

Use of deep neural networks for classification of micro-Doppler signature from radar data

Master's thesis in MPCOM, MPSYS

Daoyuan Yang, 940405-5791

Liting Zhou, 940622-2829

Department of Electrical engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

MASTER'S THESIS 2018

Use of deep neural networks for classification of micro-Doppler signature from radar data

Daoyuan Yang, Liting Zhou



Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Use of deep neural networks for classification
of micro-Doppler signature from radar data
Daoyuan Yang, Liting Zhou

© Daoyuan Yang, Liting Zhou 2018.

Supervisor: Kasra Haghighi, UniqueSec AB

Examiner: Thomas Rylander, Chalmers University of Technology.

Master's Thesis 2018

Department of Electrical engineering

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 723750147

Gothenburg, Sweden 2018

Abstract

This thesis explores the usage of deep neural networks for the classification of micro-Doppler signatures collected by means of radar. First, we conduct the simulation of a micro-Doppler model based on a Frequency-Modulated Continuous-Wave (FMCW) radar in combination with a point target, a bike wheel and a walking human, where the model is also validated with measurements. Second, we explore the use of multiple deep-learning algorithms for micro-Doppler signature classification. A training set of 12 classes of human activities is measured and it is used to train Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Although the overall test-accuracy is around 90%, the neural networks tend to mislabel some classes that have similar probabilistic distribution.

Keywords: FMCW radar, micro-Doppler, CNN, RNN, classification, deep learning

Acknowledgements

We would like to thank Kasra Haghighi from Uniquesec AB, the thesis supervisor, for providing us with the opportunity to work on the project and for his continuous support during the past half year. We also appreciate the examiner, professor Thomas Rylander at Chalmers University of Technology, for helping us with thesis writing and for the permission to use computing cluster.



Contents

List of Figures	xv
List of Tables	xix
1 Introduction	1
1.1 Background	1
1.2 Problem formulation	2
1.3 Contribution	2
1.4 Outline	3
2 Theory	5
2.1 Modeling micro-Doppler signature for FMCW radar	5
2.1.1 Stationary target	7
2.1.2 Moving target with approximately constant velocity	8
2.1.3 Moving target with rapidly changing velocity	10
2.2 Micro-Doppler model of complex non-rigid bodies	11
2.3 Classification of micro-Doppler signature by machine learning	12
2.3.1 Traditional machine learning (ML) algorithms	13
2.3.2 Feature extraction algorithms	14
2.3.2.1 Principal component analysis (PCA)	14
2.3.2.2 Linear discriminant analysis (LDA)	15
2.3.3 Classifiers	16
2.3.3.1 K-nearest neighbors (k-NN)	16
2.3.3.2 Support vector machine (SVM)	17
2.3.4 Deep learning algorithms	18

2.3.4.1	Logistic regression	18
2.3.4.2	Feedforward neural networks	19
2.3.4.3	Convolutional neural networks	20
2.3.4.4	Recurrent neural networks	22
2.3.4.5	Optimization and regularization of neural networks	22
3	Hardware Description	25
3.1	FMCW radar evaluation kit introduction	25
3.1.1	Hardware connection	26
3.1.2	Antenna	26
3.1.3	Transceiver	27
3.1.4	Power and controller board	28
3.2	Hardware limitations	28
4	Modelling and Simulation	33
4.1	Introduction	33
4.2	Simulation and measurement settings	33
4.3	Point target	35
4.3.1	Simulation	35
4.4	Bicycle wheel	36
4.4.1	Simulation	37
4.4.2	Experimental measurement	39
4.5	Walking human	41
4.5.1	Simulation	41
4.5.2	Experimental measurement	45
5	Classification With Neural Networks	51
5.1	Data measurement campaign	51
5.2	Data preprocessing	53
5.3	Neural networks structure and classification results	54
5.3.1	CNN based models	55
5.3.1.1	Neural network structure	55

5.3.1.2	Performance	55
5.3.1.3	Error analysis	57
5.3.1.4	The influence of DC component	59
5.3.2	RNN based neural networks	60
5.3.2.1	Performance	60
5.3.2.2	Error analysis	61
5.4	Predict only the activity (4-class problem)	61
5.5	Tuning parameters of neural networks	62
6	Discussion	67
6.1	Limitations	67
6.1.1	Overfitting	67
6.1.2	Not enough data	68
6.1.3	Changes of the environment	68
6.2	Future work	68
6.2.1	Changing radar setup	69
6.2.2	Using practical activities	69
6.2.3	Exploring other algorithms	70
6.2.4	Using simulation data to train	70
7	Conclusion	71
	Bibliography	73
A	Appendix 1 Spectrogram examples	I
B	Appendix 2 Neural Network model summary	V
B.1	Software used in this thesis	V
B.2	Model summary of Double input CNN	V
B.3	Model summary of RNN based neural networks	VII

List of Figures

2.1	A plot of two successive sawtooth radar chirps' frequency.	6
2.2	The block diagram of an FMCW radar.	7
2.3	A long chirp can be divided into multiple successive short chirps to meet the constant velocity assumption.	11
2.4	A feedforward neural network with one hidden layer.	20
2.5	An example of a convolutional layer and a pooling layer.	21
2.6	The computational graph of a simple RNN with one hidden layer and a feedback connection of the hidden layer to itself (biases not shown).	22
3.1	The radar kit includes three parts: a transceiver board (RS2400K), a power and controller board (CO1000A), and a horn antenna with a gain of 20dB.	27
3.2	Noise power spectrogram density (PSD) measurement of the radar kit.	31
3.3	The actual signal transmission model of the radar kit. Two sawtooth chirps (approximated by staircase waveform) are plotted. Dashed line in each chirp represents the omitted steps, and one second gap is in between of two chirps due to serial communication.	32
4.1	Point target moves straight towards and away from the radar.	36
4.2	Point target oscillates tangentially to the radar.	37
4.3	The positions of the radar and the bike wheel in two situations. Due to the axis settings, the size of the wheel looks different in both figures.	38
4.4	Spectrograms of the wheel when the number of spokes is (a) one, (b) two, or (c) eight.	39

4.5	Spectrogram of the wheel when the radar is placed at the z-axis. The wheel has one spoke.	40
4.6	Bike wheel measurement environment and spectrograms.	41
4.7	The joints of the human model.	43
4.8	The dimension of human parts.	44
4.9	Human ellipsoid model.	44
4.10	Trajectories of human joints during a gait cycle.	44
4.11	The spectrogram generated by four body parts in high resolution FMCW radar.	47
4.12	The spectrogram generated by four body parts in low resolution FMCW radar.	47
4.13	The spectrogram of a walking human with high resolution radar settings <i>S1</i>	48
4.14	The spectrogram of a human walking with low resolution radar settings <i>S2</i>	48
4.15	Micro-Doppler measurement for human walking towards the radar with swinging arms.	49
4.16	Micro-Doppler measurement for human walking towards the radar without swinging arms.	49
5.1	Radar setup model for data measurement campaign.	52
5.2	Radar kit assembly with a 3D printed box.	53
5.3	Data measurement campaign environment.	53
5.4	The structure of the <i>small CNN</i> unit.	56
5.5	The structure of the <i>Double input CNN</i> which is built by two <i>small CNN</i> units.	56
5.6	The plot of training history of <i>Double input CNN</i>	56
5.7	Confusion matrix of <i>Double input CNN</i> (12-class problem).	57
5.8	The example of a mislabeled sample.	59

5.9	RNN based neural networks' structure. <i>Input</i> : spectrogram of window length 64 or 128; <i>conv1d</i> : 24 one dimensional convolutions with filter size 3; <i>RNN unit</i> : either GRU or LSTM unit with output size 24; <i>avg</i> : the forward output and backward output of bidirectional RNN units are averaged; <i>Dense</i> : two fully connected layers. Dashed lines represent recursive connection.	60
5.10	Confusion matrix of bidirectional GRU model.	62
5.11	Confusion matrix of bidirectional LSTM model.	63
5.12	Confusion matrix of <i>Double input CNN</i> (4-class problem).	64
5.13	Training history of an overfitting model based on <i>Double input CNN</i> when the data is divided into four different classes.	66
A.1	Spectrogram samples of the data set, calculated by STFT with window length of 64 and an overlap of 16. Each row represents one posture. The order of the postures is: walking with swinging arms (walk_wa), walking without swinging arms (walk_woa), boxing while standing still (boxing) and standing still (standing).	II
A.2	The spectrograms of the same samples as in figure A.1, but calculated by STFT window length of 128 and an overlap of 32. Each row represents one posture. The order of the postures is: walking with swinging arms (walk_wa), walking without swinging arms (walk_woa), boxing while standing still (boxing) and standing still (standing).	III

List of Tables

3.1	RS3400K/00 24 GHz FMCW Transceiver Evaluation Kit Specification.	26
3.2	Command categories for radar controller.	28
4.1	Radar parameters for simulation and measurement.	34
5.1	Class description and the number of samples.	52
5.2	Performance comparison between <i>small CNN</i> unit and <i>Double input CNN</i> (12 class problem).	55
5.3	Some common mistakes made by <i>Double input CNN</i>	58
5.4	RNN based neural networks performance.	61
5.5	Performance comparison between <i>small CNN</i> unit and <i>Double input CNN</i> (4-class problem).	65
5.6	Examples of the status of a machine learning model.	65
B.1	Software used in this thesis	V

1

Introduction

1.1 Background

The original utility of radars is range and speed measurement. Another important application based on radars is target classification. Nowadays, radar-based target classification has gained a lot of interest due to demands in many areas such as security, autonomous driving and health care [1]. One important feature that enables the classification of the target is the micro-Doppler signature generated by the target [2]. The micro-Doppler phenomenon results from the individual motions of the target's different parts, such as the arms and legs of a walking human. Thus, classification of targets may be accomplished if the targets have different micro-Doppler signatures.

Traditional methods for micro-Doppler signature classification requires significant domain knowledge and calculation to devise discriminative features. Recently, deep learning algorithms such as deep Convolutional Neural Networks (CNN) and deep Recurrent Neural Networks (RNN) have gained large research interests, since they can solve some complex problems without an explicit model. Therefore, the classification may be done without sophisticated processing and feature extraction of the signal itself, where classical machine learning algorithms would require pre-processing that can successfully separate the classes by, e.g., hyper-planes.

1.2 Problem formulation

The goal of the project is to achieve activity classification of human activities measured by a Frequency-Modulated Continuous-Wave (FMCW) radar. Four research questions will be addressed in the thesis:

1. What are the working principles of FMCW radars?
2. What is the performance of a commercially available FMCW radar at low price?
3. How should the micro-Doppler signatures be modeled for such FMCW radars?
4. What preprocessing of the raw radar signal is needed to form a data set that can be used to train machine learning models such as neural networks?
5. For vulnerable road users what classification performance can be achieved by different neural networks?

To answer these questions, the project has two main parts: (i) the first part is the modelling, simulation and experimental evaluation of the micro-Doppler signatures measured by an FMCW radar; and (ii) the second part is to classify different human activities by CNN and RNN.

1.3 Contribution

The main contribution of this project is to explore deep learning algorithms' capability to classify the micro-Doppler signatures produced by different human activities. Despite some limitations in range resolution and noise floor of the off-the-shelf radar hardware, we demonstrate that neural network models are able to classify micro-Doppler signatures although the model size is kept small. These techniques can be implemented in low-cost embedded systems and, thus, contribute to many applications in areas such as surveillance systems.

