study in this work.

In this study, we evaluate two types of commonly adopted neural network classifiers: CNN and CNN-LSTM. The architecture of the two types of classifiers

is shown in Fig. 6.3. We use three convolutional layers, each followed by batch normalization, activation (ReLU), and max-pooling. The output of the convolutional layers is flattened, followed by either LSTM and two fully connected layers, or three fully connected layers, to produce the prediction.

Data Cube Slicing	Pre-processing	Classifier
first Tx-Rx pair and first chirp	R-T	2DCNN
first Tx-Rx pair, range aggregation	D-T	2DCNN
first chirp, range aggregation	A-T	2DCNN
first Tx-Rx pair	R-D-T	3DCNN/2DCNN-LSTM
first chirp	R-A-T	3DCNN/2DCNN-LSTM
whole cube, range aggregation	D-A-T	3DCNN/2DCNN-LSTM
whole cube	R-D-A-T	3DCNN-LSTM

Table 6.2: Heatmap based mmWave FMCW radar gesture and activity recognition.

The frame can sometimes be used as one dimension in the heatmap, e.g., in a Range-Doppler-Time Cube, which can be fed into 3DCNN classifier, or used as sequence timestamps, to act as input into LSTM. We use TABLE 6.2 to summarize possible data slicing, pre-processing and downstream classifier combinations. Note that since we are using FMCW radar, for Doppler-Time, Angle-Time and Doppler-Angle-Time pre-processing, we still extract Range information first, and then estimate Doppler and/or AoA information for each Range bin, then sum along the Range axis. Some of the combinations have appeared in previous works, as we discuss in Section 6.2, and in these works the selection of the pipeline is largely based on the target application and radar configuration.

6.5 Requirements

In this work we aim to build an end-to-end neural network for general motion recognition tasks, by replacing the conventional DFT with a neural network pre-processing module. We believe that in this way, the network can adaptively extract the most important features, and avoid the drawbacks of conventional DFT pre-processing as discussed before. As the radar signal is complex-valued, the most important requirement for the architectural design is that the proposed system should be able to consume complex values. The proposed pre-processing module also needs to have stronger feature extraction abilities than conventional DFT.

We also want our system to be task agnostic, and can be used for a wide variety of human motion recognition from a few centimeters (e.g., hand gesture recognition) to meter level (e.g. activity recognition). Note that micro-motion recognition (micron level) is not considered in the proposed system.

To summarize, the requirements of the proposed method are:

- R1: The proposed method can better extract features from raw data than DFT pre-processing based methods.
- R2: The proposed method can be used for different tasks.

6.6 Learnable pre-processing

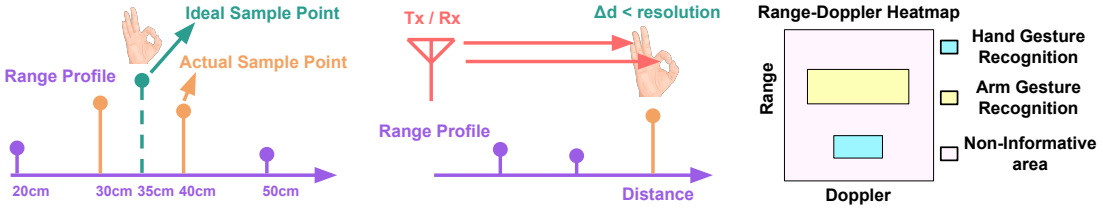
6.6.1 Problems with Conventional DFT pre-processing

For raw radar signals, the information of the targets is not encoded in certain data points, but rather hidden in the phase and magnitude variation of the signal. For example, the distances to the targets are encoded in the frequency components of a single chirp, and the Doppler and AoA information are contained in the phase variation, either from time domain (chirps) or spatial domain (receiver array).

To expose this information, the pre-processing is often performed with conventional signal processing techniques, typically through DFT, as discussed before. The DFT can be viewed as a linear transformation of the raw radar data from the time/spatial domain to the frequency domain with equally spaced Fourier bases. By transforming to the frequency domain, the information corresponding to the targets is directly exposed. The pre-processed output can be largely regarded as 2D/3D images to be fed into downstream neural networks based on architectures widely explored in the computer vision domain.

Though conventional DFT pre-processing is simple, efficient and straightforward, there are three issues which impact its utility:

Lack of adaptive transformation bases. All the Fourier transformation bases are equally spaced and fixed, and the transformed data points might not best represent the target. For example in Fig. 6.4 (a), if we have a radar configuration where the



(a) With fixed bases, the position of the target falls between two sample points.

(b) Resolution limitation makes it hard to capture fine-grained movements.

(c) The informative area is only a small portion of the whole conventional DFT output.

Figure 6.4: Three limitations of the conventional DFT pre-processing. In (a) and (b) Range dimension is used for illustration, and the problem also exists for Doppler and AoA dimensions.

distance between two data points in the Range Profile is $10cm$. The position of the actual target could be $35cm$ which falls between two sample points ($30cm$ and $40cm$) in the Range Profile. Although the adjacent data points may contain the target information, the representation is sub-optimal and the downstream network will need to interpolate adjacent data points.

Limited resolution. Another problem is that DFT has inherent resolution limitation, which makes it hard for the DFT transformation output to represent the target properties on certain tasks. For example, the Range resolution of mmWave radar is typically $> 5cm$, which makes it difficult to distinguish two fingers in hand gesture recognition. Also, the resolution limitations of Doppler and AoA make it hard to capture fine-grained movements, e.g., movements of two adjacent fingers, as shown in Fig. 6.4 (b). Note that such problem could not be simply fixed by 'Zero Padding' in time domain, as 'Zero Padding' in time domain corresponds to interpolation in frequency domain, which does not increase the resolution.

Decoupled with task-oriented information. Last but not least, as conventional DFT is a universal solution for extracting frequency domain information, and is not designed specifically for certain tasks, the part of the conventional DFT output that we are actually interested in only occupies a fraction of the whole pre-processing output, depending on specific tasks. For example in the Range dimension, in tasks like hand-gesture recognition, the movement is typically within 5cm, and only one or two points in the Range Profile contain the information of the gesture

performed (excluding multipath). Even for activity recognition tasks where the user may perform actions across the longer range (e.g., 50cm), and such action can be captured with tens of data points in Range Profile, it's still only a small portion of the whole Range spectrum. The Doppler and AoA dimensions DFT also have similar limitations. As illustrated in Fig. 6.4 (c), suppose we are doing Range-Doppler pre-processing with conventional DFT, for some specific tasks only a small portion of the pre-processing output contains target information. The non-informative part of the output may contain multipath reflections, noise and environmental reflections. Sending this information into the downstream neural network may lead to much higher computational cost, and even worse, make the neural network learn fake features and overfit to the training data.

It is possible to have a specific DFT for each task, with different FFT lengths, non-uniform frequency spacings or frequency masks such that the domain transformation can focus more on relevant features corresponding to specific tasks. However, this means that we must choose specific parameters for each task, which relies on the experience of the developer and is very time-consuming to tune these hyperparameters.

6.6.2 CubeLearn Module Architecture

To deal with the above-mentioned limitations of conventional DFT pre-processing, in this work we present a learnable pre-processing module to replace the DFT pre-processing, which can be trained together with the downstream classifier in an end-to-end way, and aim to learn the best domain transformation for a specific task in a data-driven manner.

A neural network does not suffer from the limitations of conventional DFT. Through end-to-end training, the network is able to learn the transformation bases that best suit the target task. In addition, a neural network does not have the resolution limitation as in conventional DFT signal processing. In fact, recent research has shown that super-resolution of Range, Doppler and AoA can be achieved through deep neural networks for raw radar signals [39, 40]. Furthermore,

the pre-processing module could be trained to extract more informative features related to the actual physical model of the task, rather than noise and environmental reflections, potentially carrying more useful information to the downstream classifier.

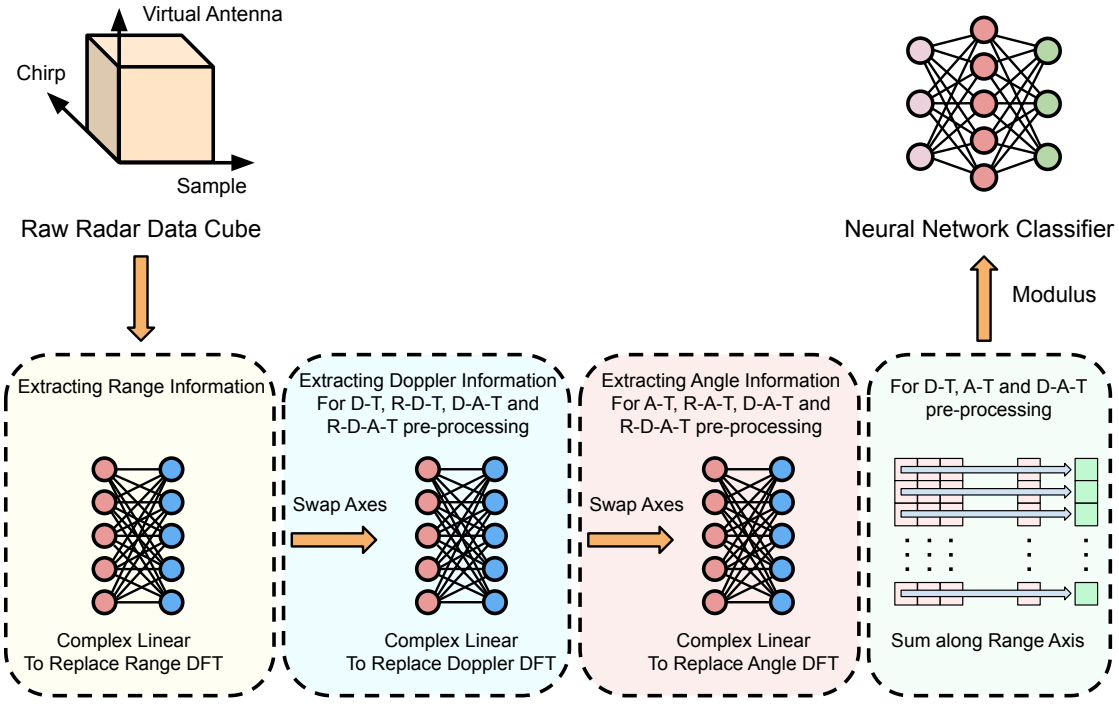


Figure 6.5: Architecture of the proposed CubeLearn module.

The architecture of our proposed CubeLearn module is shown in Fig. 6.5. The basic idea behind the design is to follow the conventional DFT processing method, as introduced in Section 6.3, but in a learnable way. As the raw radar signal is complex-valued, we use complex linear layers to directly consume the raw radar signal as input. Instead of processing all the information in the raw radar data cube in one layer, we use a stacked hierarchy of complex linear layers, each layer replacing a certain step in conventional DFT pre-processing (i.e., Range, Doppler and AoA). This is because if we are to use a single complex layer, there would be too many parameters, making the neural network significantly more complex. For example, if we want to replace R-D pre-processing of size (256×128), with a single complex linear layer we would need $256 \times 128 \times 256 \times 128 \approx 10^9$ complex-valued weights, which is practically unacceptable. However, with two stacked complex linear layers we only need $256 * 256 + 128 * 128 = 81920$ complex-valued weights.

As we are using FMCW radar, the first step is to extract Range information from each chirp with a complex linear layer. This can be followed by other complex linear layers to extract Doppler and AoA information for different ranges. We can also aggregate the Doppler and AoA information at different ranges by calculating the sum along the Range axis. To make the proposed CubeLearn module fairly act as a drop-in for the conventional DFT, we calculate the absolute (modulus) value before feeding the processed data into the downstream neural network classifier. However, this is not essential.

6.6.3 CubeLearn Module Weight Initialization

As conventional DFT pre-processing has been widely adopted and demonstrated universality across different tasks, we use the DFT pre-processing as prior knowledge to initialize our proposed CubeLearn module, and let the neural network adjust the weights to adapt to a specific task through end-to-end training. A conventional DFT of size N applied on the input of size M can be represented as

$$\begin{cases} \hat{x}[k] = \sum_{m=0}^{M-1} e^{-j\frac{2\pi}{N}km} x[m], & M \leq N \\ \hat{x}[k] = \sum_{m=0}^{N-1} e^{-j\frac{2\pi}{N}km} x[m], & M > N \end{cases} \tag{6.1}$$

where x is the input vector, $k = 1, 2, ..., N$, and j is $\sqrt{-1}$.

DFT is a linear transformation from time domain to frequency domain, which can be represented by a complex linear layer with no bias in a neural network. For an input of size M and an output of size N ($N \geq M$), the weights of a complex linear layer can be represented as a N by M matrix as

$$w = \begin{bmatrix} w(0,0) & w(0,1) & ... & w(0,M-1) \\ w(1,0) & w(1,1) & ... & w(1,M-1) \\ ... & ... & ... & ... \\ w(N-1,0) & w(N-1,1) & ... & w(N-1,M-1) \end{bmatrix} \tag{6.2}$$

The value of the k^{th} output neuron would be

$$output[k] = \sum_{m=0}^{M-1} w(k,m) input[m] \tag{6.3}$$

as a result, we initialize the weights as

$$w(a,b) = e^{-j\frac{2\pi}{N}ab} \tag{6.4}$$

then we have

$$output[k] = \sum_{m=0}^{M-1} e^{-j\frac{2\pi}{N}km} input[m] \tag{6.5}$$

which exactly represents DFT. The bias of the complex linear layers are set to 0.

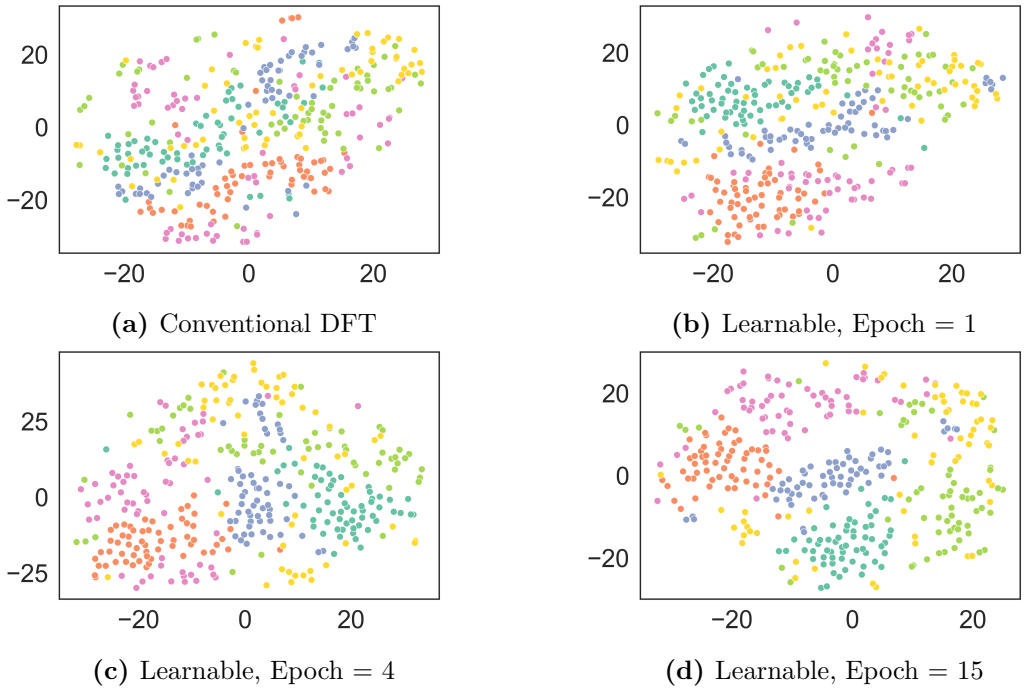
For Doppler and AoA transformation layers we also need to shift the upper half and the lower half of the weights, since we want zero speed and zero AoA (i.e. in front of the sensor) initially to be in the middle of the learned heatmap. This is because for continuous movements, for example from negative radial speed to positive radial speed, we want the reflection in the heatmap to be continuous as well, rather than jumping from one end to the other.

Note that though we initialize the complex layers as DFT, through end-to-end training the transformation is no longer identical to a Fourier Transform once trained.

6.6.4 Effect of CubeLearn Module

To better demonstrate the effects of our proposed CubeLearn module compared to the conventional DFT pre-processing and validate our assumptions, in this part we demonstrate how the proposed CubeLearn module adaptively adjust the weights, by visualizing the output of both the conventional DFT pre-processing module and the CubeLearn module through the training process. We use t-SNE, a widely adopted method for dimension reduction and data visualization [277]. Without loss of generality, we use D-T pre-processing + 2DCNN classifier model on Hand Gesture Recognition task here as an example. Detailed definition of the task as well as the description of the data is introduced later in Section 6.7. There are 12 hand gestures in total, however, to make the figures more clear, we only visualize the pre-processed output of 6 gestures.

In Fig. 6.6, we first visualize the output of the conventional DFT pre-processing in Fig. 6.6 (a), then we visualize the change of the output of the proposed learnable D-T pre-processing module in Fig. 6.6 (b)-(d). From the Fig. 6.6 (a) we can clearly see that, although in local regions, the output from certain classes appears to be denser, indicating that the conventional DFT extracts some common properties



(a) Conventional DFT **(b)** Learnable, Epoch = 1

(c) Learnable, Epoch = 4 **(d)** Learnable, Epoch = 15

Figure 6.6: t-SNE visualization of the conventional DFT pre-processed data and the learnable pre-processed data at different training epochs. Different colors represent different classes. Best viewed in color.

of the same class samples, on the whole, the features from different classes still tend to mix together. This shows that the conventional DFT pre-processing is not ideal for effectively extracting features from the raw radar signal. With the output from conventional DFT pre-processing, it could be quite challenging for the downstream classifier to produce accurate predictions.

On the other hand, we can see from Fig. 6.6 (b)-(d), with the use of the CubeLearn module, the extracted features of different classes become more distinguishable as the training proceeds. This shows that during end-to-end training, the optimizer is able to adjust the weights in the CubeLearn module to produce more informative outputs. As the outputs of the pre-processing module become more distinguishable, it greatly reduces the burden of the downstream classifier, and makes the task of achieving a higher accuracy simpler, which satisfies our requirement R1. In later sections, we provide further evaluation and discussion of the strengths and limitations of the proposed CubeLearn module.