1.4 Outline

The rest of the thesis is structured as follows:

- Chapter 2 presents a theoretical model of FMCW radars for some different scenarios and related work in micro-Doppler signature classification research. A brief introduction to machine learning is also given in this chapter.
- Chapter 3 describes the detailed specification and usage of the FMCW radar developer kit, which is used for the rest of the project. This is followed by a discussion on some limitations of the radar hardware.
- Chapter 4 demonstrates the simulation of the FMCW radar with three kinds of moving targets and a comparison with the measurement.
- Chapter 5 describes the data measurement campaign, the pre-processing and the classification results of some neural networks (both CNN based and RNN based). For each neural network, the network structure is shown followed by classification result and error analysis.
- Chapter 6 discusses the limitations of the methods and some future work.
- Chapter 7 concludes the thesis.

2

Theory

2.1 Modeling micro-Doppler signature for FMCW radar

Micro-Doppler signature can be observed by many kinds of radars including pulse radars, Continuous-Wave (CW) radars and FMCW radars. For example, in [2], Chen shows that the micro-Doppler signatures of a helicopter and a human can be measured by a pulse radar by means of both simulation and measurement. It is also possible to observe micro-Doppler signatures with a CW radar where the baseband signal is a sinusoidal wave [3]. However, pulse radars and CW radars have their drawbacks. Pulse radars require a very accurate clock and high peak power. CW radars are not able to estimate the range of the target.

To avoid these drawbacks, this thesis focuses on the FMCW radar. The FMCW radar is a special type of CW radar since it modulates the frequency of the baseband signal with some waveform such as sawtooth, triangle and sinusoidal waveform. A commonly used waveform is the sawtooth that implies that the radar signal's frequency sweeps linearly in time from f_c (carrier frequency) to $f_c + B$, where B is the bandwidth. The sweep period is t_p and sweep slope is $k = B/t_p$. The expression of the transmitted signal with amplitude A_t is given by:

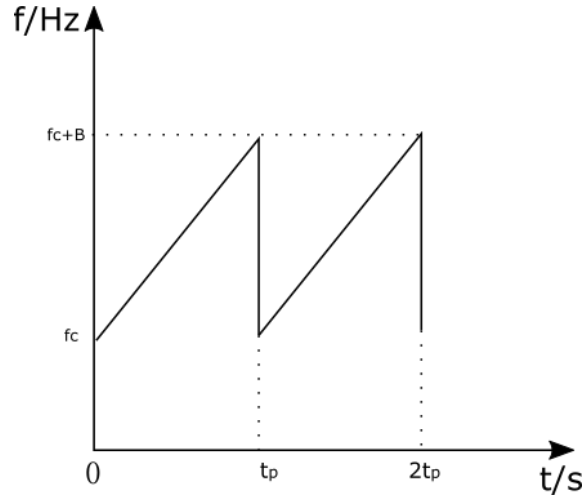


Figure 2.1: A plot of two successive sawtooth radar chirps' frequency.

$$s_{tx}(t) = A_t \cos \left[2\pi \left(f_c t + \frac{k}{2} t^2 \right) \right], \quad (0 \leq t \leq t_p) \quad (2.1)$$

Figure 2.1 shows the frequency of two transmitted sawtooth waveforms where such a linear variation of the frequency also is referred to as a chirp.

The block diagram shown in figure 2.2 illustrates how the transmitted signal is generated. The baseband signal is generated by a sawtooth signal generator and a Voltage-Controlled Oscillator (VCO). An ideal VCO generates a sinusoidal signal whose frequency is proportional to the input voltage. Then, the baseband signal is mixed with the carrier signal and divided into two paths by the power splitter. One path is fed to the antenna and the other one is fed to the receiver. The power splitter requires a high isolation between the two output ports to avoid that the transmitted signal directly enters the receiver.

In the following discussion, an FMCW model in combination with a point target is discussed for three cases. Here, we assume the radar is monostatic (transmitter and receiver are collocated) and it has an isotropic antenna. The radar is stationary and it is located at \mathbf{R}_r in a global coordinate system. The target is located at $\mathbf{R}_t(t)$. It is moving with the velocity $\mathbf{V}_t(t)$.

For this situation, we consider the three cases:

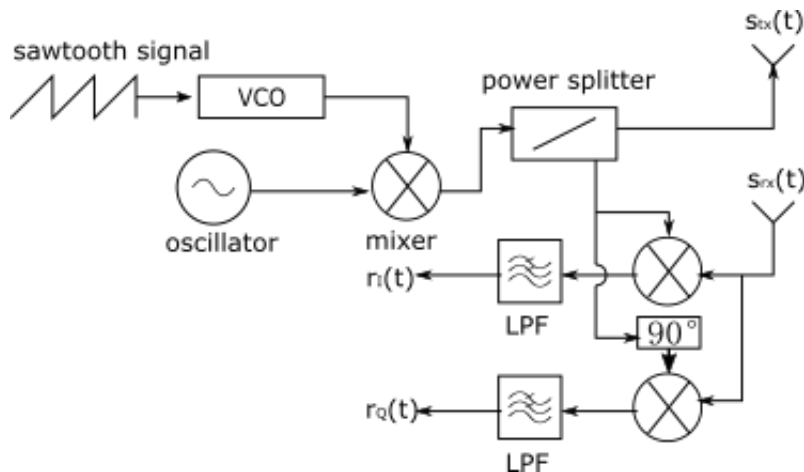


Figure 2.2: The block diagram of an FMCW radar.

- The target is stationary.
- The target moves at approximately constant velocity.
- The target moves at rapidly varying velocity.

2.1.1 Stationary target

First, we consider the case when the target is stationary and it is located at a distance $d = |\mathbf{R}_t(0) - \mathbf{R}_r|$ from the radar. The target is the only object illuminated by the radar.

Because of the propagation time of the electromagnetic wave, the received signal is delayed by $\tau = 2d/c$ with respect to the transmitted signal, where c is the wave propagation speed and A_r is the received amplitude determined by the radar equation [4]. The received signal is expressed as:

$$s_{rx}(t) = A_r \cdot s_{tx} \left(t - \frac{2d}{c} \right) \quad (2.2)$$

Figure 2.2 shows how the received signal is demodulated. The received signal $s_{rx}(t)$ is mixed with transmitted signal and the 90-degree shifted version of the transmitted signal separately. The mixed signals pass low-pass filters and we get an

in-phase ($r_I(t)$) signal and a quadrature-phase ($r_Q(t)$) signal in baseband. A complex baseband signal $r(t)$ is constructed based on the combination of $r_I(t)$ and $r_Q(t)$ as follows:

$$\begin{aligned} r(t) &= r_I(t) + jr_Q(t) = Ae^{j\Phi(t)} \\ \Phi(t) &= 2\pi kt\tau - \pi k\tau^2 + 2\pi f_c\tau \end{aligned} \tag{2.3}$$

where A is the amplitude of the baseband signal and it depends on miscellaneous gains of the whole process.

The baseband signal has the frequency:

$$f_b = \frac{\partial\Phi}{2\pi\partial t} = k\tau \tag{2.4}$$

where f_b is called the range beat frequency. The distance between the radar and the object can be calculated by substituting $\tau = 2d/c$ into equation (2.4).

2.1.2 Moving target with approximately constant velocity

Second, we assume that the target is moving. Its position, velocity and acceleration are denoted as $\mathbf{R}_t(t)$, $\mathbf{V}_t(t)$ and $\mathbf{A}_t(t)$ respectively. The velocity can be assumed to be approximately constant when the chirp duration is short enough. *Short* duration means that the target's velocity does not change significantly during the chirp, i.e. $\int_t^{t+t_p} \mathbf{A}_t(\tau)d\tau \approx \mathbf{0}$. While the target is moving at a constant velocity, the received signal has another frequency component induced by radial speed where this frequency component is referred to as the Doppler frequency. Denote the radial speed of the target as:

$$v_r(t) := \frac{\mathbf{V}_t(t) \cdot (\mathbf{R}_t(t) - \mathbf{R}_r)}{|\mathbf{R}_t(t) - \mathbf{R}_r|} \tag{2.5}$$

Because of the constant velocity assumption, the radial speed is also assumed to be constant during one chirp, i.e. $v_r(t) \approx v_r(0)$, for $0 \leq t \leq t_p$. In order to measure the range and the speed from the received baseband signal, multiple successive chirps

are transmitted. We analyze one chirp here, and it can be extended to the following chirps easily.

Denote $d(t) := |\mathbf{R}_r - \mathbf{R}_t(t)|$ as the distance between the target and the radar. Then, the initial distance is $d(0)$. Substitute τ into equation (2.3) with $\tau(t) = 2(d(0) + v_r(0)t)/c$ and, after some manipulation [5], the phase term (we still assume the same receiver structure as previously described) can be expressed as:

$$\begin{aligned} \Phi(t) = 2\pi \left[\frac{2d(0)k}{c} \left(1 - \frac{2v_r(0)}{c} \right) t + 2\frac{v_r(0)}{c} f_c t \right. \\ \left. + \frac{2kv_r(0)}{c} \left(1 - \frac{v_r(0)}{c} \right) t^2 + \frac{2d(0)}{c} \left(f_c - \frac{kd(0)}{c} \right) \right], \quad (0 \leq t \leq t_p) \end{aligned} \quad (2.6)$$

To further simplify equation (2.6), the range beat frequency (f_b) and Doppler frequency (f_d) are defined as:

$$\begin{aligned} f_b &= \frac{2d(0)k}{c} \\ f_d &= \frac{2v_r(0)}{c} f_c \end{aligned} \quad (2.7)$$

In addition, a slowly moving object implies that we have $(1 - v_r(0)/c) \approx 1$ and $(1 - 2v_r(0)/c) \approx 1$, because $v_r(t)/c \approx 0$. By substitution of the range beat frequency and the Doppler frequency we get:

$$\begin{aligned} \Phi(t) \approx 2\pi \left[f_b t + f_d t \right. \\ \left. + \frac{2kv_r(0)}{c} t^2 + \frac{2d(0)}{c} \left(f_c - \frac{kd(0)}{c} \right) \right], \quad (0 \leq t \leq t_p) \end{aligned} \quad (2.8)$$

Equation (2.8) consists of four terms. From left to right they are a frequency term (range beat) which is proportional to range, another frequency term (Doppler shift) which is proportional to radial speed, a cross term that represents the range and Doppler coupling effect, and a constant phase term. A more detailed derivation can be found in [5].

Extending equation (2.8) to the multiple-chirp situation, the corresponding received baseband signal of each chirp contains the information of the range and velocity of the target at the start of each chirp. Specifically, assume N chirps are transmitted, and the chirps are indexed from $i = 1, 2, \dots, N$, the received baseband signal corresponding to each chirp can be expressed as:

$$\begin{aligned} \Phi(t) \approx 2\pi [& f_{bi}t + f_{di}t \\ & + \frac{2kv_r(it_p)}{c}t^2 \\ & + \frac{2d(it_p)}{c} (f_c - \frac{kd(it_p)}{c})], (it_p \leq t \leq (i+1)t_p) \end{aligned} \quad (2.9)$$

where f_{bi} and f_{di} is the beat frequency and Doppler frequency of the i -th received baseband signal:

$$\begin{aligned} f_{bi} &= \frac{2d(it_p)k}{c} \\ f_{di} &= \frac{2v_r(it_p)}{c} f_c \end{aligned} \quad (2.10)$$

Some algorithms, such as the two dimensional Fourier transform [6], use multiple chirps to compute the range and radial velocity. In our micro-Doppler classification problem, the spectrogram produced by multiple successive chirps is also more desirable than single chirp because when the measurement time is longer, the received signal reveals more information about the movement pattern of the target.