6.7 Experiment Setup

To evaluate our proposed CubeLearn module, we conduct extensive experiments on three most commonly seen interaction tasks with mmWave radar, including hand gesture recognition, arm gesture recognition and human activity recognition. As there is no existing public dataset that contains raw complex radar data available for the above tasks, we collected our own dataset for evaluation. We try to follow existing publications on experiment settings and gesture/activity set designs. To better understand the evaluation results, we provide the design of tasks, hardware settings, and software implementation details in this section.

6.7.1 Experiment Design

In this part we introduce our design of the hand gesture set, arm gesture set and activity set in our experiment. Fundamentally hand gesture recognition is the most difficult task due to the subtle finger movements and occlusion of digits. The detailed designs are as follows:

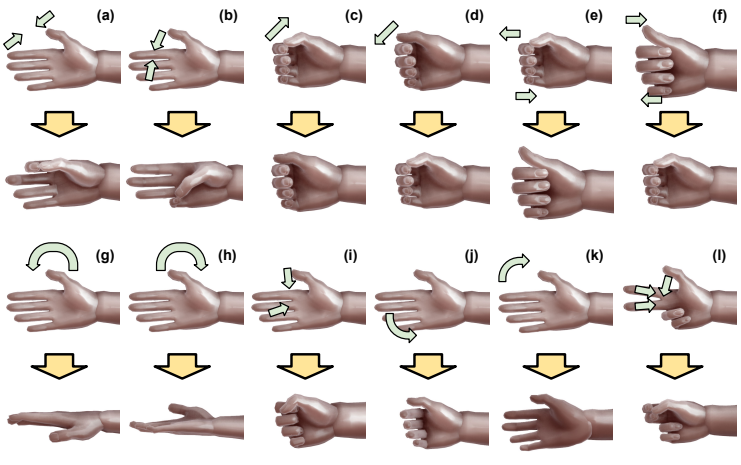


Figure 6.7: Hand Gesture Set Design.

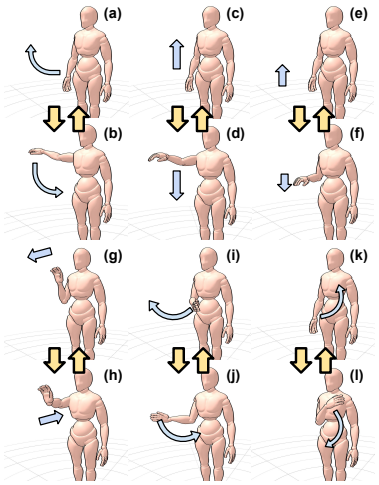


Figure 6.8: Arm Gesture Set Design.

6.7.1.1 Task 1: Hand Gesture Recognition (HGR)

For hand gesture recognition, we follow the gesture set design of several previous works [95, 126, 266] and define our gesture set used in this work. Our gesture set

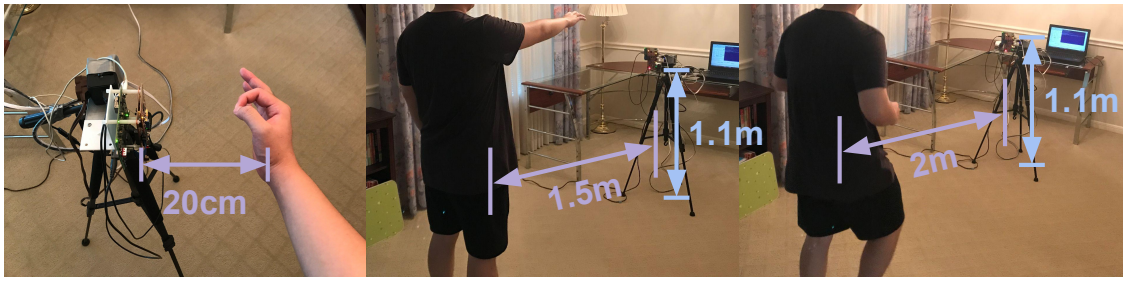


Figure 6.9: Experiment Settings.

design is shown in Fig. 6.7. The hand gesture set contains: (a) Pinch Index; (b) Pinch Pinky; (c) Slide Right; (d) Slide Left; (e) Rub Forward; (f) Rub Backward; (g) Rotate Left; (h) Rotate Right; (i) Fist; (j) Swipe Left; (k) Swipe Right; (l) Zoom Out. The palm is facing the sensor at the beginning of each gesture, and the gestures are based on the right hand and are performed 20cm away in front of the sensor, as shown in Fig. 6.9 (a).

Note that our hand gesture set design is much more difficult than the hand gesture sets in most of the previous works on hand gesture recognition with mmWave radar, as our gesture set only involves hand movement, mostly fingers, and does not include forearm movement.

6.7.1.2 Task 2: Arm Gesture Recognition (AGR)

Arm gestures are typically used in longer distances, e.g., several meters, such as controlling the TV from the sofa, or interacting with smart home devices. We design 12 commonly used gestures based on several previous works on radar arm gesture recognition [20, 97], as shown in Fig. 6.8. The gestures only involve right arm gestures and are performed 1.5m away in front of the radar, as shown in Fig. 6.9 (b). The arm gesture set contains: (a) Side Lift; (b) Side Down; (c) Front Lift; (d) Front Down; (e) Forearm Lift; (f) Forearm Down; (g) Push; (h) Pull; (i) Swipe Right; (j) Swipe Left; (g) Diagonal Lift; (h) Diagonal Down.

6.7.1.3 Task 3: Human Activity Recognition (HAR)

Another commonly seen application is human activity recognition (HAR). Different from HGR and AGR, HAR usually involves continuous movements, which means the motions do not have a certain 'start time' and 'end time'. Instead of trying to identify each start time as we do in gesture recognition, we simply cut the data into equal-length samples. We also follow the design in previous works [44, 127, 192], excluding those we believe that are not appropriate for participants to perform (e.g., crawling) or those that are too physically demanding (e.g., jumping) for typical smart-home applications. We include the following 6 activities in our dataset: (a) Marching on the spot; (b) Jogging on the spot; (c) Clapping; (d) Waving right hand; (e) Sweeping the floor with the right hand; (e) Rubbing left arm with right hand. The activities are performed 2m away in front of the radar in our data collection, as shown in Fig. 6.9 (c).

6.7.2 Data Collection Setup

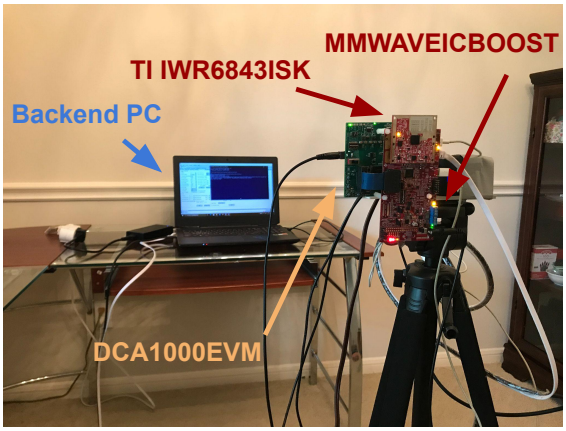


Figure 6.10: Sensor Setup.

In this paper we use a commercial-off-the-shelf Texas Instruments IWR6843ISK mmWave sensor carried by MMWAVEICBOOST, and DCA1000EVM for raw data streaming. The sensor is mounted on a tripod, with height of around 110cm, as shown in Fig. 6.10. mmWave radar configuration is shown in Table 6.3. For motion recognition tasks like gesture recognition and activity recognition, we would want

to have a finer range resolution to capture more appearance information. As a result, we use a bandwidth of 3.6GHz here. Besides, to capture subtle movements, we also prefer a finer Doppler resolution, so we configure the radar to send 128 chirp loops per frame. Smaller bandwidth or fewer chirps per frame decreases the resolution, and can be sub-optimal for this task..

Configuration	Value
Starting Frequency (GHz)	60.25
Frequency Slope (GHz/ms)	60
Bandwidth (GHz)	3.6
Sampling Rate (ksps)	10000
Samples per Chirp	256
Chirp Loops	128
Frame Period (ms)	100

Table 6.3: Sensor Configurations.

Raw radar data is streamed and stored in PC continuously. For hand gesture and arm gesture recognition, the starting time of each action is logged and the data corresponding to each action is later extracted from the raw data file according to the starting time with length of 1 second (10 frames).

We invite 8 participants for data collection[1], including 4 males and 4 females, aged 26 to 55, with height approximately from 160cm to 175cm, and bodyweight approximately from 45kg to 90kg. The background environment of data collection could vary across participants, but the experiment settings are the same. The data from the 8 participants are divided into two parts: 6 people are used as 'in-set' users, which mean their data is used for both training and 'in-set' testing; the remaining 2 people are 'out-of-set' users, whose data is used to test the generalization abilities of the trained model.

6.7.3 Neural Network Implementation

We implement the neural networks involved in this work with PyTorch. The complex layers are implemented based on an open-source Python module called CplxModule[2].

[1]The study has received ethical approval *CS_C1A_021_018*

[2]`https://github.com/ivannz/cplxmodule`

For Range and Doppler transformations, we configure the output size of the linear layer to be the same as the input size. We only use azimuth AoA information in this work, and for extracting AoA information, we configured the output size to 64 instead of the input size 8, so that the pre-processed data can be better fed into the convolutional layer for further feature extraction. For D-A-T and R-D-A-T pre-processing, we use the first half of the raw data on sample and chirp axes because of GPU memory limitation. This does not lose much information on Range and Doppler dimensions, as we will discuss in Section 6.11.

For the implementation of the neural network classifiers, we use the same settings for each model, as shown in Fig. 6.3. The kernel sizes for the convolutional layers are set to 3 and the channels are set to 4, 8, and 16, respectively. We apply max-pooling operation of size 2 on each dimension after each convolutional layer excluding the frame dimension. The LSTM is set to have a hidden size of 512. The output sizes of the three fully-connected layers are set to 512, 128 and 12/6 (depending on the number of classes), respectively.

We use the cross-entropy loss to train the network. The neural networks are trained with Adam optimizer. Each model is trained for 30 epochs, and the best weights are saved according to validation accuracy and loss (accuracy is prioritized than loss).

6.8 Evaluation

In this section we first evaluate the accuracy of DFT pre-processing based pipelines and CubeLearn based pipelines on HGR task, then extend to AGR and HAR tasks. Some of the pipelines with conventional DFT pre-processing are widely adopted by previous works on related topics, e.g., D-T pre-processing + CNN classifier [44, 125–127, 265] and R-D-T pre-processing + CNN-LSTM classifier [43, 131], etc. As a result, they also serve as a comparison to previous related works.

We also evaluate the trained models on the 'out-of-set' users whose data is not included in the training set. Generalization accuracy tends to be lower, as recognition accuracy can be impacted by the shape and size of the user's hand/body,

how the user performs the action, as well as the environmental noise. This situation can certainly be mitigated by collecting more training data with various users and environments, or adopt transfer learning and life-long learning techniques. As our main aim here is to evaluate the generalization ability of our proposed CubeLearn based methods against conventional DFT pre-processing based pipelines, we do not focus on improving the generalization ability of a specific model/task here.

Besides validating the performance of our proposed CubeLearn module, this section also serves as a comparison between pre-processing and classifier combinations on different tasks.

6.8.1 Hand Gesture Recognition Evaluation

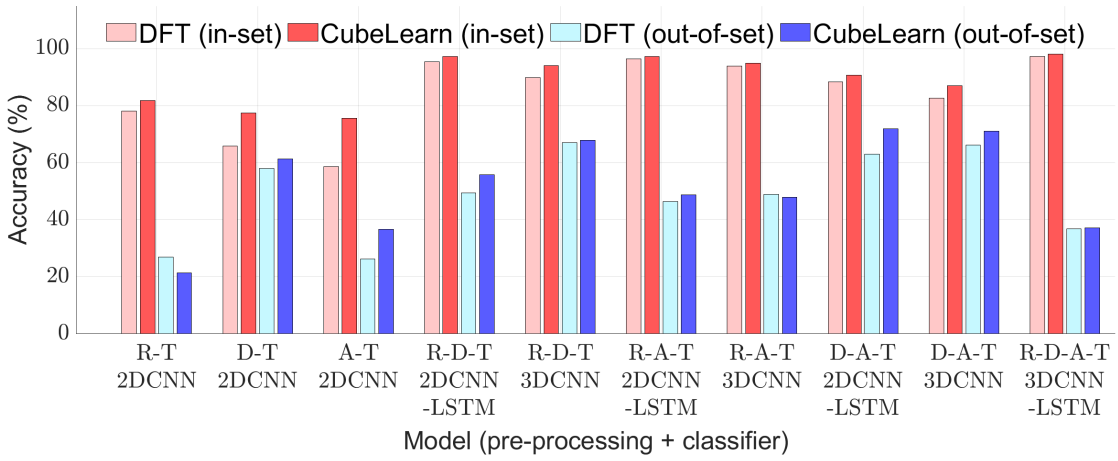


Figure 6.11: Hand Gesture Recognition Accuracy.

We first evaluate our proposed CubeLearn module against conventional DFT on HGR task. The results are shown in Fig. 6.11.

6.8.1.1 Impact of CubeLearn Module

We can see from the figure that, with CubeLearn, the model is able to have consistently better performance than the DFT pre-processing counterparts. This is especially pronounced on the weaker performing DFT models, e.g., R-T pre-processing + 2DCNN classifier, A-T pre-processing + 2DCNN classifier and the widely adopted D-T pre-processing + 2DCNN classifier, with over 10% of accuracy improvement for the latter two models. Even for some of the models that already

have a reasonably high recognition accuracy ($\geq 90\%$) with conventional DFT pre-processing, our proposed CubeLearn module is usually able to further improve attained accuracies by around $1 \sim 5\%$. These results show that our proposed CubeLearn module is a very powerful tool to boost recognition accuracy on motion recognition tasks with mmWave radar.

6.8.1.2 Impact of the Combination of Pre-processing and Classifier

For 'in-set' testing, we generally have better performance with higher dimensional pre-processing. For instance, R-D-A-T pre-processing based model achieves the best testing accuracy of 97.36% with conventional DFT and 98.06% with CubeLearn. And also, two-dimensional pre-processing (R-D-T, R-A-T and D-A-T) have significantly better performance than one-dimensional pre-processing (R-T, D-T, A-T). Higher dimensional pre-processing inherently has more informative features, and the additional information can be utilized to better distinguish between different hand gestures. Also for R-D-T, R-A-T and D-A-T pre-processing, we find that 2DCNN-LSTM classifier behaves better than 3DCNN classifier, probably because the LSTM can better track the temporal relationships between frames in a gesture sample through the explicit latent feature space.

6.8.1.3 Generalization to Out-of-set Users

When encountering out-of-set users, our proposed CubeLearn can still achieve better performances than DFT pre-processing for the majority of the models.

Having a closer investigation we can observe that the CubeLearn tends to further increase the generalization ability on more generalizable models, as these models capture more features that represents the underlying physical model, rather than some special properties of the training data. For example, in D-A-T + 2DCNN-LSTM classifier, with DFT pre-processing module the model is able to have around 63% of testing accuracy on 'out-of-set' subjects, and the proposed CubeLearn module can improve the accuracy to around 72%.

Comparing across different models, we find that the models with Doppler information have better generalization abilities. This is because Doppler information is invariant to the location where the gesture is performed, as long as the gesture or activity is oriented towards the radar. As CubeLearn can extract more distinguishable Doppler features through a learnable complex linear layer, the generalization abilities of Doppler-based models tend to further improve compared to conventional DFT pre-processing counterparts.

However, with the strong feature extracting ability of the CubeLearn module, some specific characteristics of the training samples could also be extracted and carried over, which sometimes makes it harder for the model to generalize to previous unseen subjects. For example for R-T pre-processing + 2DCNN classifier, although CubeLearn can attain better test performance with 'in-set' users, it generalizes less well than conventional DFT pre-processing, as the Range feature itself is inherently not able to generalize well on 'out-of-set' subjects, as the conventional DFT based pre-processing model achieving less than 30% accuracy. Luckily when solving real-world problems, we would most likely pick models that use features with strong generalization ability, and in these cases, CubeLearn module could also be beneficial to the testing accuracy on 'out-of-set' users.

With respect to radar data cube slicing, differently from 'in-set' testing, higher dimensional pre-processing does not always lead to better generalization performance. For example, the D-A-T + 3DCNN model has better generalization ability than R-D-A-T + 3DCNN-LSTM model, also because that the R-D-A-T features overfit to the training subjects. This could also be a function of the training set size - a dataset of hundreds of users would likely generalize well, even for higher dimensional models.

6.8.2 Generalization to AGR and HAR Task

We further evaluate our proposed method against conventional DFT on the other two tasks, and the results are shown in Fig. 6.12 and Fig. 6.13. AGR task is similar to HGR task, where each gesture is represented by a sequence of frames, but it is much easier as the movements are bigger which makes them more obvious to

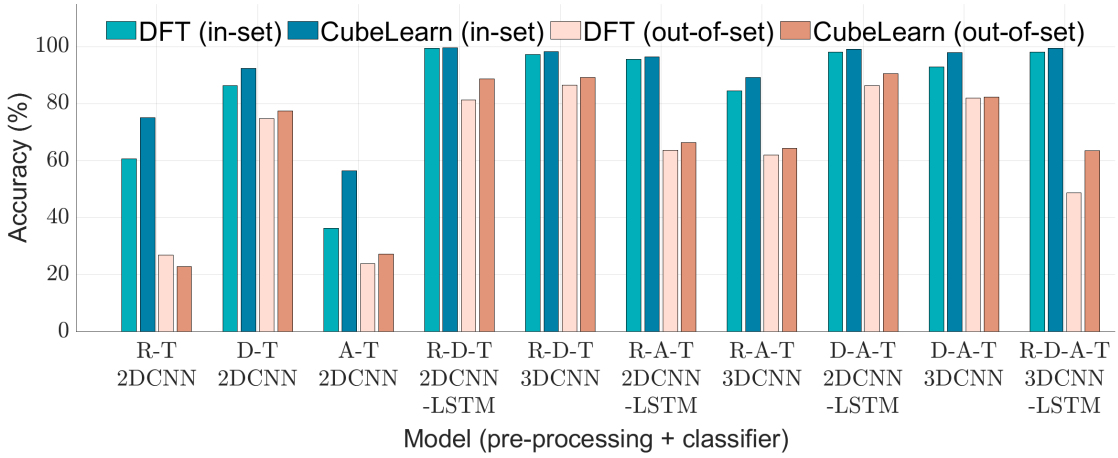


Figure 6.12: Arm Gesture Recognition Accuracy.

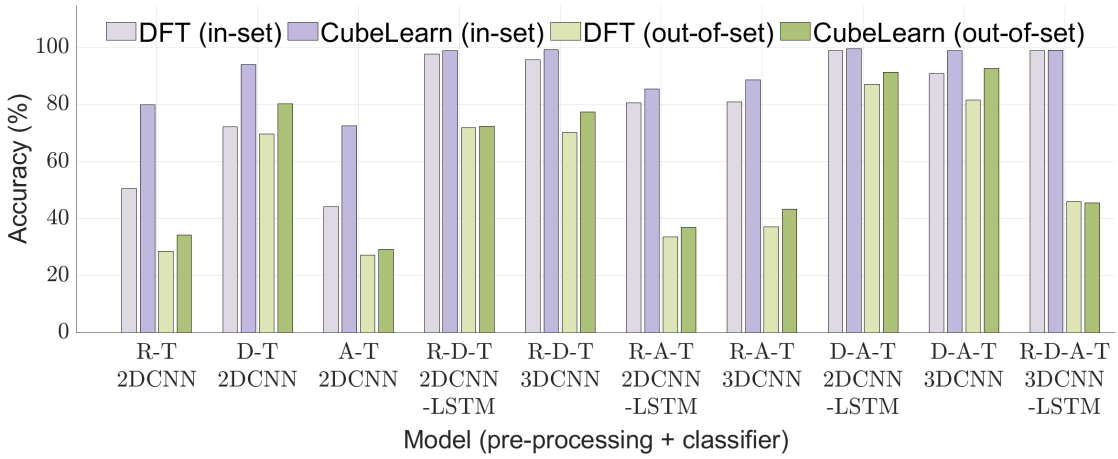


Figure 6.13: Human Activity Recognition Accuracy.

mmWave radar. HAR is fundamentally different as it mainly involves recognizing continuous and repetitive movements.

From the 'in-set' testing result we can see that, the model performances on AGR and HAR are generally better than HGR task. Still, our CubeLearn module can consistently improve the performance of different pipelines, by extracting the most informative features from the raw radar signal cube. The proposed module can be extremely beneficial sometimes, e.g., increasing the accuracies of some of the simpler models (R-T, D-T, A-T) by almost 15% to 20%. This is an important finding, as R-T and D-T are the most basic signals which all FMCW radars will support, regardless of the number of antennas, and they are more computational efficient for edge devices. This means that a simpler radar (with respect to hardware

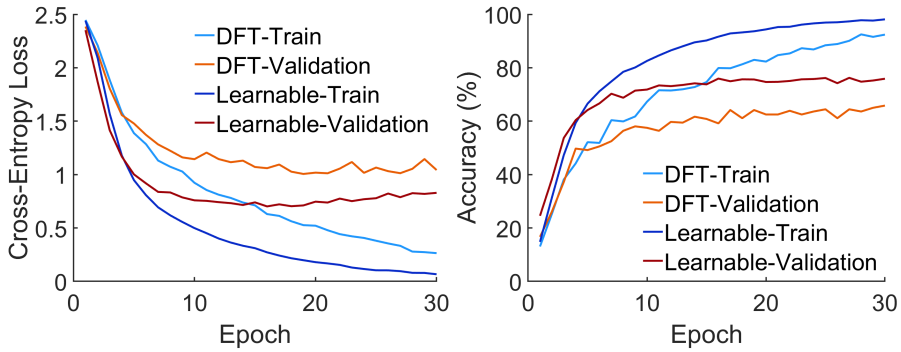
configuration) could achieve the performance of a much more sophisticated device, merely through the use of CubeLearn. The benefit of the proposed pre-processing method on different tasks shows that our design satisfies the requirement R2.

Different from HGR case, in HAR task we find using LSTM in the downstream classifier does not always yield better performance. For example when using R-A-T pre-processing, the 3DCNN classifier has better performance than the 2DCNN-LSTM classifier. This suggests that the LSTM might be better at recognizing gestures than continuous motion, and for continuous motion recognition, both classifiers are worth trying.

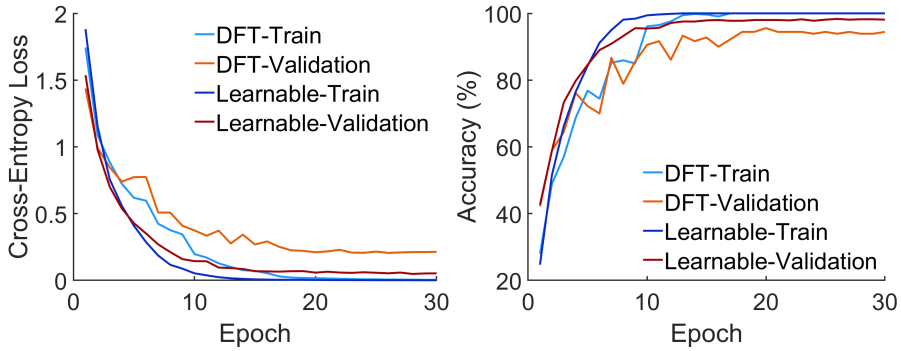
We find that the accuracies for 'out-of-set' test are also higher as the tasks themselves are less challenging to generalize due to their more pronounced motion. The accuracies of some models on 'out-of-set' subjects are comparable to the accuracies on 'in-set' subjects, such D-T + 2DCNN in HAR task. By analyzing the 'out-of-set' test results, we can arrive at a similar conclusion as in the HGR task, i.e., our proposed CubeLearn Module can be beneficial for most models, and Doppler features are more important for motion recognition than Range or AoA features.

6.8.3 Convergence Analysis

We further compare the convergence speed of methods based on DFT pre-processing and CubeLearn module. We visualize the training process, including the training accuracy/loss and the validation accuracy/loss, for D-T pre-processing + 2DCNN classifier on HGR task and R-D-T pre-processing + 3DCNN classifier on HAR task in Fig. 6.14. In both cases, the model with CubeLearn module is able to converge in fewer epochs, typically within 15 epochs, with loss decreasing and accuracy improving much faster. Furthermore, from Fig. 6.14 (b) we can see that, though the models converge to approximately the same training loss and accuracy, completely fitting the training set, there is a constant validation loss/accuracy difference between the models, which shows that the network with the CubeLearn module can actually better learn the physical model behind the task. Note that this shows that the superior performance of the CubeLearn module is not merely because



(a) HGR: D-T + 2DCNN



(b) HAR: R-D-T + 3DCNN

Figure 6.14: Visualization of the training processes.

of more trainable parameters, as in this case CubeLearn module based model and conventional DFT based model both completely fit the training data, while the CubeLearn based model still yields better performance on the validation/testing set.

6.9 Ablation Study

6.9.1 Impact of CubeLearn Module Learning Rate

The learning rate of the CubeLearn module controls the speed at which the complex layer weights vary, which can be set differently from the learning rate of the neural network classifier. With a higher CubeLearn module learning rate, the weights of the complex linear layers change faster during training - the coupling between these the pre-processing and classification networks could cause the system as a whole to converge to a sub-optimal set of weights. To find the optimal learning rate for the complex layers (or the ratio between the learning rates of the CubeLearn module and the classifier), in this part we study the impact of the CubeLearn

module learning rate on the model overall accuracy. We use the D-T pre-processing + 2DCNN classifier model on HGR task here for demonstration. The learning rate of the neural network classifier is set to 0.0003 for all the cases. Each model is run multiple times. The result is shown in Fig. 6.15.

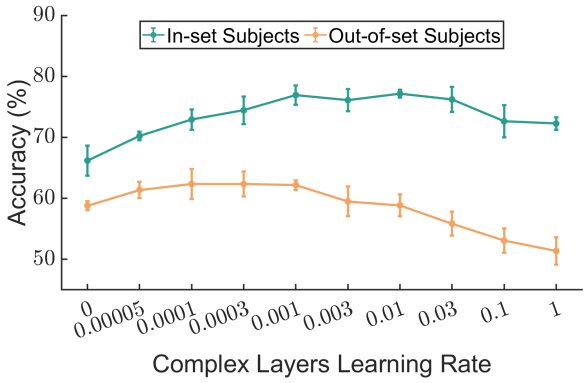


Figure 6.15: 'In-set' test accuracy and 'out-of-set' test accuracy with different CubeLearn module learning rate. The learning rate of the neural network classifier is fixed at 0.0003.

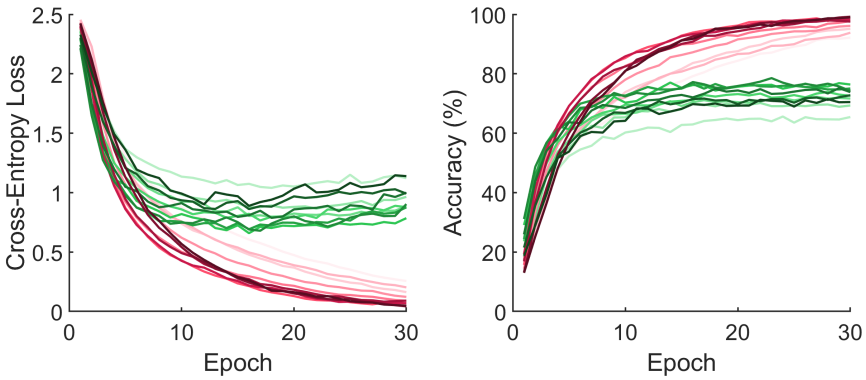


Figure 6.16: Loss and Accuracy curve. Left: Training Loss v.s. Validation Loss. Right: Training Accuracy v.s. Validation Accuracy. Red color represents training set, and green color represents validation set. The darker the color, the higher the learning rate. The learning rates are the same as in Fig. 6.15.

From the results we can see that, both 'in-set' and 'out-of-set' testing, with the increase of the CubeLearn module learning rate, the model accuracy first increases, then decreases. Note that when the learning rate is 0, the model degenerates to a conventional fixed DFT pre-processing + 2DCNN classifier model. With the learning rate from 0.00005 to 0.0001, the optimizer is able to adjust the weights of the CubeLearn module to achieve better model accuracy, for 'in-set' and 'out-of-set'

subjects test. However, as we discussed in Section 6.6, the CubeLearn module could extract features other than those that represent the physical model. As a result, with the learning rate further increasing, the model is likely to overfit to the training subjects, and even to the training samples. As shown in the figure, the generalization accuracy begins to decrease when the learning rate of the CubeLearn module is larger than 0.001, which shows that the model is overfitted to the training subjects. The 'in-set' test accuracy is able to be maintained at a relatively high level until the learning rate reaches 0.03. When the learning rate is larger than 0.03, the model further overfits to the training samples, leading to an accuracy degradation on the 'in-set' test as well.

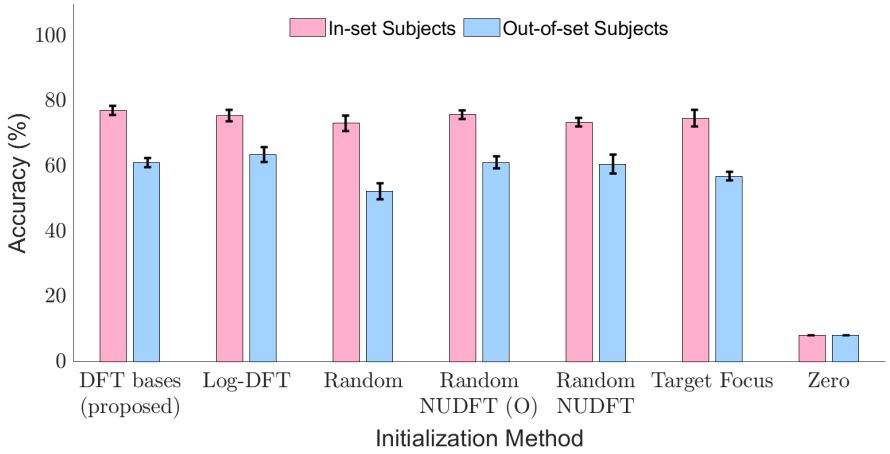
Fig. 6.16 shows the training and validation loss and accuracy curve of the models with different CubeLearn module learning rates. Darker color represents a higher learning rate. We can see that for training set, with a higher learning rate the model is able to converge at a smaller training loss, and larger training accuracy. However, a suitable learning rate could make the model converge faster. For validation set, either a learning rate that is too small or too large yields sub-optimal performance, and the smallest loss and highest accuracy can only be achieved with a proper learning rate of the complex layers, which is 0.001 in this case. Note that the optimal learning rate ratio between the CubeLearn module and the neural network classifier can vary for different pre-processing and classifier combinations as well as on different tasks.

6.9.2 Impact of Weight Initialization Method

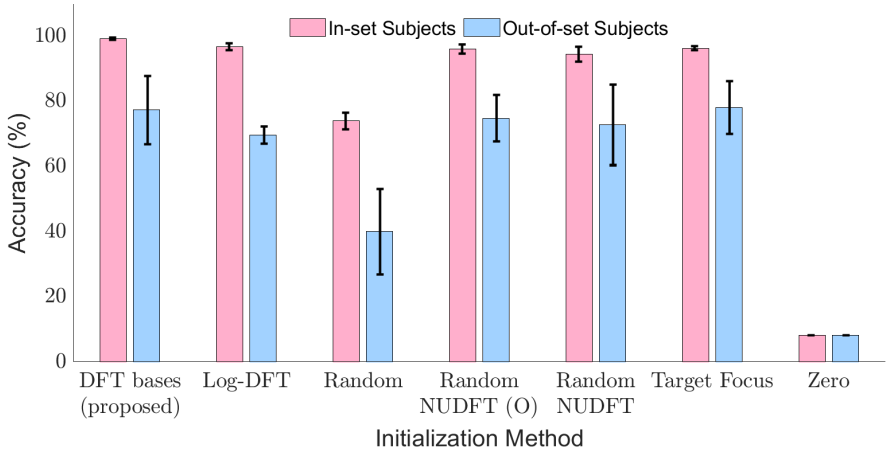
The initialization of the weights in the stacked complex linear layers affects the model accuracy as well as convergence. In this part we try different weight initialization strategies, including:

- Log-DFT initialization, where the Fourier Transform bases are more focused on the low frequency part.
- Default Pytorch Random initialization (He Initialization [278]) for both real and imaginary parts of the complex weights.

- Random Non-uniform Discrete Fourier Transform bases initialization, both organized and unorganized. NUDFT can be seen as sampling in the frequency domain with uneven intervals. Organized here means that the bases are sorted from low-frequency to high frequency.
- Target Focus initialization, where the information (e.g., distance, velocity, AoA) regarding the target is known and the bases of the Fourier initialization are generated close to the target frequency with normal distribution.
- Zero Initialization, where all the weights are set to 0.



(a) D-T pre-processing + 2DCNN classifier



(b) R-D-T pre-processing + 3DCNN classifier

Figure 6.17: Comparison of accuracies on 'in-set' and 'out-of-set' users with different complex layer initialization methods.

We compare the complex linear layers initialization methods with two most common models: D-T pre-processing + 2DCNN classifier on HGR task and R-D-T

pre-processing + 3DCNN classifier on HAR task. The evaluation result is shown in Fig. 6.17. Obviously, the network is not able to be trained with all the weights of the complex layers initialized as 0. Apart from this, the model generally is not very sensitive to the weight initialization method, and is able to converge in each case.

Log-DFT initialization focuses on the low frequency part, which produces slightly better generalization accuracy for the HGR task, because in the Range domain, the high frequencies correspond to distant targets, which is the background environment in our case. However, we observe a slight performance downgrade for the HAR task. This is because the user is 2 meters away from the sensor in the HAR task, which is approximately in the middle of the frequency spectrum rather than low-frequency part. With the Log-DFT initialization focusing on the low-frequency part, there are in fact fewer bases initially focus on the target. As a result, Log-DFT might be beneficial for recognizing targets that are close to the sensor, however, this type of initialization needs careful design.

We find that the network is able to converge with the default random initialization (He Initialization), but with worse 'in-set' and 'out-of-set' accuracy. Especially for the HAR case, the accuracy drops to below 80% and the 'out-of-set' generalization accuracy even drops to below 40%. For HGR the 'out-of-set' generalization accuracy also drops significantly. Using random initialization basically means giving up the prior knowledge about radar signal processing, which is not desirable, especially when the number of samples in the training set is limited.

The organized Random NDUFT initialization is theoretically quite similar to DFT bases initialization, for the bases are sampled with uniform distribution across the frequency spectrum and re-organized from low frequency to high frequency. However, from the result we see that this kind of initialization is not as stable as uniform DFT bases initialization, with slightly worse average performance. Un-organized NUDFT bases initialization produces worse results than the organized one, since it increases the difficulty for the convolutional layers to extract useful features.

Target focus initialization is very hard to use in practice, since it requires prior knowledge of the location, velocity and AoA of the target, and a lot of

engineering effort. Interestingly, we do not observe better performance even with careful parameter tuning.

In conclusion, in practice it is reasonable to simply use DFT bases to initialize the weights of the complex linear layers due to their simplicity, high accuracy and stable performance.