2.1.3 Moving target with rapidly changing velocity

In the last case, the target's velocity changes significantly during the sweep time t_p of the chirp, which is a result of non-zero acceleration. However, a long chirp can be divided (virtually) into multiple successive chirps which are short enough such that the assumption of constant velocity is valid. In other words, if there exists a maximum chirp duration t_{max} for the constant velocity assumption to hold but $t_p > t_{max}$, we can assume N successive chirps is transmitted such that $t_p/N \ll t_{max}$.

Each short chirp then sweeps from $f_c + iB/N$ to $f_c + (i + 1)B/N$ during the time interval from it_p/N to $(i + 1)t_p/N$. In figure 2.3, a long chirp is divided into four shorter chirps and each lasts $t_p/4$. The range resolution of an FMCW radar [7] is:

$$\delta d = \frac{c}{2B} \quad (2.11)$$

Because the long chirp is divided, the bandwidth B of each short chirp decreases as the range resolution δd increases although the total bandwidth is fixed.

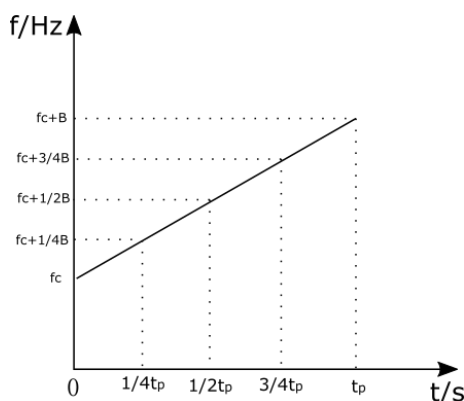


Figure 2.3: A long chirp can be divided into multiple successive short chirps to meet the constant velocity assumption.

2.2 Micro-Doppler model of complex non-rigid bodies

The models in the previous section apply to point targets. A point target is an idealized target that is useful in a mathematical model partly because we assume that it reflects the incident wave equally in all directions. Using a point target can simplify the radar model, but the objects of interest in practice have complex shape that scatters the incident wave non-uniformly. Non-rigid bodies such as humans and animals can also change their shape during the movement. The deformation of the body makes it difficult to simulate the scattered electromagnetic field. An approach to solve the problem is approximating a non-rigid body by a collection of connected

rigid bodies. In [8], the micro-Doppler signature of a walking human measured by a pulse radar is simulated. By approximating the limbs and the trunk of the human by ellipsoids of different sizes, the motion of different parts of human can be clearly identified from the spectrogram. In the high-frequency limit the backscattering of an ellipsoid is similar to a point scatterer in the sense that the backscattering is associated with a small area on the ellipsoid where the surface normally points towards the radar, which applies to metal ellipsoids in particular and to varying degree other materials that may approximate metal. However, the Radar Cross Section (RCS) of an ellipsoid depends on its size and the aspect angle to the radar. Here we analyzed the range and radial velocity contributions to the output signal of an FMCW radar for the point target. In the next chapter we will apply this model in combination with ellipsoids in order to simulate the micro-Doppler signature of a walking human.

2.3 Classification of micro-Doppler signature by machine learning

The problem of interest is to classify radar targets based on their micro-Doppler signatures. Related work in this area can be divided into two categories. In the first category, sophisticated pre-processing is first applied to the radar signal to extract certain features, and these features are then used to train some traditional classifiers such as a support vector machine (SVM). For example, Kim and Ling [9] used SVM on six features to classify seven activities of a human. The other category involves deep neural networks that usually require only very little pre-processing. Kim et al. [10] recognized seven gestures with the aid of a convolutional network (CNN). CNN also succeeded in human detection and human activities classification [11].

In this section, an overview of commonly used algorithms in related work is given.

2.3.1 Traditional machine learning (ML) algorithms

The workflow of traditional ML shown below is an iterative process. The step four to six are usually iterated until the performance for both the training set and the test set are satisfying.

1. Preprocess the raw data.
2. Divide the data into a training set and a test set.
3. Extract features from the data.
4. Select a classifier and a set of hyperparameters.
5. Train the classifier with the training set.
6. Validate the accuracy given the test set.

The first step is pre-processing the raw data recorded by the radar sensor. One way is to apply the short-time Fourier transform (STFT) to the raw data. The raw data can be further improved by clutter suppression algorithms such as notch filtering [12] and noise threshold [9].

To achieve high accuracy, a good feature extraction algorithm must be selected, which is the second step. A "good" feature extraction algorithm significantly reduces the dimension of the input data while keeping the important features for classifiers to work with. It has several advantages over directly feeding the raw data into the classifier. First, the raw data expressed as a spectrogram usually has a dimension of a few thousand, which easily results in an overfit when the training dataset is small. Second, the feature extraction saves computational resources by reducing the dimension of the data.

After feature extraction, the workflow enters an iterative process in which a classifier is selected and trained while the hyperparameters are tuned in order to achieve a good performance. In the following sections, a few feature extraction algorithms and classifiers will be briefly introduced.

2.3.2 Feature extraction algorithms

Sometimes people need to deal with high dimensional data such as images. Feature extraction algorithms can extract necessary features from the high dimensional data and, thus, form a low dimensional representation of it. Feature extraction is necessary when the classifier suffers from high dimension of input data and the number of training data is small. Features can be either handcrafted or learned by ML algorithms. For example, six physical features are designed and extracted from the spectrogram of human activities [9] for classification. This is an example of handcrafted feature extraction. Apparently, it requires a lot of domain knowledge to design such features to represent the original data. ML algorithms such as the principal component analysis (PCA) and the linear discriminant analysis (LDA) are two linear algorithms that do not require much understanding of the data. They are commonly used to solve problems such as image classification and handwriting recognition. Jingli et al. extend the PCA and the LDA to 2-dimensional spectrogram and use SVM as a classifier to classify human activities [13]. The PCA and the LDA are described in this subsection while the detailed derivation of the equations can be found in chapter 15 and chapter 16 of [14].

2.3.2.1 Principal component analysis (PCA)

PCA projects high dimensional vectors to a lower dimensional space. The projection to this lower dimensional space is determined by the dataset. The algorithm is unsupervised, i.e. no class label is needed.

Suppose the dataset contains N data points (\mathbf{x}^n , $1 \leq n \leq N$) and the data points are D dimensional real vectors. The PCA projects \mathbf{x}^n to a M -dimension vector \mathbf{y}^n by:

$$\mathbf{y}^n = \mathbf{B}\mathbf{x}^n + \mathbf{c}_1 \text{ for } 1 \leq n \leq N \quad (2.12)$$

where \mathbf{c}_1 is a constant bias vector. The projection matrix \mathbf{B} which is orthogonal ($\mathbf{B}\mathbf{B}^T = \mathbf{B}^T\mathbf{B} = \mathbf{I}$) and the bias vector are selected to minimize the square distance

error between the reconstructed data point $\tilde{\mathbf{x}}^n$ and the original data point:

$$\begin{aligned}\tilde{\mathbf{x}}^n &= \mathbf{B}^T \mathbf{y}^n + \mathbf{c}_2 \\ \mathcal{L}(\mathbf{B}, \mathbf{Y}, \mathbf{c}_2) &= \sum_{n=1}^N \sum_{j=1}^D (\tilde{x}_j^n - x_j^n)^2 \text{ (Loss function to minimize)}\end{aligned}\quad (2.13)$$

$$\mathbf{Y} := [\mathbf{y}^1, \dots, \mathbf{y}^N]$$

where \mathbf{c}_2 is another constant bias vector.

It turns out that the projection matrix \mathbf{B} and the bias vectors \mathbf{c}_1 , \mathbf{c}_2 are closely related to the mean and the covariance matrix of the dataset. The PCA can be conducted as follows:

1. calculate the mean vector and the covariance matrix of the dataset:

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}^n, \quad \mathbf{S} = \frac{1}{N-1} \sum_{n=1}^N (\mathbf{x}^n - \mathbf{m})(\mathbf{x}^n - \mathbf{m})^T \quad (2.14)$$

2. conduct eigenvalue decomposition on \mathbf{S} to get eigenvectors $\mathbf{e}_1, \dots, \mathbf{e}_M$ and eigenvalues $\lambda_1, \dots, \lambda_M$ (sorted in descending order with respect to eigenvalues), discard the rest of eigenvectors and eigenvalues.
3. form the matrix as $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_M]$, and the projection function is given by:

$$\mathbf{y}^n = \mathbf{E}^T (\mathbf{x}^n - \mathbf{m}) \quad (2.15)$$

4. the reconstruction function is:

$$\tilde{\mathbf{x}}^n = \mathbf{E} \mathbf{y}^n + \mathbf{m} \quad (2.16)$$

2.3.2.2 Linear discriminant analysis (LDA)

The LDA is a type of supervised feature extraction algorithms that utilize the labels of the data points. The purpose of the LDA is to find a projection \mathbf{W} (dimension $D \times M$ such that the separation of projected data points $\mathbf{y} = \mathbf{W}^T \mathbf{x}$ within the same class is minimized while the separation of projected data points from different classes is maximized.

Assume the dataset contains C classes and that each class has N_c data points, where $1 \leq c \leq C$. Let \mathcal{X}_c denote the subset that contains all the data points from a class c , where the mean \mathbf{m}_c and the covariance matrix \mathbf{S}_c of each subset are calculated similarly to the equation (2.14). The algorithm is conducted as follows:

1. compute the between-class scatter matrix \mathbf{A} :

$$\mathbf{A} = \sum_{c=1}^C N_c (\mathbf{m}_c - \mathbf{m})(\mathbf{m}_c - \mathbf{m})^T \quad (2.17)$$

where \mathbf{m} is the mean of the whole dataset

2. compute the within-class scatter matrix \mathbf{B} :

$$\mathbf{B} = \sum_{c=1}^C N_c \mathbf{S}_c \quad (2.18)$$

3. compute the *Cholesky factor* of \mathbf{B} , denote it as $\tilde{\mathbf{B}}$
4. compute the M principal eigenvectors of $\tilde{\mathbf{B}}^{-T} \mathbf{A} \tilde{\mathbf{B}}^{-1}$ as $\tilde{\mathbf{W}}^T = [\mathbf{e}_1, \dots, \mathbf{e}_M]$
5. the projection matrix is $\mathbf{W} = \tilde{\mathbf{B}}^{-1} \tilde{\mathbf{W}}^T$

2.3.3 Classifiers

A classifier can predict the label of a novel input based on the training data. In the study of Jiajin et al. [15], three kinds of classifiers (k-nearest neighbors, support vector machine and Bayes linear classifier) are used to classify simulated radar data. The first two classifiers are commonly used and usually perform better than Bayes linear classifier, so they are discussed in this section. The detailed discussion can be found in chapter 14 and chapter 17 of [14].