6.9.3 Impact of Network Structure

We further conduct an ablation study of the pre-processing module structure. We also use the D-T pre-processing + 2DCNN classifier model on HGR task and R-D-T pre-processing + 3DCNN classifier model on HAR task here, as they are the most commonly seen models in previous works. Besides, learnable D-T pre-processing can be directly compared to RadarNet [263] and learnable R-D-T pre-processing can be directly compared to 2D Sinc/Wavelet Filters [45]. As mentioned in Section 6.2, RadarNet and 2D Sinc/Wavelet Filters are two previously proposed methods for end-to-end radar signal processing. We compared the performance of our proposed structure to the following variations:

- **Adding Activation between Complex Linear Layers (ModReLU, $\mathbb{C}$ReLU, zReLU)** Activation layers are vital to neural networks as they enable the neural network to learn non-linearities. Here we try adding an activation layer after each complex linear layer in the CubeLearn module. We tested three activation functions: ModReLU [66], $\mathbb{C}$ReLU [65] and zReLU [67]. ModReLU changes the values whose magnitude is below a certain threshold to 0, while keeps other values including their phase information, and can be expressed as

$$\sigma_{modReLU}(z) = \begin{cases} (|z| + b)\frac{z}{|z|} & \text{if} |z| + b \geq 0 \\ 0 & \text{if} |z| + b < 0 \end{cases} \tag{6.6}$$

  where b is the learnable threshold. $\mathbb{C}$ReLU applies ReLU activation on real and imaginary part respectively,

$$\sigma_{\mathbb{C}\text{ReLU}}(z) = \text{ReLU}(\text{Real}(z)) + i\text{ReLU}(\text{Imag}(z)) \tag{6.7}$$

zReLU only keeps the value of those in the first quadrant (i.e., both real part and imaginary part are non-negative), and set other values to 0,

$$z\text{ReLU}(z) = \begin{cases} z & \text{if } 0 \leq \theta_z \leq \frac{\pi}{2} \\ 0 & \text{otherwise} \end{cases} \tag{6.8}$$

where θ_z is the angle of the complex value.

- **Learnable Non-uniform Discrete Fourier Transform (NUDFT)** Although in our proposed method, we initialize the weights of the complex layers as the bases for Discrete Fourier Transform, it is very likely that the linear transformation could not be explained as Fourier Transform when the network is fully trained. This is because the weights of the complex linear layers have more degree of freedom than Fourier Transform restrictions. Here we test constraining the weights of the complex linear layers to Fourier Transform bases so that the network actually learns a Non-uniform Discrete Fourier Transform.

- **Log after Modulus (Log)** In some previous works, log operation is applied after DFT pre-processing to represent the heatmap as a power spectrum, before feeding the data into the neural network classifier [60, 263]. So we tested the performance with the log operation after the CubeLearn module.

- **Clutter Removal (CR)** In the conventional radar processing chain, sometimes the chirp average is subtracted from the data slice of each chirp, to remove the information of static clutters from the radar signal. In our case, we can also apply clutter removal on the raw radar data before sending the data into the CubeLearn module.

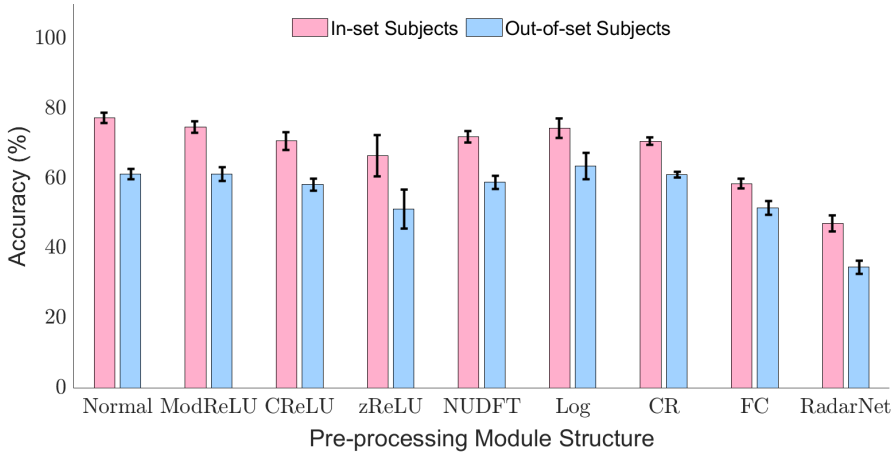
- **Fully-Complex Network (FC)** The neural network classifier can be replaced with corresponding complex counterparts. Here we compare the model to a fully-complex neural network. In the fully complex model, the modulus operation after the CubeLearn is removed, so the complex values are kept. The convolutional layers and linear layers use complex-valued weights instead

of real-valued weights, and the ReLU activation is changed to ModReLU. Modulus operation is added after the last linear layer before the log-softmax activation.

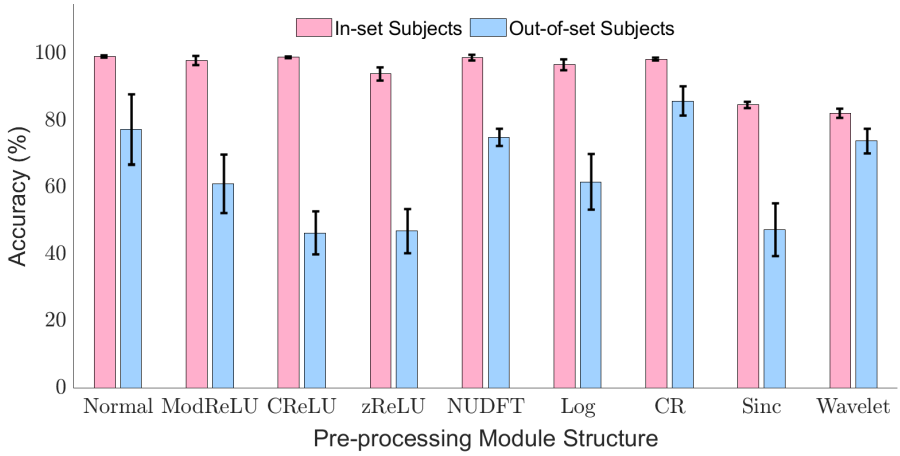
- **RadarNet** Besides possible modifications to our proposed complex linear layers, we also compare our method to RadarNet. As RadarNet is designed for CW radar, which can not be directly applied to our data collected with FMCW radar, we first do Range DFT on the raw radar data, and then apply the module introduced in RadarNet for Doppler estimation on each Range bin, then sum along the Range axis. Furthermore, the authors suggested in their paper to use tanh activation to replace log operation. However, in our implementation the tanh activation results in the training not being able to converge, so we use log operation instead. To have a fair comparison, we use the same downstream 2DCNN classifier structure.

- **2D Sinc Filters and 2D Wavelet Filters** We compare our CubeLearn module to 2D Sinc Filters and 2D Wavelet Filters proposed in [45]. As this study mainly focuses on the comparison of pre-processing modules, we use the 2D Sinc Filters and 2D Wavelet Filters to replace our proposed learnable R-D-T pre-processing, and use the same downstream 3DCNN classifier structure. As the 2D Sinc Filters and 2D Wavelet Filters takes real-valued data, we only use the real part of the original radar raw data as input. Besides, the radar configuration in this work is different from the configuration in the original paper, as we have the definition of the frame. So in this work the 2D Sinc and Wavelet Filters are applied on the data slice for each frame, and the frame number serves as an additional dimension for the downstream 3DCNN classifier. In practice we find that the 2D Sinc Filters and Wavelet Filters are very sensitive to layer parameters, so we test different sets of parameters and report the best one.

The result is shown in Fig. 6.18. We can see that, adding an activation function between the complex linear layers decreases the model performance, especially



(a) HGR: D-T pre-processing + 2DCNN classifier



(b) HAR: R-D-T pre-processing + 3DCNN classifier

Figure 6.18: Comparison of accuracies on 'in-set' and 'out-of-set' users with different CubeLearn module structures.

the generalization ability. This is probably because in the stacked complex linear layers, extracting the Doppler information through the transformation of the latter complex linear layer relies on the output phase information of the previous complex linear layer. Adding a non-linear activation between the complex linear layers, especially $\mathbb{C}$ReLU and zReLU, could result in the valuable phase information being altered or lost.

NUDFT structure produces a slightly worse result than the proposed CubeLearn module, since the change of the weights is restricted and not as flexible. As for adding the log operation after the CubeLearn module, we do not observe performance improvement. In fact, for the HAR case, adding the log operation downgrades

the generalization ability, probably because the difference of peak magnitude and average magnitude is smaller, making the learned features less distinguishable.

Clutter removal operation is actually a double-edged sword. On one hand, it removes the background, and make the model learn properties of the target, which increases the model generalization ability, as we can see from the HAR case. On the other hand, it actually cancels out all the static information, including the information of the target we are interested in. As a result, for HGR task where the hand is very close to the sensor and occupying a larger field of view, removing the static information actually slightly decreases the model performance. Besides, if we are to recognize the static target, e.g., in static gesture recognition, clutter removal will certainly not work. As a result, whether adding clutter removal is beneficial or not depends on the task. For tasks to recognize moving targets that are relatively further away, adding clutter removal could increase the model generalization ability. While recognizing partial static or static targets, especially in cases where the target occupies a large field of view and the reflection from the environment has limited impact, adding clutter removal could decrease the performance.

We also see that using the processing module in RadarNet lowers the accuracy. This is because RadarNet only consumes the real input, so it loses half of the information, which leads to a worse accuracy. 2D Sinc Filters and 2D Wavelet Filters proposed in [45] also takes real-valued input, so the accuracy is not as good as other structures. In addition, although 2D Sinc Filters and 2D Wavelet Filters have fewer trainable parameters, they actually take up more memory during computation, as they generate a large number of filters during inference. We find that compared to our proposed CubeLearn module with complex linear layers, 2D Sinc Filters and 2D Wavelet Filters are very easy to become overfitted to the training samples, and are very sensitive to the layer parameters. For 'out-of-set' test, we observe that 2D Wavelet Filters have competent generalization abilities, much higher than 2D Sinc Filters.

In conclusion, compared to other variations, our proposed stacked complex linear layers achieve good and stable performance. Adding clutter removal could

possibly improve the generalization abilities of the model in some tasks, but needs to be used with caution.

6.10 Complexity and Running Time Analysis

The Discrete Fourier Transform is usually implemented with Fast Fourier Transform algorithm, which is very computationally efficient. Using the stacked complex linear layers introduces more parameters in the network, which will result in larger complexity and longer inference time. In this part we analyze the number of parameters introduced by our proposed CubeLearn module, and compare the running time against classical hybrid baselines on PC and Raspberry Pi. The experiment PC we use features an AMD Ryzen 3800X CPU with 64GB memory, and Nvidia RTX2080Ti graphics card, and the Raspberry Pi we use is Raspberry Pi 4 with 8GB main memory. For DFT based pre-processing, we use 'torch.fft' in 'pytorch' package for possible GPU acceleration when testing on PC, and use 'scipy.fftpack.fft' in 'scipy' package on Raspberry Pi since 'torch.fft' is not supported on ARM architecture. The structures of the neural networks are discussed in Section 6.7.3. Note that the running time reported in this section is based on models for HGR/AGR task.

Pre-process	Classifier	Parameters		
		Classifier	complex linear	↗ (%)
R-T	2DCNN	1085092	65536	6.04
D-T	2DCNN	560804	81920	14.61
A-T	2DCNN	298660	66048	22.11
R-D-T	2DCNN + LSTM	16358564	81920	0.5
	3DCNN	15309356	81920	0.54
R-A-T	2DCNN+LSTM	8232100	66048	0.8
	3DCNN	7182892	66048	0.92
D-A-T	2DCNN+LSTM	4562084	82432	1.81
	3DCNN	3512876	82432	2.35
R-D-A-T	3DCNN+LSTM	3483692	82432	2.37

Table 6.4: The increase of number of parameters with CubeLearn module. '↗' shows percentage of the increase. The parameter number of complex linear layers is the number of complex weights. Results based on models for hand/arm gesture recognition.

The total number of parameters with conventional DFT pre-processing and CubeLearn for each model is summarized in Table 6.4. We can see that replacing

Pre-process	Classifier	PC Time (ms)			Raspberry Pi Time (ms)		
		DFT	learnable	↗ (%)	DFT	learnable	↗ (%)
R-T	2DCNN	0.99	1.25	26.10	15.00	35.71	138.10
D-T	2DCNN	1.05	1.65	56.25	77.41	484.13	525.42
A-T	2DCNN	1.08	1.61	49.85	39.51	165.32	318.41
R-D-T	2DCNN + LSTM	1.63	2.27	38.76	625.72	1018.70	62.80
	3DCNN	1.14	1.75	53.36	419.31	745.10	77.70
R-A-T	2DCNN+LSTM	1.62	2.21	36.13	308.38	528.02	71.22
	3DCNN	1.15	1.69	46.54	222.17	357.00	60.68
D-A-T	2DCNN+LSTM	3.18	5.29	66.00	1486.26	2456.30	65.27
	3DCNN	2.92	5.02	71.80	1626.13	2513.14	54.55
R-D-A-T	3DCNN+LSTM	18.73	24.38	30.16	-	-	-

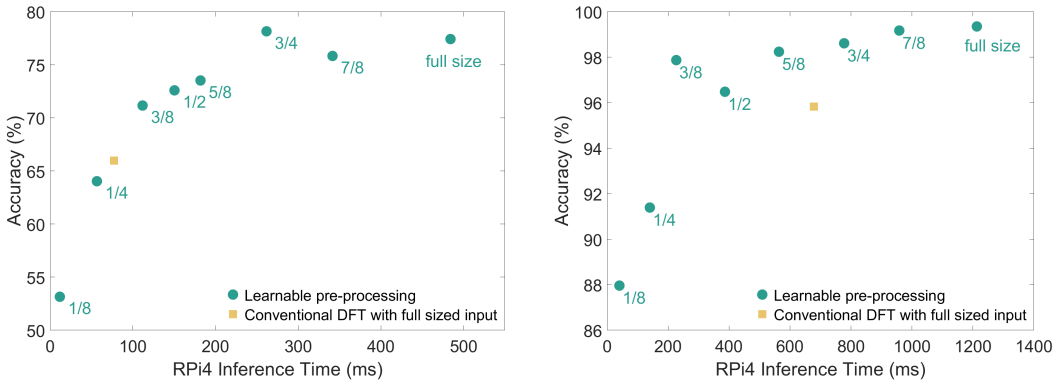
Table 6.5: Running Time on PC and Raspberry Pi. '↗' shows the increase of running time in percentage. The running time is the total time for each model, i.e., DFT pre-processing + classifier and CubeLearn + classifier. '-' means the model is too big to be loaded into the memory for testing. Results based on models for hand/arm gesture recognition.

the DFT pre-processing with our proposed CubeLearn module does not increase the number of model parameters significantly, as for most models, the total parameters of the complex linear layers account for less than 10% of the total parameters. The more complex the model is, the smaller as a proportion of parameters the CubeLearn will introduce.

Table 6.5 shows the running time for each model on PC and Raspberry Pi. Not surprisingly, the computation of complex linear layers is more expensive than FFT, as each layer is actually called multiple times during inference. With GPU acceleration on PC, adding the learnable complex layers would add 47.7% of inference time for one sample on average (SD=15%). When running with the ARM CPU on Raspberry Pi 4, the running time for each sample increases 152.7% on average (SD=163.1%). The computational cost grows exponentially with the number of dimensions of the raw data cube used. For example, with D-A-T or R-D-A-T pre-processing, the neural network has to perform complex linear transformation on all three dimensions, which leads to a large increase in inference time. However, in practice, complex models (e.g., D-A-T and R-D-A-T pre-processing based models) are less likely to be used on edge devices. For simpler models that can run real time on edge devices, CubeLearn could greatly improve the recognition accuracy, as shown in previous sections, with acceptable complexity increase. Furthermore, we can have a smaller

input size with our proposed end-to-end network, to achieve better performance and even smaller running time than DFT baselines, as introduced in the next section.

6.11 Impact of Raw Data Size



(a) HGR: D-T pre-processing + 2DCNN classifier

(b) HAR: R-D-T pre-processing + 3DCNN classifier

Figure 6.19: Accuracy and inference time with different size of raw data input.

The raw radar signal contains repetitive patterns on ADC samples in a chirp, and across chirps in a frame. As a result, lowering the number of samples and chirp number in a frame would possibly still yield a similar level of recognition performance, while making the neural network much more light-weight. In this part we tested the performance of two models with various input sizes: D-T pre-processing + 2DCNN classifier on HGR task and R-D-T pre-processing + 3DCNN classifier on HAR task, as well as the running speed on Raspberry Pi4. Note that the running time of the R-D-T pre-processing + 3DCNN classifier is different from the one reported in Section 6.10 because each sample lasts 2 seconds for HAR task, instead of 1 second for HGR task which we studied previously.

The input here is actually a series of sample-chirp data slices. For both axes, we try using the first 1/8, 1/4, ..., 7/8 of the data points on each dimension as input. As we have a smaller size of the input, the size of the pre-processing module and the neural network classifier is also smaller, which reduces the computational complexity. The result is shown in Fig. 6.19, the text beside each data point

refers the ratio of the samples and chirps we use for each frame. To have a fair comparison, each model is run multiple times.

We can see that the full-sized input actually contains redundant information. With even 3/8 of the original input size on sample and chirp dimension, we are still able to achieve relatively high performance, better than full sized input with conventional DFT pre-processing. For the HAR case, with 3/8, 1/2 or 5/8 size on each dimension of the original input, the inference time on Raspberry Pi 4 of the CubeLearn based model is smaller than conventional DFT pre-processing based model with full-sized input, but achieves better classification accuracy. Especially with 3/8 of size on both dimensions, the accuracy could reach 98% while the inference time is approximately 200ms.

In fact, for specific tasks, we can further reduce the complexity and running time by, e.g., using fewer neurons to represent the transformation output with the initialization of only the Fourier bases corresponding to the distance and velocity of the target. Besides, we can also utilize hardware-level parallel, processing the data of the current frame while receiving the data for the next frame. As a result, we believe that with an optimized implementation as future work, our proposed end-to-end model would execute in real-time on resource constrained devices, with superior performance to the DFT front-end.

6.12 Discussion and Future Work

6.12.1 Model Selection under Different Situations

Situation			Model Selection	
Hardware	Computational Capacity	Task	Pre-processing	Classifier
MIMO FMCW	Very High	HGR/AGR/HAR	Learnable R-D-A-T	3DCNN-LSTM
MIMO/SISO FMCW	Medium	HGR/AGR	Learnable R-D-T	2DCNN-LSTM
MIMO/SISO FMCW	Medium	HAR	Learnable R-D-T	3DCNN or 2DCNN-LSTM
MIMO/SISO FMCW	Low	HGR/AGR	DFT/Learnable D-T	2DCNN
SISO FMCW	High	HGR/AGR	Learnable R-D-T	2DCNN-LSTM
SISO FMCW	High	HAR	Learnable R-D-T	3DCNN or 2DCNN-LSTM
CW radar	High	HGR/AGR/HAR	Learnable D-T	2DCNN
CW radar	Low	HGR/AGR/HAR	DFT D-T	2DCNN

Table 6.6: Model selection under different situations.

We use Table 6.6 to show our model selection recommendations under different situations. As we have discussed in Section 6.8, R-D-A-T pre-processing + 3DCNN-LSTM classifier shows good accuracy on different motion recognition tasks, especially with our proposed CubeLearn module, for information on all three dimensions (Range, Doppler and AoA) are utilized for target classification. However, it mainly has two problems which limit its adoption. The first one is computational complexity, which can be higher especially on edge devices without GPU acceleration (and also D-A-T pre-processing). Another problem is that it requires multiple receivers for AoA estimation, which is not always supported by the hardware itself, so this type of model is best suitable for MIMO radar with abundant computational resource.

Doppler information is the most important for motion recognition. As a result, considering complexity and accuracy on both 'in-set' users and 'out-of-set' users, D-T or R-D-T pre-processing can be good choices, especially for 1Tx-1Rx radar configuration. Besides, D-T pre-processing can be extended to other types of radar as well, e.g., Continuous-wave (CW) radar.

6.12.2 Future Work

The current CubeLearn module still adopts the skeleton of conventional DFT, which leads to relatively high computational complexity. As we can tell from our complexity evaluation, the running time of learnable R-D-A-T pre-processing is unacceptable especially when running on edge devices, even for a relatively simple 3Tx-4Rx MIMO radar, not to mention the massive MIMO radars. An open question is whether there exists a more elegant and efficient universal feature extractor for radar data, e.g., consider PointNet [267] for point clouds. Although we believe our proposed stacked complex linear layers demonstrates how to extract features more efficiently from a raw radar cube, it is clear that this is a rich and open area for future investigation.

The robustness of interaction techniques is also a very important concern. As we find through the experiments, the generalization accuracy is quite sensitive to range/angle features, and relatively robust to Doppler features. How to exploit such characteristics to a mmWave gesture/activity system that produces robust

predictions regardless of the position, orientation and speed of the action is a very important problem to be solved before actual real-world adoption.

Besides, in this work we only validated the performance improvement with our CubeLearn module on classification task. Whether directly extracting information from raw radar data would benefit regression tasks, e.g., tracking and localization, remains an open question to be explored in future work.

6.13 Chapter Summary

In this chapter, we propose a CubeLearn module to replace conventional Discrete Fourier Transform (DFT) pre-processing, and build an end-to-end deep learning frame work for mmWave FMCW radar gesture/activity recognition task, addressing Research Question 4 raised in Chapter 1. We demonstrate that through end-to-end training, our proposed CubeLearn module is able to extract more distinguishable features than conventional DFT pre-processing, which makes it easier for the downstream classifier to make correct predictions and improve the overall model accuracy. We evaluate our proposed method on our own collected dataset of hand gestures, arm gestures and activities, and the results show that for all the tasks, the classification accuracy can be consistently improved, and the training is able to converge in fewer epochs. We envision our proposed method as a small step towards a universal feature extractor for end-to-end deep learning on raw mmWave radar data.

7 Discussion

Contents

7.1 Applying Deep Learning on Different Radar Data Representations 169
7.2 Future Directions 173

7.1 Applying Deep Learning on Different Radar Data Representations

In the conventional FMCW radar processing chain for point cloud generation, we can have different types of radar data representation, as shown in Fig. 7.1. For each step of processing, the radar data representation becomes more straightforward and easier to understand and process, yet with potential information loss. For example, taking the absolute value of range-angle heatmap, we can estimate the target locations in the scene. However, it loses the phase information, which is vital for target velocity estimation. Point clouds show the 3D location of the targets in the environment even more clearly, but it's inherently sparse as non-peak values are filtered out from the heatmaps before generating the point cloud.

It is worth noting that through answering the four Research Questions raised in Chapter 1 in this thesis, we explore applying deep learning technique on different

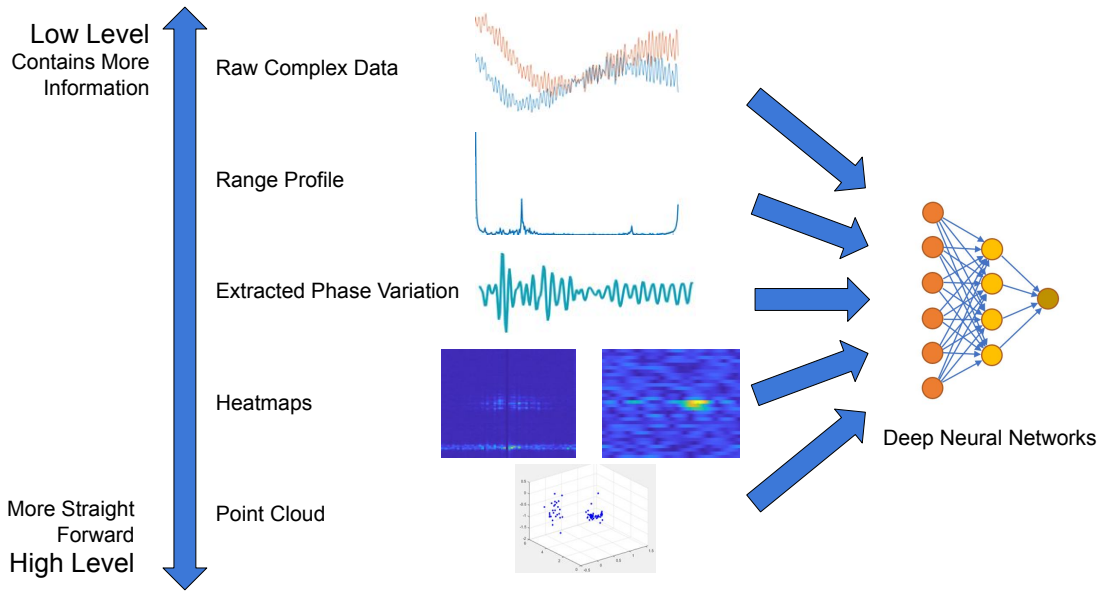


Figure 7.1: Applying deep learning techniques on different radar representations.

levels of radar data. On the lowest level, we present using stacked complex linear layers to directly and adaptively extract information from raw radar data in Chapter 6, which is one of the few works that directly use complex-valued raw FMCW radar data as input of deep neural network. On a higher level, we applied convolutional neural network on the unwrapped phase variation signal of target peak in range profile, for frequency extraction and uncertainty estimation in Chapter 4. Furthermore, we applied deep learning on range-angle heatmaps for target tracking in Chapter 5 and demonstrated superior performance compared to different conventional signal processing pipelines. Also we used voxelized point cloud as deep neural network input for target identification based on the changing patterns of the point cloud cluster in Chapter 3. This thesis demonstrate that deep learning techniques can be applied on different level of radar data representation, to surpass the performance of fully conventional signal processing methods, and can even tackle problems that are very hard or impossible for conventional methods to solve, such as human identification, gesture recognition, and uncertainty estimation.

We can choose what radar data representations to use as deep neural network input based on radar configurations and different target applications. We also

design the neural network to better fit the input data type, and the downstream applications. Here we summarize the common designs of deep neural network on different radar representations, as well as the downstream target applications:

- **Point Cloud** We consider using radar point cloud as input to deep neural network for applications that only involves macro features, such as target location and moving patterns classification. Typical downstream applications include gesture recognition [20], pose regression [87], human identification [91], etc. There are mainly three ways to use radar point cloud as deep neural network inputs. The first way is to voxelize the point cloud and produce 'image-like' data cube as input to deep neural networks, such as the method used in our proposed 'mID' system presented in Chapter 3. This type of method is straight forward, and can generate image-like data to be processed with network structures that have been well-explored in Computer Vision area. However, using voxelized data is relatively slower and consumes more running memory. Another way is to project the point cloud onto certain planes, e.g., X-Y plane and X-Z plane, which can be further processed in similar ways as images [86]. PointNet [267] and RandLA-Net [279] are deep neural network structures proposed in recent years that are specifically designed to extract information from point clouds, and have demonstrated exceptional performance on many tasks like target detection, classification and segmentation. PointNet-style has also been used as feature extractors for mmWave radar point cloud [91], and improved the efficiency of the processing pipeline.

- **Heatmap** For heatmaps, the absolute value are often used as input into neural networks, as most networks only deal with real-valued data. Convolutional neural networks are often applied to extract features from heatmaps, similar to processing images. The extracted features are used for further classification or regression with convolutional layers, recurrent layers and/or fully-connected layers. Similar to point cloud, heatmaps are often used as input for downstream

applications regarding the classification or regression of macro targets and movements. There are different types of radar heatmaps. Range-Doppler heatmaps contains information regarding the distance and velocity of the targets, and are often used for classification of movements, such as gesture recognition. Range-Angle heatmaps contains information about the target distance and angle-of-arrival to the radar, and could be used as input for target localization, such as in our proposed 'mDrone' in Chapter 5.

- **Phase Variation** On a micro level, tiny displacements can be tracked through the phase change of the corresponding peak in the range profile. Often it is used for estimating the frequency of periodic movements, such as heart rate and breathing rate. As introduced in Chapter 4, convolutional neural network can be applied to extract the information from the phase variation signal, and estimate the frequency of the micro-displacement.

- **Range Profile** Range profile can not only be used for target distance estimation, but it also contains information about the target reflection properties. For example, the magnitude of the peak can be used for target material classification. The information of the target is contained not only in the target peak, but also in the adjacent points in the range profile because of multi-path effect. We can use convolutional neural network for target classification based on the magnitude of a certain period in the range profile that we are interested in. For example, in our published work about mmWave radar mapping [12], we use convolutional neural network to classify target materials for semantic mapping with mmWave radar, which showed superior performance than conventional methods.

- **Raw Data** With raw data as input to neural network directly, we can get rid of a hybrid approach, and extract useful features that best represent the underlying physical models directly from raw radar data. This is currently a direction that is being actively explored. The main difficulty of designing end-to-end deep learning method on radar data is that the input data is

complex-valued, and we need to specifically design the neural network to take complex data as input. Some research take only real-valued raw signal as input where half of the information is lost [45, 263], and other network designs for taking the complex data as input includes treating the real and imaginary part as two separate channels [264], as well as our proposed stacked complex layers introduced in Chapter 6. Theoretically, the raw radar data contains all the information, including micro and macro features, so we should be able to build end-to-end deep neural networks for any applications that can be solved with other data representations.

Generally speaking, for different downstream applications, previous arts first determine the radar data representation to use, then design the network according to the pre-processed data format and downstream applications. However, with the development of complex-valued neural networks, we formulate a universal feature extractor which directly extracts features from the raw radar signal, and all we have to do is to design the downstream classifier/regressor.

7.2 Future Directions

The use of mmWave radar has been widely studied in recent years. However, there are a number of limitations impacting applications in our everyday lives. Currently there are still some areas and directions that are worth exploring:

Universal Feature Extractor We present an end-to-end neural network in Chapter 6 and demonstrated that the structure can extract features from raw radar data and solve several downstream tasks. However, we believe there's still a long way to go towards a real highly efficient and universal feature extractor. mmWave radar can be used for a wide variety of tasks, and a universal feature extractor could greatly lower the burden of neural network design for specific downstream applications. We believe that this is a promising research direction.

Reducing Complexity and Running Time The amount of data radar generates can be huge. For Multiple-Input Multiple-Output (MIMO) antenna

array, the data cube generated each frame is the product of transmitter number, receiver number, chirps and samples per chirp, and the raw data size can be up to several megabytes per second. The huge size of raw data would be a great burden for edge devices with limited computational abilities. As a very promising sensor to be embedded into smart devices in the future, how to process the data with high efficiency in real time is something that has to be solved. The idea of compressed sampling enables more sparse sampling of the raw signal while minimize the information loss, and we believe this is something that can be explored further in the future.

Robustness Although researchers have demonstrated various approaches to use mmWave radar for different applications, the robustness of the system is still under exploration. For example, in hand gesture recognition, the users' hand size, hand shape, and their unique habits of performing gestures, will impact the generalization ability of pre-trained models. Besides, the position and orientation of the user's hand will also result in classification performance downgrade. To have truly seamless mid-air gesture recognition system, as well as other actual applications, we need to further investigate how to improve robustness.

Communication and Sensing We are heading towards the era of 5G communication, with millimeter wave as carrier [280]. In the near future we could see all kinds of smart devices and IoT devices with 5G communication abilities coming into our daily lives. These devices could possibly be equipped with MIMO antenna array, which, with small modification, can work as FMCW radar as well. A promising idea would be enabling not only communication abilities, but also sensing abilities for these IoT devices, in this way they can better interact with human subjects without having to install another sensor. In recent years there are already some explorations in combining communication and sensing for a mmWave carrier, and we believe that it would be beneficial to further investigate in this direction.

8 Conclusion

In this thesis, we study applying deep learning techniques for processing mmWave FMCW radar data, for better tracking and recognition performance in the context of smart environments. Extensive research introduced in this thesis show that 1) deep learning techniques have higher accuracy and robustness on tasks like micro and macro tracking compared to conventional signal processing methods; 2) deep learning can extract features from different types of radar data for target recognition tasks, such as human identification, gesture recognition and activity recognition, which could be very challenging for conventional signal processing methods to solve.

In Chapter 3 we introduce 'mID', a novel human tracking and identification system, which is the first to use mmWave radar point cloud for human identification. In the proposed pipeline, we first associate clusters between frames and track the human target, then extract features with deep neural networks from the sparse and unstable point cloud for identification. We demonstrate that our deep learning based method can identify 12 people with an accuracy of 89%. Besides, we also propose an approach for intruder detection with 73% accuracy. Through this work we show the strong feature extraction ability of deep neural networks, as it is very hard to extract manually crafted features from radar point clouds with conventional methods.

In Chapter 4 and Chapter 5 we present using deep learning techniques for micro and macro tracking, respectively, which shows significant performance improvement

compared to conventional signal processing methods. In Chapter 4 we proposed a robot based system for heart rate monitoring at home, applying convolutional neural network on extracted phase variation data for heart rate estimation. Besides improving the accuracy, by formulating the loss with Bayesian model, we show that we are able to estimate the uncertainty of the neural network prediction. The uncertainty estimate is very important for health care applications, like heart rate monitoring. In Chapter 5 we proposed using deep neural network on range-angle heatmaps for indoor drone localization. Our solution is easy to setup, requiring only one anchor and no drone modification. Our learning based method also has much better performance than conventional signal processing methods in terms of accuracy and robustness.

In Chapter 6, we propose an end-to-end deep learning method for mmWave FMCW radar gesture/activity recognition tasks, which is one of the few explorations in this direction. By replacing the conventional Discrete Fourier Transform (DFT) pre-processing with stacked complex layers, we are able to directly take complex raw radar data cube as input, and produce consistent better performance compared to previous hybrid baselines. We show that with deep neural network replacing the conventional DFT pre-processing, the more informative features can be better extracted from the raw data, which benefits the downstream neural network classifier. We believe that this is a small step towards a universal feature extractor for mmWave FMCW radar data.

We are moving towards an era of pervasive IoT and smart environment computing, and sensing with mmWave FMCW radar is an important and promising direction. Through the research in this thesis, we show that the combination of deep learning and radar signal processing could significantly improve the performance on tracking and recognizing targets with mmWave radar, and we believe that it serves as an important contribution in this field.

References

[1] Mark Weiser. "The computer for the 21st century". In: *ACM SIGMOBILE mobile computing and communications review* 3.3 (1999), pp. 3–11.

[2] Jeannette Chin, Vic Callaghan, and Somaya Ben Allouch. "The Internet-of-Things: Reflections on the past, present and future from a user-centered and smart environment perspective". In: *Journal of Ambient Intelligence and Smart Environments* 11.1 (2019), pp. 45–69.

[3] Dieter Uckelmann, Mark Harrison, and Florian Michahelles. *Architecting the internet of things*. Springer Science & Business Media, 2011.

[4] Roberto Minerva, Abyi Biru, and Domenico Rotondi. *Towards a definition of the Internet of Things (IoT)*. [EB/OL]. `https://iot.ieee.org/definition.html`. 2015.

[5] Matthew B Hoy. "Alexa, Siri, Cortana, and more: an introduction to voice assistants". In: *Medical reference services quarterly* 37.1 (2018), pp. 81–88.

[6] Jinyang Li, Zhenyu Li, Gareth Tyson, and Gaogang Xie. "Your privilege gives your privacy away: An analysis of a home security camera service". In: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE. 2020, pp. 387–396.

[7] Jinyang Li, Zhenyu Li, Gareth Tyson, and Gaogang Xie. "Characterising Usage Patterns and Privacy Risks of a Home Security Camera Service". In: *IEEE Transactions on Mobile Computing* (2020).

[8] Josephine Lau, Benjamin Zimmerman, and Florian Schaub. "Alexa, are you listening? Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers". In: *Proceedings of the ACM on Human-Computer Interaction* 2.CSCW (2018), pp. 1–31.

[9] Nathan Malkin, Joe Deatrick, Allen Tong, Primal Wijesekera, Serge Egelman, and David Wagner. "Privacy attitudes of smart speaker users". In: *Proceedings on Privacy Enhancing Technologies* 2019.4 (2019).

[10] Jimmy Liranzo and Thaier Hayajneh. "Security and privacy issues affecting cloud-based IP camera". In: *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. IEEE. 2017, pp. 458–465.

[11] Deepti Sehrawat and Nasib Singh Gill. "Smart sensors: Analysis of different types of IoT sensors". In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE. 2019, pp. 523–528.

[12] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A Stankovic, Niki Trigoni, and Andrew Markham. "See through smoke: robust indoor mapping with low-cost mmWave radar." In: *MobiSys*. 2020, pp. 14–27.

[13] Mostafa Alizadeh, George Shaker, João Carlos Martins De Almeida, Plinio Pelegrini Morita, and Safeddin Safavi-Naeini. "Remote monitoring of human vital signs using mm-wave FMCW radar". In: *IEEE Access* 7 (2019), pp. 54958–54968.

[14] Yong Wang, Wen Wang, Mu Zhou, Aihu Ren, and Zengshan Tian. "Remote monitoring of human vital signs based on 77-GHz mm-wave FMCW radar". In: *Sensors* 20.10 (2020), p. 2999.

[15] Toan Khanh Vodai, Kellen Oleksak, Tsotne Kvelashvili, Farnaz Foroughian, Chandler Bauder, Paul Theilmann, Aly Fathy, and Ozlem Kilic. "Enhancement of Remote Vital Sign Monitoring Detection Accuracy Using Multiple-Input Multiple-Output 77 GHz FMCW Radar". In: *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology* (2021).