2.3.3.1 K-nearest neighbors (k-NN)

The k-NN assigns a label to a new input based on the k nearest training data to the input by counting the number of neighbors belonging to each class. Here, the k is a

hyperparameter that is selected empirically in cross-validation. Define the training set as $\mathcal{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ and the corresponding labels are $y^1, \dots, y^N \in \{1, \dots, C\}$, the k-NN algorithm is as follows:

1. compute the distance between the input \mathbf{x} and every training data points $d^n = d(\mathbf{x}, \mathbf{x}^n)$, for $n = 1, \dots, N$
2. select the k nearest training data points, and count the number of data points belonging to each class as l_1, \dots, l_C
3. assign the input to the class $y = \operatorname{argmax}_y l_y$

The distance function can be Euclidean distance or other distance such as *Mahalanobis* distance [14].

2.3.3.2 Support vector machine (SVM)

For simplicity, we introduce the SVM with hard decision margin first. Suppose the same notation of training set as k-NN, but the labels are binary: $y^1, \dots, y^N \in \{-1, 1\}$. The SVM finds two hyperplanes $\mathbf{w}^T \mathbf{x} + b = \pm 1$ that separate the two classes while maximizes the distance (or the margin $\frac{2}{|\mathbf{w}|}$) between the two hyperplanes. To find the weights \mathbf{w} and the bias b , a quadratic optimizing problem needs to be solved:

$$\begin{aligned} & \text{minimize } \frac{1}{2} |\mathbf{w}|^2 \\ & \text{subject to: } y^n (\mathbf{w}^T \mathbf{x}^n + b) \geq 1, n \in \{1, \dots, N\} \end{aligned} \tag{2.19}$$

A more practical SVM is the one with soft margin, which allows some mislabeling to happen in the training data. It is useful because the training data is not always linearly separable. Then, the optimization problem becomes:

$$\begin{aligned} & \text{minimize } \frac{1}{2} |\mathbf{w}|^2 + \frac{1}{2} C \sum_n (\xi^n)^2 \\ & \text{subject to: } y^n (\mathbf{w}^T \mathbf{x}^n + b) \geq 1 - \xi^n, \xi^n \geq 0, n \in \{1, \dots, N\} \end{aligned} \tag{2.20}$$

or,

$$\begin{aligned} & \text{minimize } \frac{1}{2}|\mathbf{w}|^2 + C \sum_n \xi^n \\ & \text{subject to: } y^n(\mathbf{w}^T \mathbf{x}^n + b) \geq 1 - \xi^n, \xi^n \geq 0, n \in \{1, \dots, N\} \end{aligned} \tag{2.21}$$

Equation (2.21) and equation (2.20) define the 1- and 2-norm soft-margin SVM separately. The constant C is a hyperparameter that controls how much mislabeling is allowed. The slack variable ξ^n is the distance between \mathbf{x}^n and the correct margin. The optimization problems can be easily solved and there are many existing computer implementation. Assume (\mathbf{w}^*, b^*) is the optimal solution. Then, the input data is assigned to the class $\text{sgn}(\mathbf{w}^{*T} \mathbf{x} + b^*)$.

2.3.4 Deep learning algorithms

Strictly speaking, deep learning (or deep neural network) belongs to ML algorithms. The word "deep" means that the number of layers of a neural network is large. One important reason to develop and use deep learning is that it can learn and represent highly complex functions for data in a high dimensional space, where traditional ML often fails [16]. In this section, we give an overview of some basics of neural networks beginning with the simplest case, which is logistic regression.

2.3.4.1 Logistic regression

Given the input vector \mathbf{x} , the weights \mathbf{w} and bias b , the logistic regression calculates the following output y :

$$\begin{aligned} a &= \mathbf{w}^T \mathbf{x} + b \\ \hat{y} &= \sigma(a) = \frac{1}{1 + e^{-a}} \end{aligned} \tag{2.22}$$

The sigmoid function $\sigma(a)$ maps any real number to $(0, 1)$. In a binary classification problem with $y \in \{0, 1\}$, one can interpret the output of sigmoid as the likelihood

$\hat{y} = p(y = 1|\mathbf{x}; \mathbf{W}, \mathbf{b})$. The loss function is defined as the cross-entropy of the prediction \hat{y} and the ground truth y :

$$\mathcal{L}(\hat{y}, y) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \quad (2.23)$$

while the cost function of N training examples is:

$$\mathcal{J}(\mathbf{w}, b) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(\hat{y}^i, y^i) \quad (2.24)$$

where the superscript i denotes the i -th training example. To train the model, a gradient descent method with learning rate α is used to find (\mathbf{w}^*, b^*) that minimize the cost function:

$$\begin{aligned} \mathbf{w} &:= \mathbf{w} - \alpha \frac{\partial \mathcal{J}}{\partial \mathbf{w}} \\ b &:= b - \alpha \frac{\partial \mathcal{J}}{\partial b} \end{aligned} \quad (2.25)$$

2.3.4.2 Feedforward neural networks

Feedforward neural networks (also known as fully connected neural networks) have multiple layers including one input layer, one output layer and several hidden layers. For each hidden and output layer, a linear operation is first applied to the output from the previous layer and the activation is calculated and fed to the next layer:

$$\begin{aligned} \mathbf{h}^{[l]} &= \mathbf{W}^{[l]} \mathbf{a}^{[l-1]} + \mathbf{b}^{[l]} \\ \mathbf{a}^{[l]} &= g(\mathbf{h}^{[l]}) \end{aligned} \quad (2.26)$$

where the superscript $[l]$ denotes the values associated to l -th layer. In equation (2.26), the parameters to be learned is the weight matrices $\mathbf{W}^{[l]}$ and the bias vectors $\mathbf{b}^{[l]}$ of every layer. The function $g(\mathbf{h})$ is a nonlinear activation function

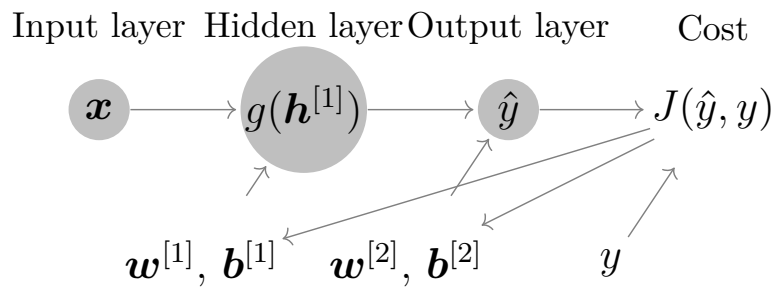


Figure 2.4: A feedforward neural network with one hidden layer.

that calculates the element-wise activation of \mathbf{h} . In figure 2.4, a feedforward neural network with one hidden layer is illustrated.

Similarly to logistic regression, the weights and biases are updated by gradient based optimization method to minimize the cost function during training. The training process is called backpropagation. Depending on the application, the activation function of the output layer is not limited to sigmoid function (suitable for binary classification). For example, in a multiple-class classification problem the activation at the output can be soft-max function whose output is a vector. The elements of the output vector of soft-max function are the prediction of the likelihood that the input belongs to each class.

2.3.4.3 Convolutional neural networks

The basic CNN consists of three kinds of layers: convolution layers, pooling layers and fully connected (feedforward) layers. The input data is often called "volume" because it is a three-dimensional array ($m \times n \times c$). During convolution step, the volume is convolved with p filters, each has a dimension of $f \times f \times c$. Much like the convolution operation in one-dimensional space, each filter slides through the input volume in the first two dimensions and produces a two-dimensional array. After stacking the output p filters and applying element-wise activation function, the 2-D convolution operation is done.

The pooling layer reduces the dimension of the volume. It is applied to each channel

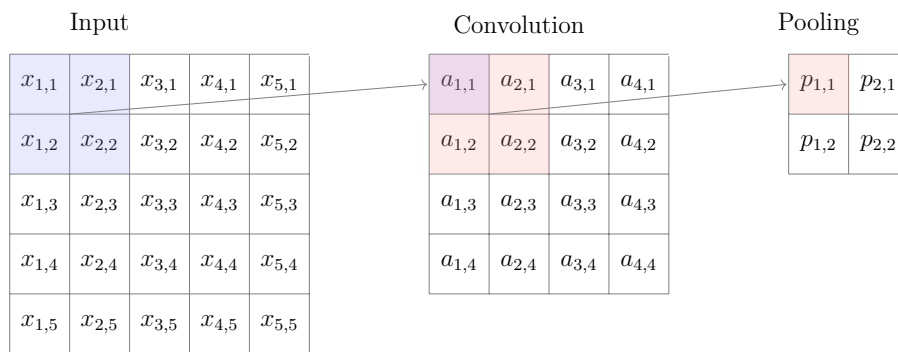


Figure 2.5: An example of a convolutional layer and a pooling layer.

separately, and the element of the output 2-D array is the average or the maximum value of a portion of the input. After some convolution and pooling layers, the volume is flattened and fed into a feedforward neural network introduced previously.

To illustrate the process further, assume that the input \mathbf{x} is a 2-D array (single channel). It is first convolved with a 2×2 filter \mathbf{w} , and a 2×2 max pooling layer follows. As figure 2.5 shows, the convolution operation takes a portion of \mathbf{x} which has the same size as the filter and computes the sum of the element-wise multiplication of that portion and the filter, followed by a non-linear activation $g(\cdot)$:

$$a_{11} = g\left(\sum_{i=1,2} \sum_{j=1,2} w_{ij}x_{ij}\right) \quad (2.27)$$

The rest of the elements are obtained similarly by shifting the filter within the input volume. The second step is max pooling, which puts a sliding window on the volume and selects the maximum value during sliding.

An important property of CNN is parameter sharing. Note that the same filter is applied to multiple portions of the input data in CNN while in feedforward network each element in the input data is multiplied by a different factor. This allows the CNN to be trained by fewer data and the number of parameters in the neural network is significantly reduced compared to feedforward neural networks.

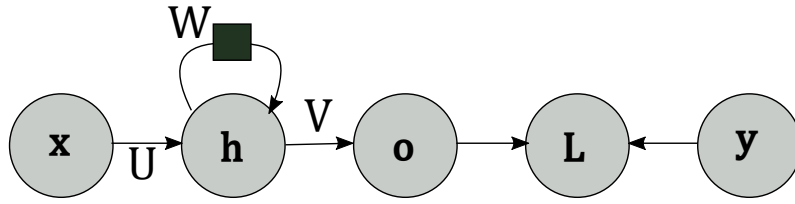


Figure 2.6: The computational graph of a simple RNN with one hidden layer and a feedback connection of the hidden layer to itself (biases not shown).

2.3.4.4 Recurrent neural networks

Recurrent neural networks (RNN) have succeeded in processing sequential data such as natural language. RNN has very flexible structures, and we select an example shown in figure 2.6 to illustrate how it works. The recurrent connection with a solid square means that the hidden unit is connected to itself at the next time step. The weight matrices are denoted as \mathbf{U} , \mathbf{V} and \mathbf{W} . Denote the variables at time step t by adding a superscript (t) . Then, the output vector $\mathbf{o}^{(t)}$ is calculated from the input $\mathbf{x}^{(t)}$ and the previous hidden unit $\mathbf{h}^{(t-1)}$ as follows:

$$\begin{aligned}\mathbf{a}^{(t)} &= \mathbf{b} + \mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} \\ \mathbf{h}^{(t)} &= g(\mathbf{a}^{(t)}) \text{ (activation)} \\ \mathbf{o}^{(t)} &= \mathbf{c} + \mathbf{V}\mathbf{h}^{(t)} \text{ (output)}\end{aligned}\tag{2.28}$$

where \mathbf{b} and \mathbf{c} are biases, and $g()$ denotes the activation function, and the loss at time t is $L^{(t)}$. The loss function is selected based on the application.