[16] Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. "mmVib: micrometer-level vibration measurement with mmwave radar". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 2020, pp. 1–13.

[17] Huining Li, Chenhan Xu, Aditya Singh Rathore, Zhengxiong Li, Hanbin Zhang, Chen Song, Kun Wang, Lu Su, Feng Lin, Kui Ren, et al. "VocalPrint: exploring a resilient and secure voice authentication via mmWave biometric interrogation". In: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 2020, pp. 312–325.

[18] Chenhan Xu, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Huining Li, Chen Song, Kun Wang, and Wenyao Xu. "Waveear: Exploring a mmwave-based noise-resistant speech sensing for voice-user interface". In: *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. 2019, pp. 14–26.

[19] Yudi Dong and Yu-Dong Yao. "Secure MMWave-Radar-Based Speaker Verification for IoT Smart Home". In: *IEEE Internet of Things Journal* 8.5 (2020), pp. 3500–3511.

[20] Sameera Palipana, Dariush Salami, Luis A Leiva, and Stephan Sigg. "Pantomime: Mid-Air Gesture Recognition with Sparse Millimeter-Wave Radar Point Clouds". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.1 (2021), pp. 1–27.

[21] Kun Qian, Zhaoyuan He, and Xinyu Zhang. "3D point cloud generation with millimeter-wave radar". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.4 (2020), pp. 1–23.

[22] HP Groll and J Detlefsen. "History of automotive anticollision radars and final experimental results of a MM-Wave car radar developed by the Technical University of Munich". In: *IEEE Aerospace and Electronic Systems Magazine* 12.8 (1997), pp. 15–19.

[23] Hatem Hajri and Mohamed-Cherif Rahal. "Real time lidar and radar high-level fusion for obstacle detection and tracking with evaluation on a ground truth". In: *arXiv preprint arXiv:1807.11264* (2018).

[24] Seong Kyung Kwon, Eugin Hyun, Jin-Hee Lee, Jonghun Lee, and Sang Hyuk Son. "A low-complexity scheme for partially occluded pedestrian detection using LiDAR-radar sensor fusion". In: *2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE. 2016, pp. 104–104.

[25] Renyuan Zhang and Siyang Cao. "Extending reliability of mmwave radar tracking and detection via fusion with camera". In: *IEEE Access* 7 (2019), pp. 137065–137079.

[26] Xian Shuai, Yulin Shen, Yi Tang, Shuyao Shi, Luping Ji, and Guoliang Xing. "milliEye: A Lightweight mmWave Radar and Camera Fusion System for Robust Object Detection". In: *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. 2021, pp. 145–157.

[27] Martin Pirkl and Franz-Jose Tospann. "HiVision millimeter-wave radar for enhanced vision systems in civil and military transport aircraft". In: *Enhanced and Synthetic Vision 1997*. Vol. 3088. International Society for Optics and Photonics. 1997, pp. 8–18.

[28] Texas Instruments. *Vision: Possible*. 2020. URL: `https://news.ti.com/blog/2020/09/29/vision-possible`.

[29] Texas Instruments. *Journey of innovation*. 2020. URL: `https://news.ti.com/blog/2020/10/06/journey-innovation`.

[30] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. "Soli: Ubiquitous gesture sensing with millimeter wave radar". In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), p. 142.

[31] Joe Hindy. *Soli on the Google Pixel 4: Here's why it's a good thing.* [EB/OL]. `https://www.androidauthority.com/google-pixel-4-soli-motion-sense-1042448/`.

[32] Christian Wolff. *Frequency-Modulated Continuous-Wave Radar (FMCW Radar)*. [EB/OL]. `https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html`.

[33] Andrzej Wojtkiewicz, Jacek Misiurewicz, M Nalecz, Konrad Jedrzejewski, and Krzysztof Kulpa. "Two-dimensional signal processing in FMCW radars". In: *Proc. XX KKTOiUE* (1997), pp. 475–480.

[34] Cesar Iovescu and Sandeep Rao. "The fundamentals of millimeter wave sensors". In: *Texas Instruments* (2017), pp. 1–8.

[35] Sandeep Rao. *MIMO Radar*. [EB/OL]. `https://www.ti.com/lit/an/swra554a/swra554a.pdf`.

[36] Ralph Schmidt. "Multiple emitter location and signal parameter estimation". In: *IEEE transactions on antennas and propagation* 34.3 (1986), pp. 276–280.

[37] William F Gabriel. "Superresolution techniques in the range domain". In: *IEEE International Conference on Radar*. IEEE. 1990, pp. 263–267.

[38] Richard Roy, A Paulraj, and Thomas Kailath. "Estimation of signal parameters via rotational invariance techniques-ESPRIT". In: *MILCOM 1986-IEEE Military Communications Conference: Communications-Computers: Teamed for the 90's.* Vol. 3. IEEE. 1986, pp. 41–6.

[39] Maxime Schutz, Cyril Decroze, Michèle Lalande, and Bertrand Lenoir. "Neural Networks to Increase Range Resolution of FMCW Radar". In: *IEEE Sensors Letters* 4.8 (2020), pp. 1–4.

[40] Maximilian Gall, Markus Gardill, Thomas Horn, and Jonas Fuchs. "Spectrum-based Single-Snapshot Super-Resolution Direction-of-Arrival Estimation using Deep Learning". In: *2020 German Microwave Conference (GeMiC).* IEEE. 2020, pp. 184–187.

[41] Sandeep Rao. *Introduction to mmwave Sensing: FMCW Radars.* [EB/OL]. `https://training.ti.com/sites/default/files/docs/mmwaveSensing-FMCW-offlineviewing_4.pdf`.

[42] Jih-Tsun Yu, Li Yen, and Po-Hsuan Tseng. "mmWave Radar-based Hand Gesture Recognition using Range-Angle Image". In: *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring).* IEEE. 2020, pp. 1–5.

[43] Souvik Hazra and Avik Santra. "Short-range radar-based gesture recognition system using 3D CNN with triplet loss". In: *IEEE Access* 7 (2019), pp. 125623–125633.

[44] Karan Ahuja, Yue Jiang, Mayank Goel, and Chris Harrison. "Vid2Doppler: Synthesizing Doppler Radar Data from Videos for Training Privacy-Preserving Activity Recognition". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 2021, pp. 1–10.

[45] Thomas Stadelmayer and Avik Santra. "Data-driven Radar Processing Using a Parametric Convolutional Neural Network for Human Activity Classification". In: (2021).

[46] Waseem Rawat and Zenghui Wang. "Deep convolutional neural networks for image classification: A comprehensive review". In: *Neural computation* 29.9 (2017), pp. 2352–2449.

[47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.

[48] Zixing Zhang, Jürgen Geiger, Jouni Pohjalainen, Amr El-Desoky Mousa, Wenyu Jin, and Björn Schuller. "Deep learning for environmentally robust speech recognition: An overview of recent developments". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 9.5 (2018), pp. 1–28.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems.* 2017, pp. 5998–6008.

[50] Jamil Fayyad, Mohammad A Jaradat, Dominique Gruyer, and Homayoun Najjaran. "Deep learning sensor fusion for autonomous vehicle perception and localization: A review". In: *Sensors* 20.15 (2020), p. 4220.

[51] Alex Kendall, Matthew Grimes, and Roberto Cipolla. "Posenet: A convolutional network for real-time 6-dof camera relocalization". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2938–2946.

[52] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[53] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[54] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[55] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[57] Yingying Wang, Hu Cheng, Chaoqun Wanga, and Max Q-H Meng. "Pose Invariant Inertial Odometry for Pedestrian Localization". In: *IEEE Transactions on Instrumentation and Measurement* (2021).

[58] Zihang Lai and Weidi Xie. "Self-supervised learning for video correspondence flow". In: *arXiv preprint arXiv:1905.00875* (2019).

[59] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[60] Daniel A Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. "Complex-valued neural networks for fully-temporal micro-Doppler classification". In: *2019 20th International Radar Symposium (IRS)*. IEEE. 2019, pp. 1–10.

[61] Zhimian Zhang, Haipeng Wang, Feng Xu, and Ya-Qiu Jin. "Complex-valued convolutional neural network and its application in polarimetric SAR image classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.12 (2017), pp. 7177–7188.

[62] Ronny Hänsch and Olaf Hellwich. "Complex-valued convolutional neural networks for object detection in PolSAR data". In: *8th European Conference on Synthetic Aperture Radar*. VDE. 2010, pp. 1–4.

[63] Xiaofeng Tan, Ming Li, Peng Zhang, Yan Wu, and Wanying Song. "Complex-valued 3-D convolutional neural network for PolSAR image classification". In: *IEEE Geoscience and Remote Sensing Letters* 17.6 (2019), pp. 1022–1026.

[64] Wen Xie, Gaini Ma, Feng Zhao, Hanqiang Liu, and Lu Zhang. "PolSAR image classification via a novel semi-supervised recurrent complex-valued convolution neural network". In: *Neurocomputing* 388 (2020), pp. 255–268.

[65] Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. "Deep complex networks". In: *arXiv preprint arXiv:1705.09792* (2017).

[66] Martin Arjovsky, Amar Shah, and Yoshua Bengio. "Unitary evolution recurrent neural networks". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 1120–1128.

[67] Nitzan Guberman. "On complex valued convolutional neural networks". In: *arXiv preprint arXiv:1602.09046* (2016).

[68] Raktim Gautam Goswami, Sivaganesh Andhavarapu, and K Murty. "Phase Aware Speech Enhancement using Realisation of Complex-valued LSTM". In: *arXiv preprint arXiv:2010.14122* (2020).

[69] Yanxin Hu, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie. "DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement". In: *arXiv preprint arXiv:2008.00264* (2020).

[70] Muqiao Yang, Martin Q Ma, Dongyu Li, Yao-Hung Hubert Tsai, and Ruslan Salakhutdinov. "Complex transformer: A framework for modeling complex-valued sequence". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 4232–4236.

[71] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231.

[72] Harold W Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[73] Rudolph Emil Kalman. "A new approach to linear filtering and prediction problems". In: (1960).

[74] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. "The probabilistic data association filter". In: *IEEE Control Systems Magazine* 29.6 (2009), pp. 82–100.

[75] Samuel S Blackman. "Multiple hypothesis tracking for multiple target tracking". In: *IEEE Aerospace and Electronic Systems Magazine* 19.1 (2004), pp. 5–18.

[76] AE Aubert, L Welkenhuysen, J Montald, L De Wolf, H Geivers, J Minten, Hugo Kesteloot, and H Geest. "Laser method for recording displacement of the heart and chest wall". In: *Journal of biomedical engineering* 6.2 (1984), pp. 134–140.

[77] Anne De Groote, Muriel Wantier, Guy Chéron, Marc Estenne, and Manuel Paiva. "Chest wall motion during tidal breathing". In: *Journal of Applied Physiology* 83.5 (1997), pp. 1531–1537.

[78] Mingmin Zhao, Fadel Adib, and Dina Katabi. "Emotion recognition using wireless signals". In: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 2016, pp. 95–108.

[79] Unsoo Ha, Salah Assana, and Fadel Adib. "Contactless seismocardiography via deep learning radars". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 2020, pp. 1–14.

[80] Jian Gong, Xinyu Zhang, Kaixin Lin, Ju Ren, Yaoxue Zhang, and Wenxun Qiu. "RF Vital Sign Sensing Under Free Body Movement". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2021).

[81] Manjunath Thindlu Rudrappa, Reinhold Herschel, and Peter Knott. "Vital parameters detection of non stationary human subject using mimo radar". In: *2020 21st International Radar Symposium (IRS)*. IEEE. 2020, pp. 125–130.

[82] Peijun Zhao, Chris Xiaoxuan Lu, Bing Wang, Changhao Chen, Linhai Xie, Mengyu Wang, Niki Trigoni, and Andrew Markham. "Heart Rate Sensing with a Robot Mounted mmWave Radar". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 2812–2818.

[83] Francesco Tonolini and Fadel Adib. "Networking across boundaries: enabling wireless communication through the water-air interface". In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 2018, pp. 117–131.

[84] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. "Capturing the human figure through a wall". In: *ACM Transactions on Graphics (TOG)* 34.6 (2015), pp. 1–13.

[85] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. "Through-wall human pose estimation using radio signals". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7356–7365.

[86] Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. "mm-Pose: Real-time human skeletal posture estimation using mmWave radars and CNNs". In: *IEEE Sensors Journal* 20.17 (2020), pp. 10032–10044.

[87] Hongfei Xue, Yan Ju, Chenglin Miao, Yijiang Wang, Shiyang Wang, Aidong Zhang, and Lu Su. "mmMesh: towards 3D real-time dynamic human mesh construction using millimeter-wave". In: *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 2021, pp. 269–282.

[88] Jacopo Pegoraro, Francesca Meneghello, and Michele Rossi. "Multiperson continuous tracking and identification from mm-wave micro-Doppler signatures". In: *IEEE Transactions on Geoscience and Remote Sensing* 59.4 (2020), pp. 2994–3009.

[89] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. "mid: Tracking and identifying people with millimeter wave radar". In: *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. 2019, pp. 33–40.

[90] Zhen Meng, Song Fu, Jie Yan, Hongyuan Liang, Anfu Zhou, Shilin Zhu, Huadong Ma, Jianhua Liu, and Ning Yang. "Gait recognition for co-existing multiple people using millimeter wave sensing". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 01. 2020, pp. 849–856.

[91] Jacopo Pegoraro and Michele Rossi. "Real-time People Tracking and Identification from Sparse mm-Wave Radar Point-clouds". In: *IEEE Access* (2021).

[92] Yong Wang, Shasha Wang, Mu Zhou, Qing Jiang, and Zengshan Tian. "TS-I3D based hand gesture recognition method with radar sensor". In: *IEEE Access* 7 (2019), pp. 22902–22913.

[93] Piyali Goswami, Sandeep Rao, Sachin Bharadwaj, and Amanda Nguyen. "Real-time multi-gesture recognition using 77 GHz FMCW MIMO single chip radar". In: *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE. 2019, pp. 1–4.

[94] Yangfan Sun, Renlong Hang, Zhu Li, Mouqing Jin, and Kelvin Xu. "Privacy-preserving fall detection with deep learning on mmWave radar signal". In: *2019 IEEE Visual Communications and Image Processing (VCIP)*. IEEE. 2019, pp. 1–4.

[95] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum". In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM. 2016, pp. 851–860.

[96] Ziheng Li, Zhenyuan Lei, An Yan, Erin Solovey, and Kaveh Pahlavan. "ThuMouse: A micro-gesture cursor input through mmWave radar-based interaction". In: *2020 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE. 2020, pp. 1–9.

[97] Haipeng Liu, Yuheng Wang, Anfu Zhou, Hanyue He, Wei Wang, Kunpeng Wang, Peilin Pan, Yixuan Lu, Liang Liu, and Huadong Ma. "Real-time Arm Gesture Recognition in Smart Home Scenarios via Millimeter Wave Sensing". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4.4 (2020), pp. 1–28.

[98] Jie Wang, Liang Zhang, Changcheng Wang, Xiaorui Ma, Qinghua Gao, and Bin Lin. "Device-free human gesture recognition with generative adversarial networks". In: *IEEE Internet of Things Journal* 7.8 (2020), pp. 7678–7688.

[99] Haipeng Liu, Anfu Zhou, Zihe Dong, Yuyang Sun, Jiahe Zhang, Liang Liu, Huadong Ma, Jianhua Liu, and Ning Yang. "M-Gesture: Person-Independent Real-Time In-Air Gesture Recognition Using Commodity Millimeter Wave Radar". In: *IEEE Internet of Things Journal* (2021).

[100] Abdelwahed Khamis, Branislav Kusy, Chun Tung Chou, Mary-Louise McLaws, and Wen Hu. "RFWash: a weakly supervised tracking of hand hygiene technique". In: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 2020, pp. 572–584.

[101] Panneer Selvam Santhalingam, Yuanqi Du, Riley Wilkerson, Ding Zhang, Parth Pathak, Huzefa Rangwala, Raja Kushalnagar, et al. "Expressive ASL Recognition using Millimeter-wave Wireless Signals". In: *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2020, pp. 1–9.

[102] Josiah Wayland Smith, Orges Furxhi, and Murat Torlak. "An FCNN-Based Super-Resolution mmWave Radar Framework for Contactless Musical Instrument Interface". In: *IEEE Transactions on Multimedia* (2021).

[103] Zhengxiong Li, Baicheng Chen, Zhuolin Yang, Huining Li, Chenhan Xu, Xingyu Chen, Kun Wang, and Wenyao Xu. "Ferrotag: A paper-based mmwave-scannable tagging infrastructure". In: *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 2019, pp. 324–337.

[104] Shiwei Fang and Shahriar Nirjon. "SuperRF: Enhanced 3D RF Representation Using Stationary Low-Cost mmWave Radar". In: *Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks*. 2020, pp. 120–131.

[105] Baicheng Chen, Huining Li, Zhengxiong Li, Xingyu Chen, Chenhan Xu, and Wenyao Xu. "ThermoWave: a new paradigm of wireless passive temperature monitoring via mmWave sensing". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 2020, pp. 1–14.

[106] Zhengxiong Li, Fenglong Ma, Aditya Singh Rathore, Zhuolin Yang, Baicheng Chen, Lu Su, and Wenyao Xu. "Wavespy: Remote and through-wall screen attack via mmwave sensing". In: *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2020, pp. 217–232.

[107] Akarsh Prabhakara, Vaibhav Singh, Swarun Kumar, and Anthony Rowe. "Osprey: a mmWave approach to tire wear sensing". In: *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*. 2020, pp. 28–41.

[108] Junfeng Guan, Sohrab Madani, Suraj Jog, Saurabh Gupta, and Haitham Hassanieh. "Through Fog High-Resolution Imaging Using Millimeter Wave Radar". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11464–11473.

[109] Xin Yang, Jian Liu, Yingying Chen, Xiaonan Guo, and Yucheng Xie. "MU-ID: Multi-user identification through gaits using millimeter wave radios". In: *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE. 2020, pp. 2589–2598.

[110] Chris Xiaoxuan Lu, Muhamad Risqi U Saputra, Peijun Zhao, Yasin Almalioglu, Pedro PB de Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, and Andrew Markham. "milliEgo: Single-chip mmWave Radar Aided Egomotion Estimation via Deep Sensor Fusion". In: *ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 2020.

[111] Kshitiz Bansal, Keshav Rungta, Siyuan Zhu, and Dinesh Bharadia. "Pointillism: accurate 3D bounding box estimation with multi-radars". In: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 2020, pp. 340–353.

[112] Josiah W Smith, Shiva Thiagarajan, Richard Willis, Yiorgos Makris, and Murat Torlak. "Improved static hand gesture classification on deep convolutional neural networks using novel sterile training technique". In: *IEEE Access* 9 (2021), pp. 10893–10902.

[113] Yasin Almalioglu, Mehmet Turan, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. "Milli-RIO: Ego-motion estimation with low-cost millimetre-wave radar". In: *IEEE Sensors Journal* 21.3 (2020), pp. 3314–3323.

[114] Junchen Guo, Meng Jin, Yuan He, Weiguo Wang, and Yunhao Liu. "Dancing Waltz with Ghosts: Measuring Sub-mm-Level 2D Rotor Orbit with a Single mmWave Radar". In: *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021)*. 2021, pp. 77–92.

[115] Muhammed Zahid Ozturk, Chenshu Wu, Beibei Wang, and KJ Ray Liu. "GaitCube: Deep Data Cube Learning for Human Recognition with Millimeter-Wave Radio". In: *IEEE Internet of Things Journal* (2021).

[116] Guoying Zhao, Guoyi Liu, Hua Li, and Matti Pietikainen. "3D gait recognition using multiple cameras". In: *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*. IEEE. 2006, pp. 529–534.

[117] Yeong Sang Park, Young-Sik Shin, Joowan Kim, and Ayoung Kim. "3D ego-Motion Estimation Using low-Cost mmWave Radars via Radar Velocity Factor for Pose-Graph SLAM". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 7691–7698.

[118] Yuheng Wang, Haipeng Liu, Kening Cui, Anfu Zhou, Wensheng Li, and Huadong Ma. "m-Activity: Accurate and Real-Time Human Activity Recognition Via Millimeter Wave Radar". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 8298–8302.

[119] Aakriti Adhikari, Austin Hetherington, and Sanjib Sur. "mmFlow: Facilitating At-Home Spirometry with 5G Smart Devices". In: *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2021, pp. 1–9.

[120] Yi Zhang, Zheng Yang, Guidong Zhang, Chenshu Wu, and Li Zhang. "XGest: Enabling Cross-Label Gesture Recognition with RF Signals". In: *ACM Transactions on Sensor Networks (TOSN)* 17.4 (2021), pp. 1–23.

[121] Hem Regmi, Moh Sabbir Saadat, Sanjib Sur, and Srihari Nelakuditi. "SquiggleMilli: Approximating SAR Imaging on Mobile Millimeter-Wave Devices". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.3 (2021), pp. 1–26.

[122] Chenhan Xu, Huining Li, Zhengxiong Li, Hanbin Zhang, Aditya Singh Rathore, Xingyu Chen, Kun Wang, Ming-chun Huang, and Wenyao Xu. "CardiacWave: A mmWave-based Scheme of Non-Contact and High-Definition Heart Activity Computing". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.3 (2021), pp. 1–26.

[123] Yumeng Liang, Anfu Zhou, Huanhuan Zhang, Xinzhe Wen, and Huadong Ma. "FG-LiquID: A Contact-less Fine-grained Liquid Identifier by Pushing the Limits of Millimeter-wave Sensing". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5.3 (2021), pp. 1–27.

[124] Xinbo Zheng, Zhaocheng Yang, Kaixuan He, and Haifan Liu. "Hand gesture recognition based on range Doppler-angle trajectory and LSTM network using an MIMO radar". In: *Eleventh International Conference on Signal Processing Systems*. Vol. 11384. International Society for Optics and Photonics. 2019, 113840P.

[125] B Dekker, S Jacobs, AS Kossen, MC Kruithof, AG Huizing, and M Geurts. "Gesture recognition with a low power FMCW radar and a deep convolutional neural network". In: *2017 European Radar Conference (EURAD)*. IEEE. 2017, pp. 163–166.

[126] Qisong Wu, Dixian Zhao, et al. "Dynamic hand gesture recognition using FMCW radar sensor for driving assistance". In: *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE. 2018, pp. 1–6.

[127] Renyuan Zhang and Siyang Cao. "Real-time human motion behavior detection via CNN using mmWave radar". In: *IEEE Sensors Letters* 3.2 (2018), pp. 1–4.

[128] Desheng Liu and Hongfu Meng. "Application of FMCW Radar for the Recongnition of Hand Gesture Using Time Series Convolutional Neural Networks". In: *2020 International Conference on Microwave and Millimeter Wave Technology (ICMMT)*. IEEE. 2020, pp. 1–3.

[129] Rongjie Huang, Zecheng Li, Song Wang, Ronghao Wang, Junjie Li, and Zhiwei Xu. "A RD-T Network for Hand Gesture Recognition Based on Millimeter-Wave Sensor". In: *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*. IEEE. 2020, pp. 308–312.

[130] Yong Wang, Zedong Zhao, Mu Zhou, and Jinjun Wu. "Two Dimensional Parameters Based Hand Gesture Recognition Algorithm for FMCW Radar Systems". In: *International Conference on Wireless and Satellite Systems*. Springer. 2019, pp. 226–234.

[131] Penghui Chen, Yujing Bai, and Jun Wang. "Gesture Recognition Using LFMCW Radar and Convolutional Neural Network". In: *DEStech Transactions on Computer Science and Engineering* cscme (2019).

[132] Souvik Hazra and Avik Santra. "Radar Gesture Recognition System in Presence of Interference using Self-Attention Neural Network". In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE. 2019, pp. 1409–1414.

[133] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.

[134] Ejaz Ahmed, Michael Jones, and Tim K Marks. "An improved deep learning architecture for person re-identification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3908–3916.

[135] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. "Deepreid: Deep filter pairing neural network for person re-identification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 152–159.

[136] Igor Barros Barbosa, Marco Cristani, Alessio Del Bue, Loris Bazzani, and Vittorio Murino. "Re-identification with rgb-d sensors". In: *European Conference on Computer Vision*. Springer. 2012, pp. 433–442.

[137] Jin Wang, Mary She, Saeid Nahavandi, and Abbas Kouzani. "A review of vision-based gait recognition methods for human identification". In: *2010 international conference on digital image computing: techniques and applications*. IEEE. 2010, pp. 320–327.

[138] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. "A simple yet effective baseline for 3d human pose estimation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2640–2649.

[139] Alexander Toshev and Christian Szegedy. "Deeppose: Human pose estimation via deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1653–1660.

[140] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video". In: *International Journal of Computer Vision* 126.2 (2018), pp. 430–439.

[141] Jun Wan, Qiuqi Ruan, Wei Li, and Shuang Deng. "One-shot learning gesture recognition from RGB-D data using bag of features". In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 2549–2582.

[142] Zhipeng Liu, Xiujuan Chai, Zhuang Liu, and Xilin Chen. "Continuous gesture recognition with hand-oriented spatiotemporal feature". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 3056–3064.

[143] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. "Visual simultaneous localization and mapping: a survey". In: *Artificial intelligence review* 43.1 (2015), pp. 55–81.

[144] Thomas Schops, Torsten Sattler, and Marc Pollefeys. "Bad slam: Bundle adjusted direct rgb-d slam". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 134–144.

[145] Ronald Clark, Sen Wang, Andrew Markham, Niki Trigoni, and Hongkai Wen. "Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6856–6864.

[146] Bing Wang, Changhao Chen, Chris Xiaoxuan Lu, Peijun Zhao, Niki Trigoni, and Andrew Markham. "Atloc: Attention guided camera localization". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 06. 2020, pp. 10393–10401.

[147] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. "X-vectors: Robust dnn embeddings for speaker recognition". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 5329–5333.

[148] Yingke Zhu, Tom Ko, David Snyder, Brian Kan Wing Mak, and Daniel Povey. "Self-attentive Speaker Embeddings for Text-independent Speaker Verification". In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. 2018, p. 3573.

[149] John J Leonard and Hugh F Durrant-Whyte. "Simultaneous map building and localization for an autonomous mobile robot." In: *IROS*. Vol. 3. 1991, pp. 1442–1447.

[150] Yang Qifan, Tang Hao, Zhao Xuebing, Li Yin, and Zhang Sanfeng. "Dolphin: Ultrasonic-based gesture recognition on smartphone platform". In: *2014 IEEE 17th International Conference on Computational Science and Engineering*. IEEE. 2014, pp. 1461–1468.

[151] Hiroki Watanabe, Tsutomu Terada, and Masahiko Tsukamoto. "Gesture Recognition Method Utilizing Ultrasonic Active Acoustic Sensing". In: *Journal of Information Processing* 25 (2017), pp. 331–340.

[152] Wei Xu, ZhiWen Yu, Zhu Wang, Bin Guo, and Qi Han. "Acousticid: gait-based human identification using acoustic signal". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.3 (2019), pp. 1–25.

[153] Kaustubh Kalgaonkar and Bhiksha Raj. "Acoustic doppler sonar for gait recogination". In: *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*. IEEE. 2007, pp. 27–32.

[154] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. "Contactless sleep apnea detection on smartphones". In: *Proceedings of the 13th annual international conference on mobile systems, applications, and services*. 2015, pp. 45–57.

[155] Kun Qian, Chenshu Wu, Fu Xiao, Yue Zheng, Yi Zhang, Zheng Yang, and Yunhao Liu. "Acousticcardiogram: Monitoring heartbeats using acoustic signals on smart devices". In: *Proceedings of IEEE INFOCOM*. 2018.

[156] Jie Yang, Simon Sidhom, Gayathri Chandrasekaran, Tam Vu, Hongbo Liu, Nicolae Cecan, Yingying Chen, Marco Gruteser, and Richard P Martin. "Sensing driver phone use with acoustic ranging through car speakers". In: *IEEE Transactions on Mobile Computing* 11.9 (2012), pp. 1426–1440.

[157] Wenjun Jiang, Hongfei Xue, Chenglin Miao, Shiyang Wang, Sen Lin, Chong Tian, Srinivasan Murali, Haochen Hu, Zhi Sun, and Lu Su. "Towards 3D human pose construction using WiFi". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. 2020, pp. 1–14.

[158] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. "Wigest: A ubiquitous wifi-based gesture recognition system". In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE. 2015, pp. 1472–1480.

[159] Fadel Adib and Dina Katabi. "See through walls with WiFi!" In: *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*. 2013, pp. 75–86.

[160] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, et al. "Towards environment independent device free human activity recognition". In: *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. 2018, pp. 289–304.

[161] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures". In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. 2014, pp. 617–628.

[162] Bo Wei, Wen Hu, Mingrui Yang, and Chun Tung Chou. "Radio-based device-free activity recognition with radio frequency interference". In: *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. 2015, pp. 154–165.

[163] Yunze Zeng, Parth H Pathak, and Prasant Mohapatra. "WiWho: wifi-based person identification in smart spaces". In: *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*. 2016.

[164] Wei Wang, Alex X Liu, and Muhammad Shahzad. "Gait recognition using wifi signals". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2016, pp. 363–373.

[165] Jian Liu, Yan Wang, Yingying Chen, Jie Yang, Xu Chen, and Jerry Cheng. "Tracking vital signs during sleep leveraging off-the-shelf wifi". In: *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM. 2015, pp. 267–276.

[166] Xuefeng Liu, Jiannong Cao, Shaojie Tang, Jiaqi Wen, and Peng Guo. "Contactless respiration monitoring via off-the-shelf WiFi devices". In: *IEEE Transactions on Mobile Computing* 15.10 (2015), pp. 2466–2479.

[167] Yongsen Ma, Gang Zhou, and Shuangquan Wang. "WiFi sensing with channel state information: A survey". In: *ACM Computing Surveys (CSUR)* 52.3 (2019), pp. 1–36.

[168] Phuc Nguyen, Shane Transue, Min-Hyung Choi, Ann C Halbower, and Tam Vu. "Wikispiro: Non-contact respiration volume monitoring during sleep". In: *Proceedings of the Eighth Wireless of the Students, by the Students, and for the Students Workshop*. 2016, pp. 27–29.

[169] Youngwook Kim and Brian Toomajian. "Hand gesture recognition using micro-Doppler signatures with convolutional neural network". In: *IEEE Access* 4 (2016), pp. 7125–7130.

[170] Myoungseok Yu, Narae Kim, Yunho Jung, and Seongjoo Lee. "A frame detection method for real-time hand gesture recognition systems using CW-radar". In: *Sensors* 20.8 (2020), p. 2321.

[171] Youngwook Kim and Taesup Moon. "Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks". In: *IEEE geoscience and remote sensing letters* 13.1 (2015), pp. 8–12.

[172] Hoang Thanh Le, Son Lam Phung, and Abdesselam Bouzerdoum. "Human gait recognition with micro-doppler radar and deep autoencoder". In: *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE. 2018, pp. 3347–3352.

[173] Chris Xiaoxuan Lu, Hongkai Wen, Sen Wang, Andrew Markham, and Niki Trigoni. "SCAN: learning speaker identity From noisy sensor data". In: *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM. 2017, pp. 67–78.

[174] Robert Beringer, Andrew Sixsmith, Michael Campo, Julie Brown, and Rose McCloskey. "The "acceptance" of ambient assisted living: Developing an alternate methodology to this limited research lens". In: *International Conference on Smart Homes and Health Telematics*. Springer. 2011, pp. 161–167.

[175] Han Zou, Yuxun Zhou, Jianfei Yang, Weixi Gu, Lihua Xie, and Costas J Spanos. "WiFi-Based Human Identification via Convex Tensor Shapelet Learning." In: *AAAI*. 2018.

[176] Csaba Benedek, Balázs Nagy, Bence Gálai, and Zsolt Jankó. "Lidar-based gait analysis in people tracking and 4D visualization". In: *Signal Processing Conference (EUSIPCO), 2015 23rd European*. 2015.

[177] David D Ferris and Nicholas C Currie. "Microwave and millimeter-wave systems for wall penetration". In: *Targets and Backgrounds: Characterization and Representation IV*. Vol. 3375. International Society for Optics and Photonics. 1998, pp. 269–280.

[178] Pascal Kowalczuk. "Consumer acceptance of smart speakers: a mixed methods approach". In: *Journal of Research in Interactive Marketing* 12.4 (2018), pp. 418–431.

[179] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. "A discriminative feature learning approach for deep face recognition". In: *European conference on computer vision*. Springer. 2016, pp. 499–515.

[180] Chris Xiaoxuan Lu, Bowen Du, Peijun Zhao, Hongkai Wen, Yiran Shen, Andrew Markham, and Niki Trigoni. "Deepauth: in-situ authentication for smartwatches via deeply learned behavioural biometrics". In: *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. ACM. 2018, pp. 204–207.

[181] U.S. Census Bureau. *Historical Households Tables*. [EB/OL]. `https://www.census.gov/data/tables/time-series/demo/families/households.html`.

[182] Roy L Streit and Tod E Luginbuhl. *Probabilistic multi-hypothesis tracking*. Tech. rep. NAVAL UNDERWATER SYSTEMS CENTER NEWPORT RI, 1995.

[183] Wayne R Blanding, Peter K Willett, and Yaakov Bar-Shalom. "Multiple target tracking using maximum likelihood probabilistic data association". In: *2007 IEEE Aerospace Conference*. IEEE. 2007, pp. 1–12.

[184] Lee Middleton, Alex A Buss, Alex Bazin, and Mark S Nixon. "A floor sensor system for gait recognition". In: *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*. 2005.

[185] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. "Silhouette analysis-based gait recognition for human identification". In: *IEEE transactions on pattern analysis and machine intelligence* 25.12 (2003), pp. 1505–1518.

[186] Faezeh Tafazzoli and Reza Safabakhsh. "Model-based human gait recognition using leg and arm movements". In: *Engineering Applications of Artificial Intelligence* 23.8 (2010), pp. 1237–1246. URL: `http://www.sciencedirect.com/science/article/pii/S0952197610001417`.

[187] Johannes Preis, Moritz Kessel, Martin Werner, and Claudia Linnhoff-Popien. "Gait recognition with kinect". In: *1st international workshop on kinect in pervasive computing*. New Castle, UK. 2012, P1–P4.

[188] Moeness G Amin, Yimin D Zhang, Fauzia Ahmad, and KC Dominic Ho. "Radar signal processing for elderly fall detection: The future for in-home monitoring". In: *IEEE Signal Processing Magazine* 33.2 (2016), pp. 71–80.

[189] Xiaoxuan Lu, Hongkai Wen, Han Zou, Hao Jiang, Lihua Xie, and Niki Trigoni. "Robust occupancy inference with commodity WiFi". In: *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2016, pp. 1–8.

[190] Fadel Adib, Hongzi Mao, Zachary Kabelac, Dina Katabi, and Robert C Miller. "Smart homes that monitor breathing and heart rate". In: *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM. 2015, pp. 837–846.

[191] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. "RF-based 3D skeletons". In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM. 2018, pp. 267–281.

[192] Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani Srivastava. "Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar". In: *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*. 2019, pp. 51–56.

[193] M Ishijima. "Cardiopulmonary monitoring by textile electrodes without subject-awareness of being monitored". In: *Medical and Biological Engineering and Computing* 35.6 (1997), pp. 685–690.

[194] Douglas A Perednia and Ace Allen. "Telemedicine technology and clinical applications". In: *Jama* 273.6 (1995), pp. 483–488.

[195] Zhenhua Jia, Amelie Bonde, Sugang Li, Chenren Xu, Jingxian Wang, Yanyong Zhang, Richard E Howard, and Pei Zhang. "Monitoring a Person's Heart Rate and Respiratory Rate on a Shared Bed Using Geophones". In: *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. 2017.

[196] Sergey Tulyakov, Xavier Alameda-Pineda, Elisa Ricci, Lijun Yin, Jeffrey F Cohn, and Nicu Sebe. "Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2396–2404.

[197] Hao Wang, Daqing Zhang, Junyi Ma, Yasha Wang, Yuxiang Wang, Dan Wu, Tao Gu, and Bing Xie. "Human respiration detection with commodity wifi devices: do user location and body orientation matter?" In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2016, pp. 25–36.