2.3.4.5 Optimization and regularization of neural networks

Designing the architecture of a neural network is one piece of the puzzle, while training it can be difficult sometimes, especially when the network is very deep. Many optimization methods have been proposed to accelerate the optimization. Adam [17] is an effective method among them. Another problem arises when the neural network is well optimized on a training set but the performance is poor on

a test set. In this case, regularization methods such as drop out [18] and L^1 , L^2 regularization [16] are useful to prevent the model from overfitting.

3

Hardware Description

3.1 FMCW radar evaluation kit introduction

The hardware used for the measurement campaigns in this thesis is the RS3400K/00 programmable radar development kit from *Sivers IMA*. The kit includes a transceiver, a control board and a horn antenna. This section first presents some important parameters of each component of the development kit. Second, each component is briefly described. Two important limitations of the radar kit and how the limitations affect the measurement are discussed in the following section 3.2.

In table 3.1, some important parameters the radar are listed.

Table 3.1: RS3400K/00 24 GHz FMCW Transceiver Evaluation Kit Specification.

Parameter	Unit	
RS3400K/00 FMCW transceiver module		
Carrier frequency	24.75	GHz
Bandwidth	1.5	GHz
AN1020K/00 antenna		
Gain	20	dB
Bandwidth	22 - 33	GHz
H-plane 3 dB beamwidth @24 GHz	18.6	degree
E-plane 3 dB beamwidth @24 GHz	16.1	degree
CO1000A/00 power and controller board		
Power supply	12	V
Connection	Serial	
Sample rate	up to 20	KHz

3.1.1 Hardware connection

As shown in figure 3.1, the transceiver module (RS3400/00) is connected to the power and controller board through two dual-row PCB headers. The antenna (AN1020K/00) can be connected to the transceiver either through a male-to-male SMA connector or a coaxial cable. In this project, the antenna is attached to the board through a male-to-male SMA connector. The power and controller board (CO1000A/00) requires a 12V DC power supply. Signals can be transmitted to a computer through RS232 serial/USB cable.

3.1.2 Antenna

The radar is equipped by a horn antenna. The antenna has a rather narrow beam width and, thus, the distance between the radar and the target must be sufficiently

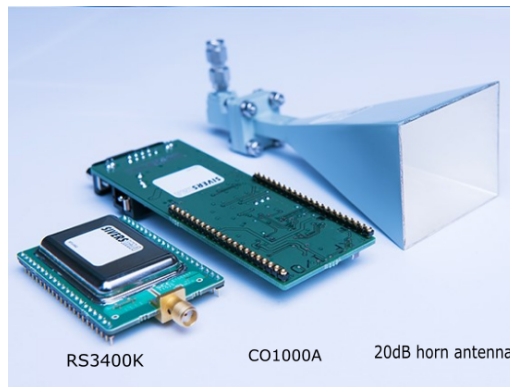


Figure 3.1: The radar kit includes three parts: a transceiver board (RS2400K), a power and controller board (CO1000A), and a horn antenna with a gain of 20dB.

large in order to illuminate the target completely with its main lobe. For example, the distance between a typical bike wheel (36 cm in radius) and the radar should be more than 3 m.

3.1.3 Transceiver

The transceiver is a compact module that integrates radio frequency circuit and a microcontroller for executing the commands of the control board. The module has a default center frequency at 24.75GHz. The bandwidth can be set from 0 to 1.5 GHz. This frequency range is actually within the ultra-wide bandwidth (UWB) of the 24 GHz radar. According to spectrum regulations and standards developed by the European Telecommunications Standards Institute (ETSI) and Federal Communications Commission (FCC), by the year 2022 in both Europe and the USA, the use of the UWB band will be phased out. The central frequency should be limited in the Narrow-Band (NB) from 24.05 - 24.25 GHz. In addition, a relatively high emission power up to 20 dBm is allowed with a bandwidth up to 250 MHz[19].

3.1.4 Power and controller board

The power and controller board is used to power the transceiver module. Also, it has a microcontroller for the FMCW frequency sweep control of the transceiver. It also has a 10-bit ADC to sample the analog signal from the transceiver.

A computer is needed to conduct the radar measurement. Communication between the computer and the control board occurs over RS-232. Human-readable commands could be used to set up desired parameters. Commonly used commands have several main categories. They are listed in table 3.2. Each category has sub-categories. INIT is used to initialize the transceiver to the previous setting rather than the default values. FREQUENCY and SWEEP are used to define the transmitted signal. Some parameters in these categories are relative to each other, e.g. start/stop frequency and bandwidth, and this implies that a modification in one parameter might result in changes of the other parameters. TRIG is used to define the trigger method of the transceiver and to put the device in a ready state for measurement. TRACE is used to return measurement result to the computer.

Command categories	Function
INIT	Initialize the transceiver.
FREQUENCY	Control frequency parameters, e.g. bandwidth.
SWEEP	Control chirp parameters, e.g. duration, numbers.
TRIG	Control trigger parameters
TRACE	Return measurement data
HELP	Provide a simple list of available commands

Table 3.2: Command categories for radar controller.

3.2 Hardware limitations

The radar kit has some unexpected features that we need to consider while designing data measurement campaign.

First, the measurement is not efficient. In the theoretical model, we described a high definition radar which can transmit successive short chirps. This allows the measurement to clearly show the change in range beat and Doppler frequency. However, the radar kit is not able to work in that way. The user needs to send a measurement command to the control board for each chirp. After the measurement command is received by the control board, it starts to measure and write measurement samples to its buffer. The buffer can store at most 1501 numeric samples. The control board returns the samples to a computer through the serial interface after the measurement is finished, and the last step takes a very long time (about one second for 1501 samples) as compared to the chirp duration (5 ms). More importantly, it is not possible to send another measurement command during this process. This delay caused by the serial communication makes it impossible to transmit and receive successive short chirps. If we were to transmit 256 5-ms chirps, it could take about 18.34 seconds to complete the data acquisition, where only 1.28 seconds are effective while the rest of the time is merely spent on communication overhead. In this case, the measured spectrogram is not able to show the continuous movement of a target.

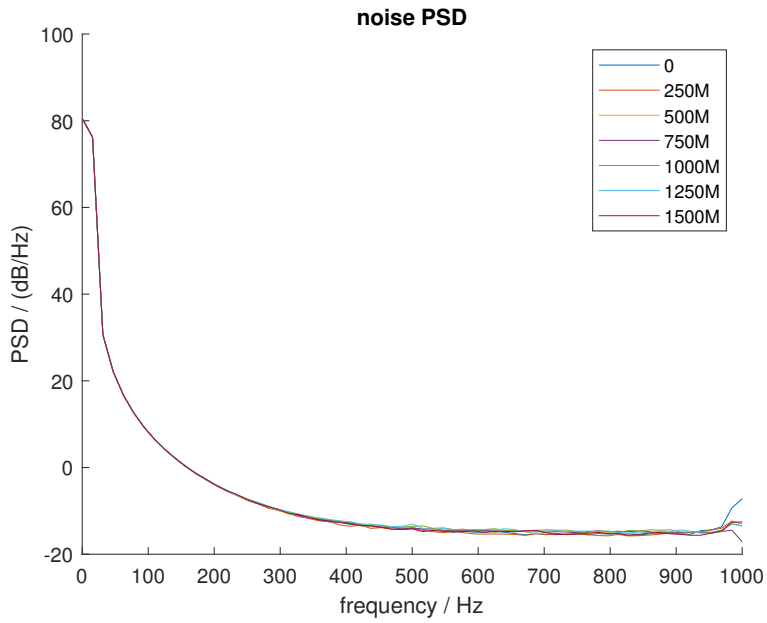
Second, the actual frequency sweep of the chirp is not a sawtooth waveform but a staircase approximation of the linear frequency sweep associated with the sawtooth. Thus, the radar transmits a sinusoidal signal with a piecewise constant frequency that is increased in multiple steps. The radar allows the user to set the duration of the sinusoidal signal for each time interval with constant frequency, (called *sweep:idle* according to the user documentation). Another parameter is *freq:points* that corresponds to the number of time intervals with constant frequency. The radar returns *freq:points* samples to the computer, so the chirp duration equals the product of *sweep:idle* and *freq:points*.

Last, measurements show that the radar returns a baseband signal with a strong DC component and some noise in its low frequency band. The amplitude and the occupied bandwidth of noise increase as the bandwidth increases. The noise measurement was conducted on the top floor of a seven story building. During the measurement, the antenna was pointed to the sky and 50 chirps were transmitted for

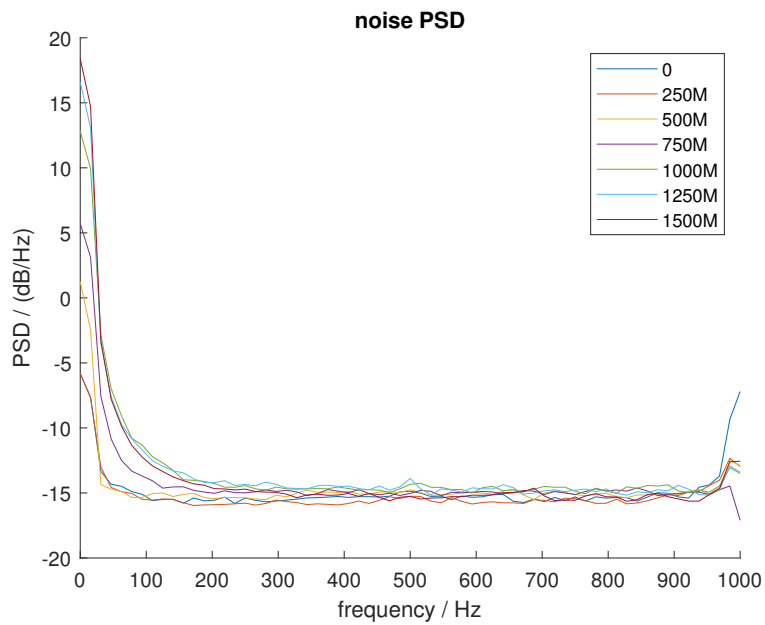
3. Hardware Description

each bandwidth. In figure 3.2a, the noise power spectrogram density (PSD) shows little difference below 400 Hz when different values for the bandwidth is used. After the DC component is suppressed (shown in figure 3.2b), the noise at the higher frequency range (400 - 900 Hz) is about -15 dB/Hz. However, as the bandwidth increases, the noise amplitude at low frequency significantly increases. Increasing the bandwidth can improve the range resolution of the radar but it also introduces noise that can potentially overwhelm the Doppler frequency induced by slow-moving target. Here, we need to make a compromise when we select the bandwidth. High bandwidth leads to high range resolution but, unfortunately, it also results in lower SNR for the low frequency band.

Due to the reasons mentioned above, we finally choose a 750 ms radar chirp with 250 MHz bandwidth as listed in table 4.1 in the next chapter. Figure 3.3 shows the actual signal transmission model of the radar kit.



(a) Noise PSD (raw data)



(b) Noise PSD (mean subtracted)

Figure 3.2: Noise power spectrogram density (PSD) measurement of the radar kit.

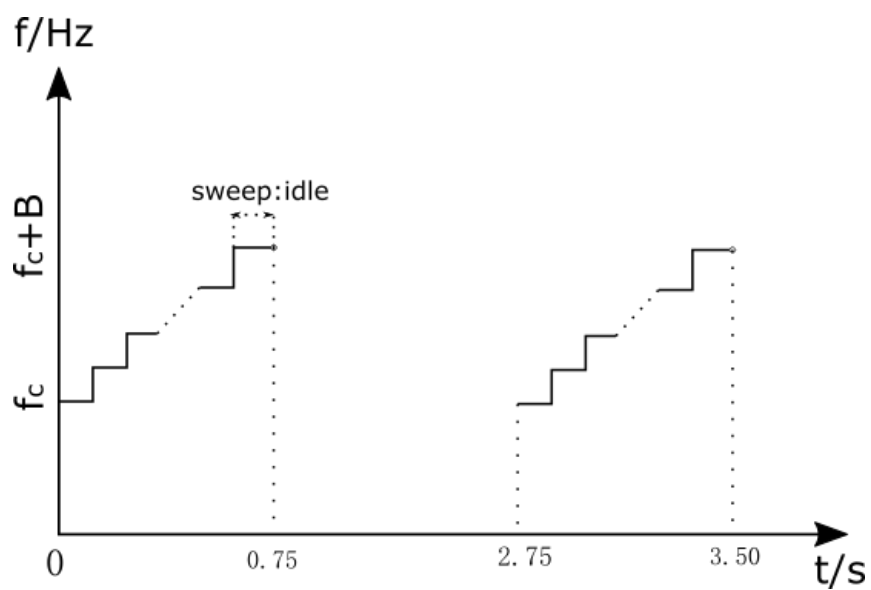


Figure 3.3: The actual signal transmission model of the radar kit. Two sawtooth chirps (approximated by staircase waveform) are plotted. Dashed line in each chirp represents the omitted steps, and one second gap is in between of two chirps due to serial communication.

4

Modelling and Simulation

4.1 Introduction

Theoretical modelling and simulation provide insights into the micro-Doppler phenomenon. A radar propagation model with a non-rigid target is difficult to construct. To simplify the problem, a human body can be modeled as jointly connected rigid parts which is described in section 4.5. To break down the problem even further, the simulation of a single rigid body, like a bicycle wheel, is shown in section 4.4. Some samples of a rotating bike wheel and walking human were measured in the laboratory. The measurement results are demonstrated after the simulation. But at the very beginning of the work, a point target is used to evaluate the radar model in section 4.3.

Although the radar sensor's limitations in noise performance and resolution cause difference between simulation and measurement, the theoretical model is still a good way to help us to understand how FMCW radar measures micro-Doppler signature.

4.2 Simulation and measurement settings

Table 4.1 lists three different sets of parameters that are used in this thesis, and they are labeled as $S1$, $S2$ and $S3$. Here, $S1$ is the ideal (high resolution) setting used in the simulation of the oscillating point, the bicycle wheel and the walking

human. $S2$ is the low resolution setting, which is similar to the actual hardware setting used in the measurement. $S3$ is the setting of the radar hardware used in the measurement. These settings are used in the simulation and measurement in the rest of the chapter.

Symbol	Meaning	Unit	S1 ¹	S2 ²	S3 ³
t_p	Chirp duration	ms	5	5	750
f_c	Carrier freq.	GHz	24	f_{cS2} ²	24
B	Bandwidth	MHz	250	1.67	250
N	# of tx chirps	/	256	600	2
f_s	Sample rate	kHz	20	1.5	2
δd	range res.	m	0.6	90	/
/	waveform	/	sawtooth	sawtooth	staircase

¹ S1 is the ideal (high resolution) setting used in the simulation corresponding to figure 2.1. The radar transmits successive 5-ms sawtooth chirps with this setting.

² S2 is the low resolution setting corresponding to figure 2.3. The radar transmits four long (750 ms) sawtooth chirps, and each has a bandwidth of 250 MHz. Each long chirp is simulated by 150 short chirps with a bandwidth of 1.67 MHz and a duration of 5ms. The carrier frequency of short chirps changes periodically, i.e., $f_{cS2} = 24 + 1.67 \times 10^{-3}n$ ($0 \leq n \leq 149$).

³ S3 is the hardware setting used in the measurement corresponding to figure 3.3. The radar transmits two 750-ms staircase chirps, and a gap between the chirps exists due to serial communication. The exact range resolution cannot be calculated from equation 2.11 because of the staircase waveform.

Table 4.1: Radar parameters for simulation and measurement.

4.3 Point target

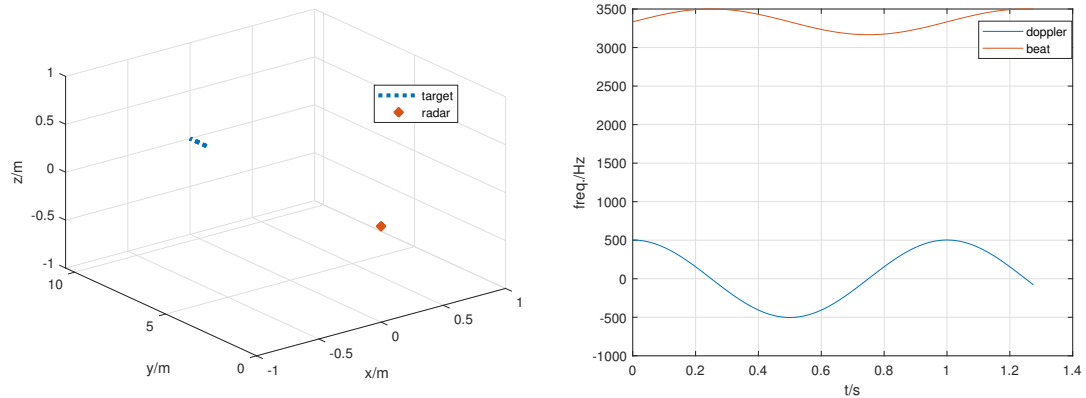
4.3.1 Simulation

An moving point target is used to simulate the radar model described in section 2.1. Here, we assume that the chirp duration t_p is small and multiple successive chirps are transmitted. The radar has an isotropic antenna and the point target moves with a sinusoidal velocity along the y-axis. The position of the target is $\mathbf{R}_t = [0, r_0 + A \cdot \cos(2\pi \cdot f_m t), 0]$ given the oscillation frequency $f_m = 0.5$ Hz and the amplitude $A = 1$ m, where the average position is $r_0 = 10$ m. To illustrate that the relative position of the target and the radar significantly influence the micro-Doppler signature, the radar will be placed at $[0, 0, 0]$ (target moving *towards* and *away* from the radar) and $[r_0, r_0, 0]$ (moving *tangentially* to the radar). The radar is assumed to be operated with a short chirp duration and a high sampling rate, where the parameters are listed as *S1* in table 4.1.

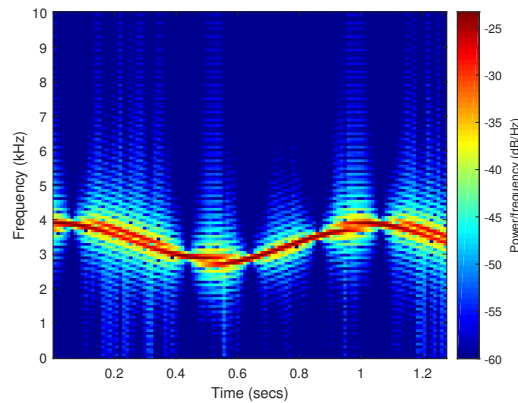
Here, the parameters r_0 and f_m are chosen in accordance with the measurements that will be discussed later. The distance between the target and the radar is 10 m in the simulation, while the distance in measurement is around 8.5 m. The $f_m = 0.5$ Hz can represent the typical frequency of a human activity such as arm swinging while walking. The three sub-figures in figure 4.1 show the situation when the point target and the radar are both on the y-axis. This allows the point target to produce the strongest variation in Doppler and range beat. Sub-figure 4.1a is the trajectory of the point target in relation to the radar, while sub-figure 4.1b is the variation of the range beat frequency and the Doppler frequency, and 4.1c is the spectrogram. The trace with the strongest amplitude in the spectrogram is approximately the sum of the two frequencies.

When the radar is re-placed at $[r_0, r_0, 0]$ and the distance of the point target relative to the radar remains approximately constant, as shown in sub-figure 4.2. The radial speed and variation in the distance are significantly reduced. As a result, the trace

4. Modelling and Simulation



(a) Trajectory of the target (blue dashed line). (b) Range beat (red) and Doppler frequency (blue).



(c) Spectrogram of the baseband signal.

Figure 4.1: Point target moves straight towards and away from the radar.

in the spectrogram becomes almost a straight line and it corresponds to a target at an approximately constant distance with an approximately zero radial velocity.

4.4 Bicycle wheel

From the theoretical simulation of a point target with the radar placed at different positions, we know that the micro-Doppler signature is highly dependent on the incident angle of the radar signal to the moving target. A bicycle wheel is a useful test case since it can produce periodic movement of the spokes as the wheel is

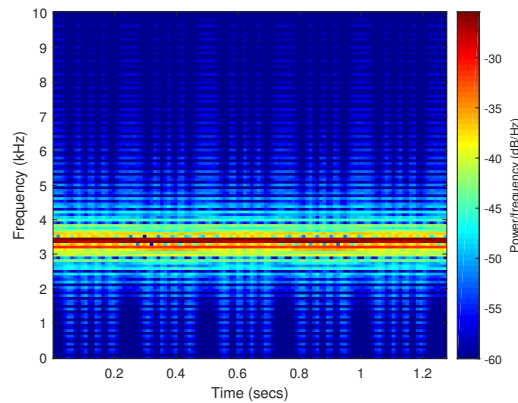
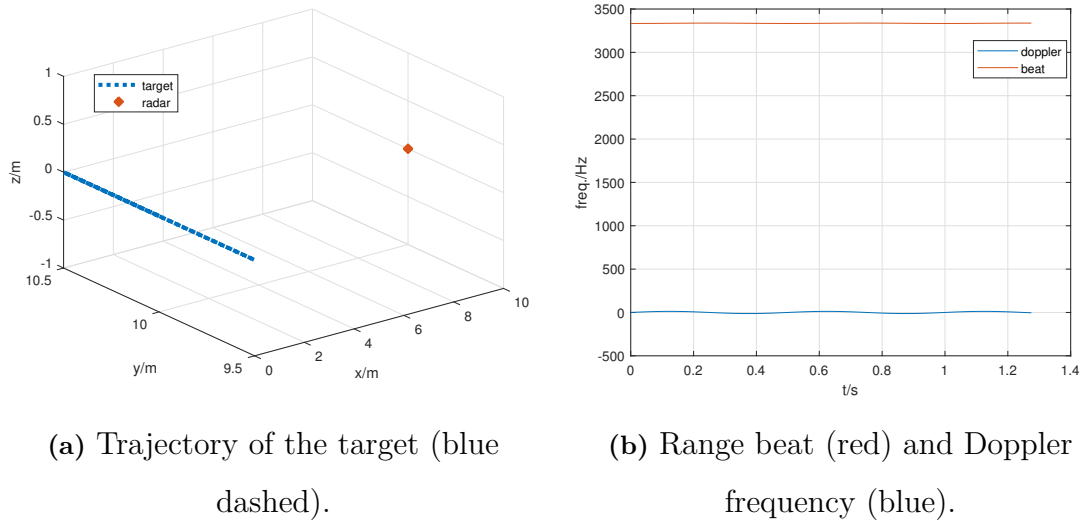


Figure 4.2: Point target oscillates tangentially to the radar.

rotating, where the bike spokes may be coarsely modeled by individual point targets (inspired by Victor’s study on helicopter engine [2]).