[198] Shigeki Sugimoto, Hayato Tateda, Hidekazu Takahashi, and Masatoshi Okutomi. "Obstacle detection using millimeter-wave radar and its visualization on image sequence". In: *International Conference on Pattern Recognition*. 2004.

[199] MYM Wong, E Pickwell-MacPherson, and YT Zhang. "Contactless and continuous monitoring of heart rate based on photoplethysmography on a mattress". In: *Physiological measurement* 31.7 (2010), p. 1065.

[200] Bhavin Chamadiya, Kunal Mankodiya, Manfred Wagner, and Ulrich G Hofmann. "Textile-based, contactless ECG monitoring for non-ICU clinical settings". In: *Journal of Ambient Intelligence and Humanized Computing* 4.6 (2013), pp. 791–800.

[201] Benjamin Eilebrecht, Tobias Wartzek, Jeroen Lem, Rainer Vogt, and Steffen Leonhardt. "Capacitive electrocardiogram measurement system in the driver seat". In: *ATZ worldwide eMagazine* 113.3 (2011), pp. 50–55.

[202] Mohd Norzali Haji Mohd, Masayuki Kashima, Kiminori Sato, and Mutsumi Watanabe. "A non-invasive facial visual-infrared stereo vision based measurement as an alternative for physiological measurement". In: *Asian Conference on Computer Vision*. Springer. 2014, pp. 684–697.

[203] Kual-Zheng Lee, Pang-Chan Hung, and Luo-Wei Tsai. "Contact-free heart rate measurement using a camera". In: *2012 Ninth Conference on Computer and Robot Vision*. IEEE. 2012, pp. 147–152.

[204] Kaveh Bakhtiyari, Nils Beckmann, and Jürgen Ziegler. "Contactless heart rate variability measurement by IR and 3D depth sensors with respiratory sinus arrhythmia". In: *Procedia Computer Science* 109 (2017), pp. 498–505.

[205] Xiaochuan He, Rafik Goubran, and Frank Knoefel. "IR night vision video-based estimation of heart and respiration rates". In: *2017 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2017, pp. 1–5.

[206] Avinash Parnandi and Ricardo Gutierrez-Osuna. "Contactless measurement of heart rate variability from pupillary fluctuations". In: *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*. IEEE. 2013, pp. 191–196.

[207] Michael H Li, Azadeh Yadollahi, and Babak Taati. "Noncontact vision-based cardiopulmonary monitoring in different sleeping positions". In: *IEEE journal of biomedical and health informatics* 21.5 (2016), pp. 1367–1375.

[208] Ming-Zher Poh, Daniel J McDuff, and Rosalind W Picard. "Advancements in noncontact, multiparameter physiological measurements using a webcam". In: *IEEE transactions on biomedical engineering* 58.1 (2010), pp. 7–11.

[209] Lingyun Ren, Yun Seo Koo, Haofei Wang, Yazhou Wang, Quanhua Liu, and Aly E Fathy. "Noncontact multiple heartbeats detection and subject localization using UWB impulse doppler radar". In: *IEEE Microwave and Wireless Components Letters* 25.10 (2015), pp. 690–692.

[210] Jussi Salmi, Olli Luukkonen, and Visa Koivunen. "Continuous wave radar based vital sign estimation: Modeling and experiments". In: *2012 IEEE Radar Conference*. IEEE. 2012, pp. 0564–0569.

[211] Jure Kranjec, S Beguš, G Geršak, and J Drnovšek. "Non-contact heart rate and heart rate variability measurements: A review". In: *Biomedical signal processing and control* 13 (2014), pp. 102–112.

[212] Xuyu Wang, Chao Yang, and Shiwen Mao. "PhaseBeat: Exploiting CSI phase data for vital sign monitoring with commodity WiFi devices". In: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2017, pp. 1230–1239.

[213] Texas Instruments. *MMWave Training Series*. `https://training.ti.com/mmwave-training-series`.

[214] Laura Anitori, Ardjan de Jong, and Frans Nennie. "FMCW radar for life-sign detection". In: *2009 IEEE Radar Conference*. IEEE. 2009, pp. 1–6.

[215] Tom Kwan and Kenneth Martin. "Adaptive detection and enhancement of multiple sinusoids using a cascade IIR filter". In: *IEEE Transactions on Circuits and Systems* 36.7 (1989), pp. 937–947.

[216] Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, and Niki Trigoni. "Deep Learning based Pedestrian Inertial Navigation: Methods, Dataset and On-Device Inference". In: *IEEE Internet of Things Journal* (2020).

[217] Alex Kendall and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems*. 2017, pp. 5574–5584.

[218] Changhao Chen, Xiaoxuan Lu, Johan Wahlstrom, Andrew Markham, and Niki Trigoni. "Deep Neural Network Based Inertial Odometry Using Low-cost Inertial Measurement Units". In: *IEEE Transactions on Mobile Computing* (2019).

[219] Alex Kendall, Yarin Gal, and Roberto Cipolla. "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7482–7491.

[220] Texas Instruments. *TI IWR6843ISK*. `http://www.ti.com/tool/IWR6843ISK`.

[221] Texas Instruments. *TI MMWAVEICBOOST*. `https://www.ti.com/tool/MMWAVEICBOOST`.

[222] *American Time Use Survey News Release*. `https://www.bls.gov/news.release/atus.htm`.

[223] DAVID J Ewing, JM Neilson, and PAUL Travis. "New method for assessing cardiac parasympathetic activity using 24 hour electrocardiograms." In: *Heart* 52.4 (1984), pp. 396–402.

[224] Masatoshi Sekine and Kurato Maeno. "Non-contact heart rate detection using periodic variation in Doppler frequency". In: *2011 IEEE Sensors Applications Symposium*. IEEE. 2011, pp. 318–322.

[225] Francesco Sartor, Gabriele Papini, Lieke Gertruda Elisabeth Cox, and John Cleland. "Methodological shortcomings of wrist-worn heart rate monitors validations". In: *Journal of medical Internet research* 20.7 (2018), e10108.

[226] Steven A Israel, John M Irvine, Brenda K Wiederhold, and Mark D Wiederhold. *The heartbeat: the living biometric*. Wiley-IEEE Press, New York, NY, USA, 2009.

[227] Chetana Hegde, H Rahul Prabhu, DS Sagar, P Deepa Shenoy, KR Venugopal, and Lalit M Patnaik. "Heartbeat biometrics for human authentication". In: *Signal, Image and Video Processing* 5.4 (2011), p. 485.

[228] Shaima Al Habsi, Maha Shehada, Marwah Abdoon, Ahmed Mashood, and Hassan Noura. "Integration of a Vicon camera system for indoor flight of a Parrot AR Drone". In: *2015 10th International Symposium on Mechatronics and its Applications (ISMA)*. IEEE. 2015, pp. 1–6.

[229] Kexin Guo, Zhirong Qiu, Cunxiao Miao, Abdul Hanif Zaini, Chun-Lin Chen, Wei Meng, and Lihua Xie. "Ultra-wideband-based localization for quadcopter navigation". In: *Unmanned Systems* 4.01 (2016), pp. 23–34.

[230] Nicola Macoir, Jan Bauwens, Bart Jooris, Ben Van Herbruggen, Jen Rossey, Jeroen Hoebeke, and Eli De Poorter. "Uwb localization with battery-powered wireless backbone for drone-based inventory management". In: *Sensors* 19.3 (2019), p. 467.

[231] Ali Yassin, Youssef Nasser, Ahmed Y Al-Dubai, and Mariette Awad. "MOSAIC: Simultaneous localization and environment mapping using mmWave without a-priori knowledge". In: *IEEE Access* 6 (2018), pp. 68932–68947.

[232] Mohammed Aladsani, Ahmed Alkhateeb, and Georgios C Trichopoulos. "Leveraging mmWave imaging and communications for simultaneous localization and mapping". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 4539–4543.

[233] John Donald. "Indoor Flight of AR Drone 2.0 UAV in Vicon Motion Capture System". In: *The UNSW Canberra at ADFA Journal of Undergraduate Engineering Research* 10.1 (2018).

[234] Milton CP Santos, Lucas V Santana, Alexandre S Brandão, Mário Sarcinelli-Filho, and Ricardo Carelli. "Indoor low-cost localization system for controlling aerial robots". In: *Control Engineering Practice* 61 (2017), pp. 93–111.

[235] Janis Tiemann, Florian Schweikowski, and Christian Wietfeld. "Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments". In: *2015 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2015* (2015).

[236] A Masiero, F Fissore, R Antonello, A Cenedese, and A Vettore. "A COMPARISON OF UWB AND MOTION CAPTURE UAV INDOOR POSITIONING." In: *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* (2019).

[237] Anton Ledergerber, Michael Hamer, and Raffaello D'Andrea. "A robot self-localization system using one-way ultra-wideband communication". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 3131–3137.

[238] Anton Ledergerber, Michael Hamer, and Raffaello D'Andrea. "Angle of arrival estimation based on channel impulse response measurements". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 6686–6692.

[239] Alessandro Benini, Adriano Mancini, and Sauro Longhi. "An imu/uwb/vision-based extended kalman filter for mini-uav localization in indoor environment using 802.15. 4a wireless sensor network". In: *Journal of Intelligent & Robotic Systems* 70.1-4 (2013), pp. 461–476.

[240] David Hoeller, Anton Ledergerber, Michael Hamer, and Raffaello D'Andrea. "Augmenting ultra-wideband localization with computer vision for accurate flight". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 12734–12740.

[241] Weide You, Fanbiao Li, Liqing Liao, and Meili Huang. "Data Fusion of UWB and IMU Based on Unscented Kalman Filter for Indoor Localization of Quadrotor UAV". In: *IEEE Access* 8 (2020), pp. 64971–64981.

[242] S. Zahran, M. M. Mostafa, A. Masiero, A. M. Moussa, A. Vettore, and N. El-Sheimy. "Micro-radar and UWB aided UAV navigation in GNSS denied environment". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 42.1 (2018), pp. 469–476.

[243] Janis Tiemann and Christian Wietfeld. "Scalable and precise multi-UAV indoor navigation using TDOA-based UWB localization". In: *2017 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2017* 2017-Janua.September (2017), pp. 1–7.

[244] Wang Shule, Carmen Martínez Almansa, Jorge Peña Queralta, Zhuo Zou, and Tomi Westerlund. "UWB-Based Localization for Multi-UAV Systems and Collaborative Heterogeneous Multi-Robot Systems". In: *Procedia Computer Science* 175.2019 (2020), pp. 357–364. arXiv: 2004.08174. URL: https://doi.org/10.1016/j.procs.2020.07.051.

[245] Yang Qi, Yisheng Zhong, and Zongying Shi. "Cooperative 3-D relative localization for UAV swarm by fusing UWB with IMU and GPS". In: *Journal of Physics: Conference Series* 1642 (2020), p. 012028.

[246] C. Aker and S. Kalkan. "Using deep networks for drone detection". In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2017, pp. 1–6.

[247] Kristijan Mirceta, Ciril Bohak, Byeong Hak Kim, Min Young Kim, and Matija Marolt. "Drone segmentation and tracking in grounded sensor scanned LiDAR datasets". In: *Proceedings of the Zbornik sedemindvajsete mednarodne Elektrotehniške in racunalniške konference ERK*. 2018, pp. 384–387.

[248] Byeong Hak Kim, Danish Khan, Ciril Bohak, Wonju Choi, Hyun Jeong Lee, and Min Young Kim. "V-RBNN based small drone detection in augmented datasets for 3D LADAR system". In: *Sensors* 18.11 (2018), p. 3825.

[249] Xianyu Chang, Chaoqun Yang, Junfeng Wu, Xiufang Shi, and Zhiguo Shi. "A surveillance system for drone localization and tracking using acoustic arrays". In: *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE. 2018, pp. 573–577.

[250] M. M. Azari, H. Sallouha, A. Chiumento, S. Rajendran, E. Vinogradov, and S. Pollin. "Key Technologies and System Trade-offs for Detection and Localization of Amateur Drones". In: *IEEE Communications Magazine* 56.1 (2018), pp. 51–57.

[251] İsmail Güvenç, Ozgur Ozdemir, Yavuz Yapici, Hani Mehrpouyan, and David Matolak. "Detection, localization, and tracking of unauthorized UAS and jammers". In: *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*. IEEE. 2017, pp. 1–10.

[252] M Caris, W Johannes, S Sieger, V Port, and S Stanko. "Detection of small UAS with W-band radar". In: *2017 18th International Radar Symposium (IRS)*. IEEE. 2017, pp. 1–6.

[253] Guohua Wei, Yuxiang Zhou, and Siliang Wu. "Detection and localization of high speed moving targets using a short-range UWB impulse radar". In: *2008 IEEE Radar Conference*. IEEE. 2008, pp. 1–4.

[254] Jedrzej Drozdowicz, Maciej Wielgo, Piotr Samczynski, Krzysztof Kulpa, Jaroslaw Krzonkalla, Maj Mordzonek, Marcin Bryl, and Zbigniew Jakielaszek. "35 GHz FMCW drone detection system". In: *2016 17th International Radar Symposium (IRS)*. IEEE. 2016, pp. 1–4.

[255] Vasilii Semkin, Jaakko Haarla, Thomas Pairon, Christopher Slezak, Sundeep Rangan, Ville Viikari, and Claude Oestges. "Analyzing radar cross section signatures of diverse drone models at mmwave frequencies". In: *IEEE Access* 8 (2020), pp. 48958–48969.

[256] Texas Instruments. *Industrial mmWave radar sensors*. [EB/OL]. URL: `https://www.ti.com/sensors/mmwave-radar/industrial/overview.html`.

[257] Infineon. *Radar Sensors*. [EB/OL]. URL: `https://www.infineon.com/cms/en/product/sensor/radar-sensors/`.

[258] Anuradha Singh, Saeed Ur Rehman, Sira Yongchareon, and Peter Han Joo Chong. "Multi-resident non-contact vital sign monitoring using radar: A review". In: *IEEE Sensors Journal* (2020).

[259] Feng Jin, Arindam Sengupta, and Siyang Cao. "mmFall: Fall Detection Using 4-D mmWave Radar and a Hybrid Variational RNN AutoEncoder". In: *IEEE Transactions on Automation Science and Engineering* (2020).

[260] Bo Wang, Liang Guo, Hao Zhang, and Yong-Xin Guo. "A Millimetre-Wave Radar-Based Fall Detection Method Using Line Kernel Convolutional Neural Network". In: *IEEE Sensors Journal* 20.22 (2020), pp. 13364–13370.

[261] Shahzad Ahmed, Karam Dad Kallu, Sarfaraz Ahmed, and Sung Ho Cho. "Hand gestures recognition using radar sensors for human-computer-interaction: A review". In: *Remote Sensing* 13.3 (2021), p. 527.

[262] Xinyu Li, Yuan He, and Xiaojun Jing. "A survey of deep learning-based human activity recognition in radar". In: *Remote Sensing* 11.9 (2019), p. 1068.

[263] Wenbin Ye, Haiquan Chen, and Bing Li. "Using an end-to-end convolutional network on radar signal for human activity classification". In: *IEEE Sensors Journal* 19.24 (2019), pp. 12244–12252.

[264] Running Zhao, Xiaolin Ma, Xinhua Liu, and Jian Liu. "An End-to-End Network for Continuous Human Motion Recognition via Radar Radios". In: *IEEE Sensors Journal* (2020).

[265] Wen Jiang, Yihui Ren, Ying Liu, Ziao Wang, and Xinghua Wang. "Recognition of Dynamic Hand Gesture Based on Mm-Wave Fmcw Radar Micro-Doppler Signatures". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2021, pp. 4905–4909.

[266] Moritz Scherer, Michele Magno, Jonas Erb, Philipp Mayer, Manuel Eggimann, and Luca Benini. "TinyRadarNN: Combining Spatial and Temporal Convolutional Neural Networks for Embedded Gesture Recognition with Short Range Radars". In: *IEEE Internet of Things Journal* (2021), pp. 1–1.

[267] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* 1.2 (2017), p. 4.

[268] S Franceschini, M Ambrosanio, S Vitale, F Baselice, A Gifuni, G Grassini, and V Pascazio. "Hand Gesture Recognition via Radar Sensors and Convolutional Neural Networks". In: *2020 IEEE Radar Conference (RadarConf20)*. IEEE. 2020, pp. 1–5.

[269] Takuya Sakamoto, Xiaomeng Gao, Ehsan Yavari, Ashikur Rahman, Olga Boric-Lubecke, and Victor M Lubecke. "Hand gesture recognition using a radar echo I–Q plot and a convolutional neural network". In: *IEEE sensors letters* 2.3 (2018), pp. 1–4.

[270] Wenbin Ye and Haiquan Chen. "Human activity classification based on micro-Doppler signatures by multiscale and multitask Fourier convolutional neural network". In: *IEEE Sensors Journal* 20.10 (2020), pp. 5473–5479.

[271] Akira Hirose. *Complex-valued neural networks: theories and applications*. Vol. 5. World Scientific, 2003.

[272] George Kechriotis and Elias S Manolakos. "Training fully recurrent neural networks with complex weights". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 41.3 (1994), pp. 235–238.

[273] George M Georgiou and Cris Koutsougeras. "Complex domain backpropagation". In: *IEEE transactions on Circuits and systems II: analog and digital signal processing* 39.5 (1992), pp. 330–334.

[274] Jingkun Gao, Bin Deng, Yuliang Qin, Hongqiang Wang, and Xiang Li. "Enhanced radar imaging using a complex-valued convolutional neural network". In: *IEEE Geoscience and Remote Sensing Letters* 16.1 (2018), pp. 35–39.

[275] Hyun-Woong Cho, Sungdo Choi, Young-Rae Cho, and Jongseok Kim. "Complex-Valued Channel Attention and Application in Ego-Velocity Estimation With Automotive Radar". In: *IEEE Access* 9 (2021), pp. 17717–17727.

[276] Kilian Rambach and Bin Yang. "MIMO radar: time division multiplexing vs. Code division multiplexing". In: (2017).

[277] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[278] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1026–1034.

[279] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. "Randla-net: Efficient semantic segmentation of large-scale point clouds". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11108–11117.

[280] Javier Gozalvez. "Samsung Electronics Sets 5G Speed Record at 7.5 Gb/s [Mobile Radio]". In: *IEEE Vehicular Technology Magazine* 10.1 (2015), pp. 12–16.