4.4.1 Simulation

Compared to the point target, a bicycle wheel could generate a more complex micro-Doppler signature. All the radar simulation parameters are identical to the case with the point target in the previous section, where these are listed as $S1$ in table 4.1. The spokes of the wheel are modeled as N_w point targets that rotate in the x-y

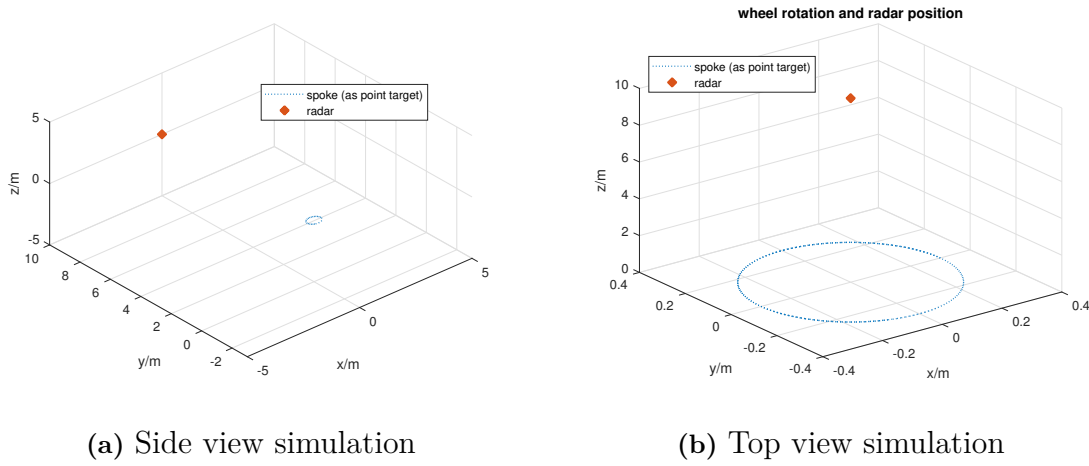


Figure 4.3: The positions of the radar and the bike wheel in two situations. Due to the axis settings, the size of the wheel looks different in both figures.

plane and around the z -axis with radius $r_w = 0.3$ m. The position of each point ($\mathbf{R}_{wi}(t)$) can be described as:

$$\begin{aligned} \mathbf{R}_{wi}(t) &= [r_w \cos(\Phi_i(t)), r_w \sin(\Phi_i(t)), 0] \\ \Phi_i(t) &= \omega t + 2\pi i/N_w, \quad (0 \leq i \leq N_w - 1) \end{aligned} \quad (4.1)$$

where ω is the angular speed (a typical bike wheel rotates at 80 rpm, which is about 8.38 rad/s) and r_w is 0.3m. To show how the micro-Doppler signature changes with the aspect angle radar, the radar is placed on the y -axis ($[0, r_0, 0]$) or on the z -axis ($[0, 0, r_0]$). Where r_0 is also 10 meters. Figure 4.3 shows these two situations.

First, consider the case when the radar is located on the y -axis, so the variation in the radial speed and in the distance of the spokes is at maximum. Figure 4.4 shows the spectrograms of the wheel as the number of spokes increases. When there is one spoke rotating, the spectrogram looks like a sinusoidal curve. As the number of spokes increases, more sinusoidal curves appear with different phases. For 8 spokes, it is difficult to visually distinguish the corresponding eight sinusoids in the spectrogram.

Second, when the radar is placed on the z -axis the variation of radial speed is zero

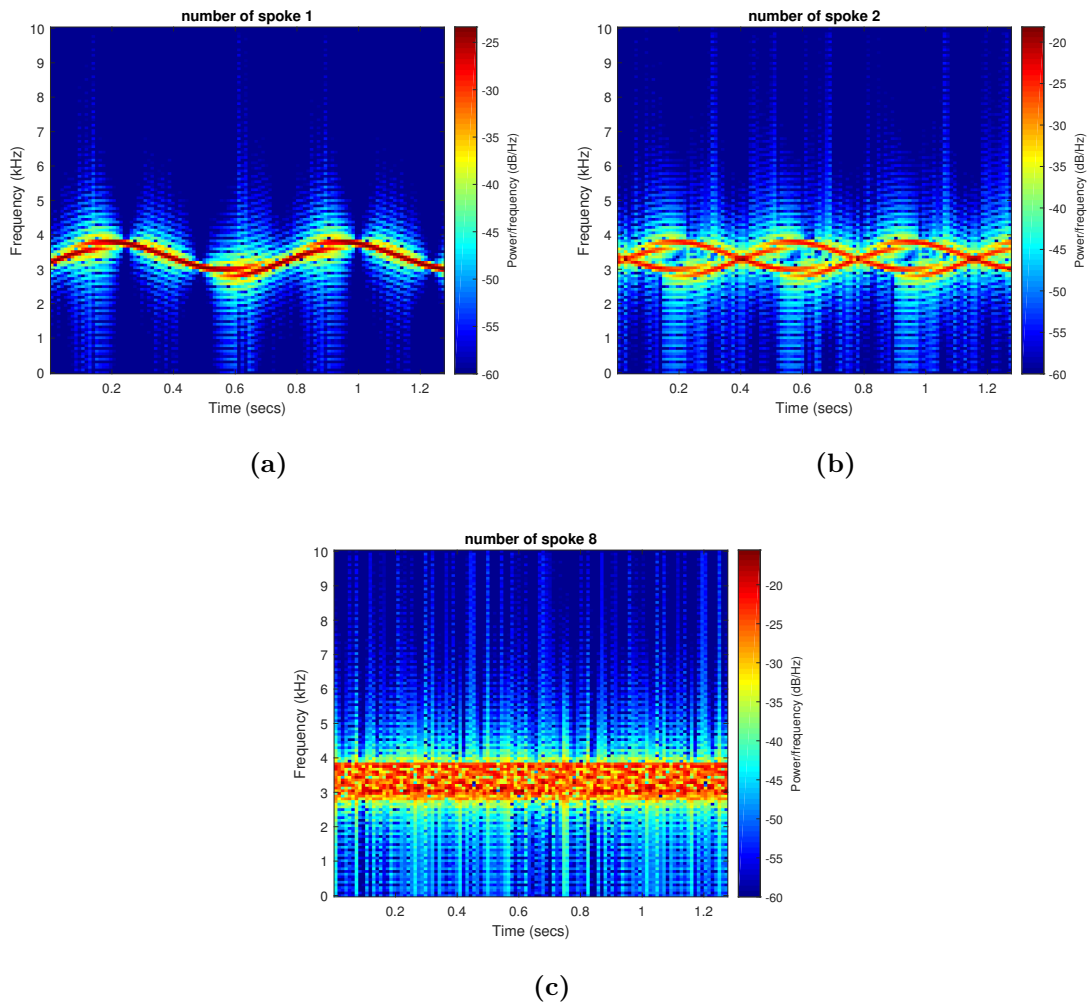


Figure 4.4: Spectrograms of the wheel when the number of spokes is (a) one, (b) two, or (c) eight.

and the distance is constant, which gives a flat curve in the spectrogram as shown in figure 4.5. No matter how many spokes there are, the result for this radar position is similar.

4.4.2 Experimental measurement

The radar setting $S3$, as listed in table 4.1, is used in the measurement. A bicycle with two wheels (36 cm in radius) was placed up-side-down in front of the radar kit in the experiment (shown in figure 4.6a). Unlike the free space environment used in the

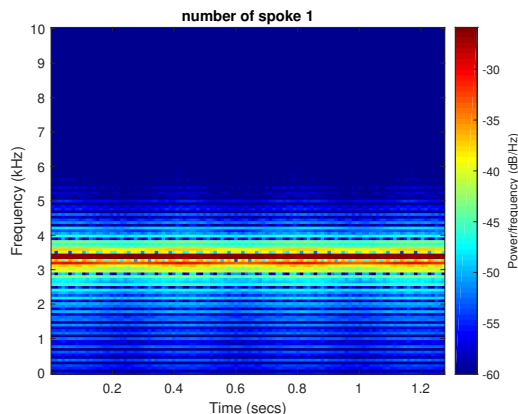
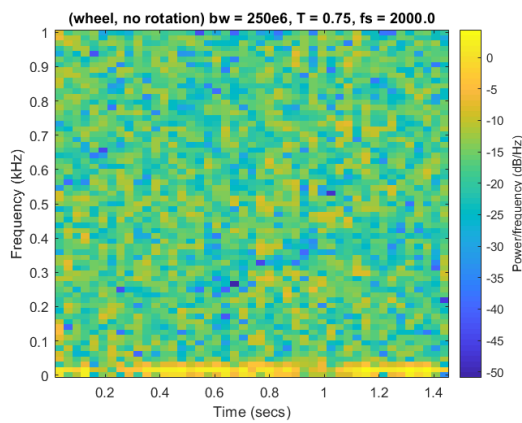


Figure 4.5: Spectrogram of the wheel when the radar is placed at the z-axis. The wheel has one spoke.

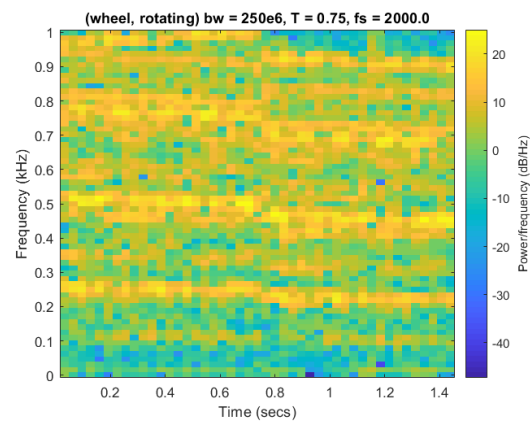
simulation, the measurement was conducted in a laboratory with many other objects such as walls and chairs. The radar’s height was around 50 cm and the distance between the radar and the bike was 3 m. This allowed the main beam to cover the whole rear wheel (but the ground reflection was inevitable). The radar sent two 0.75s chirps and between the chirps there was a pause due to serial communication delay. First, a sample was measured when the rear wheel was stationary and, then, another one is measured when it was rotating. The front wheel was kept stationary all the time. The DC component of the samples was suppressed by subtracting the mean. As shown in figure 4.6b, a curve with strong amplitude appear at low frequency when the wheel was stationary. And in figure 4.6c multiple horizontal curves span the spectrogram. Because the slope of the chirp was very small, the measurement result was similar to the one in [3], where a CW radar is used.



(a) A photo of the measurement environment.



(b) Spectrogram of the stationary bike wheel.



(c) Spectrogram of the rotating bike wheel.

Figure 4.6: Bike wheel measurement environment and spectrograms.

4.5 Walking human

4.5.1 Simulation

The global human walking model proposed by Boulic et al. [20] is the kinetic model used in the simulation. The model is based on an empirical mathematical parameterization using biomechanical experimental data instead of motion equations. The model is an average human walking model in which the information of personalized

features is averaged out from measurements of a lot of individuals [8]. A MATLAB program that simulates the micro-Doppler effect of human walking with a pulse radar is implemented in [8]. Our simulation of a walking human is based on this program, with some modifications in the radar model.

Human is a non-rigid body since the shape of the body changes during its movement. To simplify the target model, the human is modeled as jointly connected ellipsoids. Each ellipsoids independently reflect the incident wave and the radar receives the summation of the reflections. In this way, a non-rigid body problem is treated as multiple rigid body problems. There are 17 different joints that represent the base, thorax, head, left/right shoulder, left/right elbow, left/right hand, left/right hip, left/right knee, left/right ankle and left/right toes. 16 links for kinematics analysis are defined for the neck, torso, left/right shoulder, left/right upper arm, left/right lower arm, left/right hip, left/right upper leg, left/right lower leg and left/right foot. Figure 4.7 marks all the joints and links. In the high-frequency limit, closed form equations of RCS of perfect electric conducting ellipsoids can be found in [8]. The ellipsoids are connected as shown in figure 4.9. The lengths of the ellipsoids' long axis are listed in [21] and marked in figure 4.8.

The motion of the walking human can be decomposed into gait cycles. The gait cycle is defined as the period from one foot touching the ground to the same foot touching the ground again. The gait cycle can be decomposed into a sequence of small motions. Forward propulsion of the centre of gravity [22] is involved in the gait cycle. There are two phases in one gait cycle: the stance phase, when the foot remains in contact with the ground, and the swing phase: when the foot is not in contact with the ground. The model assumes that the human walks at a fixed frequency and speed.

For each gait cycle, the kinematic model of the human is described by 12 trajectories, while the so-called Denavit–Hartenberg parameters are used to describe the links' coordinates. Six of the trajectories are used to describe the six degrees of freedom of the human's spindle, referred to as a base in the model. Five of the trajectories are used to describe the joint angles of the limbs, where the left and the right trajectories

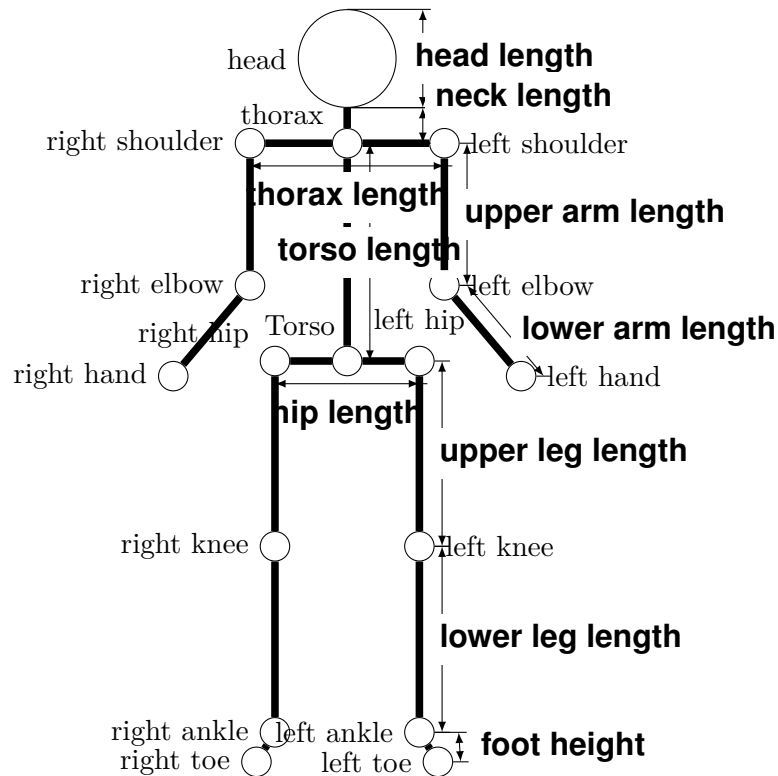


Figure 4.7: The joints of the human model.

are mirrored (given a suitable translation) with respect to the vertical plane that divides the body in two equal parts. The last trajectory is used to describe the rotation of the thorax. The 3-D trajectories of some joints are plotted in figure 4.10.

To calculate the reflected signal of the ellipsoids, the trajectory and the RCS of the ellipsoids are required. The trajectory of ellipsoids' center points can be calculated from the global human walking model. The RCS of the ellipsoids can be calculated from the relative position and orientation of the ellipsoids to the radar. Since we assume that each ellipsoid has an independent reflection, simulation of different body parts can be analyzed separately, i.e. shadowing is ignored. We assume that the human is 1.8 meters tall and walks towards the radar at a speed of 1.8 m/s. The radar is placed 10 meters away and 1.5 meters above the ground.

Figure 4.11 shows the four simulation spectrograms during three gait cycles of four selected body parts of the walking human. The setting *S1* of a high resolution radar in table corresponds to figure 4.11. The frequency in each spectrogram tends

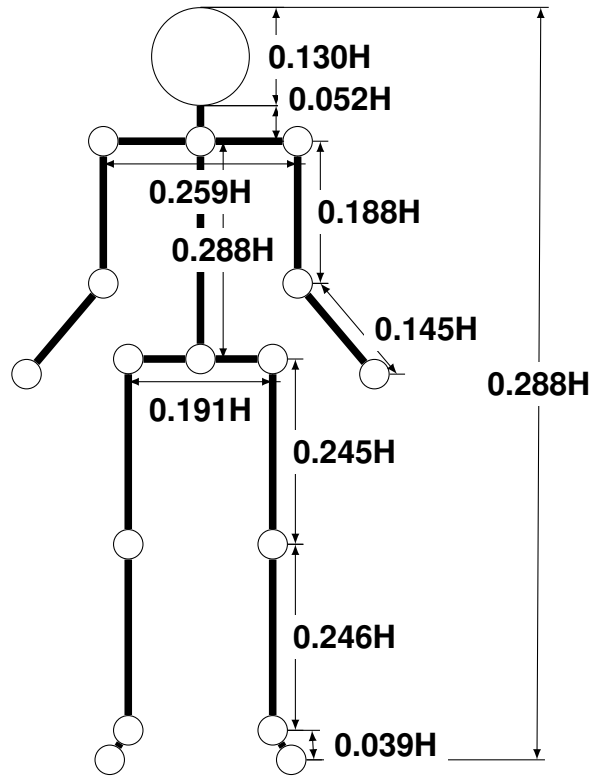


Figure 4.8: The dimension of human parts.



Figure 4.9: Human ellipsoid model.

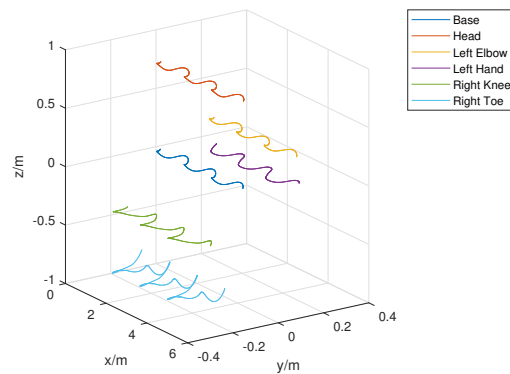


Figure 4.10: Trajectories of human joints during a gait cycle.

to decrease over time because the human is getting closer to the radar and the range beat frequency decreases. The left foot has a larger instantaneous speed and it generates a larger frequency variation as shown in the spectrogram. The torso moves steadily forward, so its frequency in the spectrogram tends to be steady.

However, the actual radar kit cannot achieve such high resolution as in the simulation

because it is unable to transmit successive chirps and the chirp duration cannot be smaller than 75 ms. In order to observe at least one full gait cycle which is roughly 1 second long, a long chirp is considered to be used in the simulation to model the actual measurement. The setting of a low resolution radar ($S2$) in table 4.1 is also considered. Figure 4.12 shows the results for a simulation with 750-millisecond-long chirps. The long chirp is modeled as 150 virtual short chirps, as described in section 2.1.3. Each short chirp is 5 ms long which is the same as in the previous simulation. Each short chirp has 1.67 MHz bandwidth so that the range resolution degrades to 90 m. The range beat frequency of the target is around 2 kHz in the previous high resolution simulation $S1$, while the range beat frequency of the same target is less than 100 Hz in current setting $S2$. Since the range resolution is very low, it is hard to extract the information about the variation in range from the spectrogram. However, the Doppler frequency is not affected by the low range resolution. In the previous simulation as seen in figure 4.13, the Doppler frequency is not distinct when the radar setting resolution is high. But in this low resolution case as seen in figure 4.12, the Doppler frequency is clear over the whole sample frequency range.

Given the superposition of the reflection from every body part, the micro-Doppler signature of the whole human body is shown in figure 4.13 and 4.14 for $S1$ and $S2$ respectively. Figure 4.13 illustrates both the range and velocity variation. However, the radar setting cannot be used in the current hardware. Figure 4.14 is the simulation using the actual radar setting with a long chirp duration. It shows detailed micro-Doppler signature, clear gait cycle of the movement and almost no signal related to range variation.

4.5.2 Experimental measurement

For the human measurement, the radar is configured according to the settings $S3$ in table 4.1. Before we plot the spectrograms, the mean of the radar signal is subtracted. Figure 4.15 shows the scenario when a person walks at a normal speed towards the radar from a distance of ten meters. Figure 4.16 shows the scenario

when a person walks towards the radar without swinging the arms. Even though none of the measurement managed to capture a full gait cycle of the person as the simulation in figure 4.14, the measurements provide different micro-Doppler signatures according to the actual movements. Also, it should be noticed that the measurement results are similar to the simulation results.

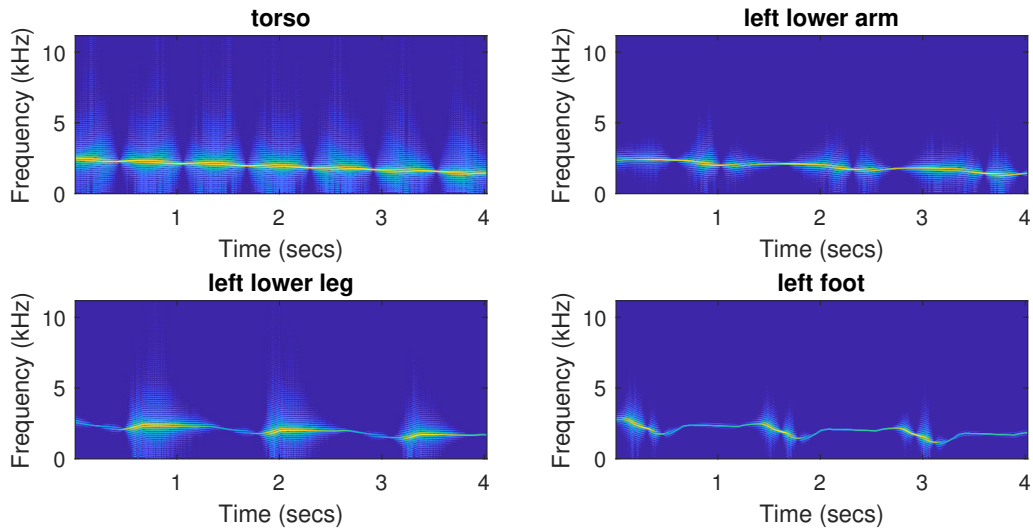


Figure 4.11: The spectrogram generated by four body parts in high resolution FMCW radar.

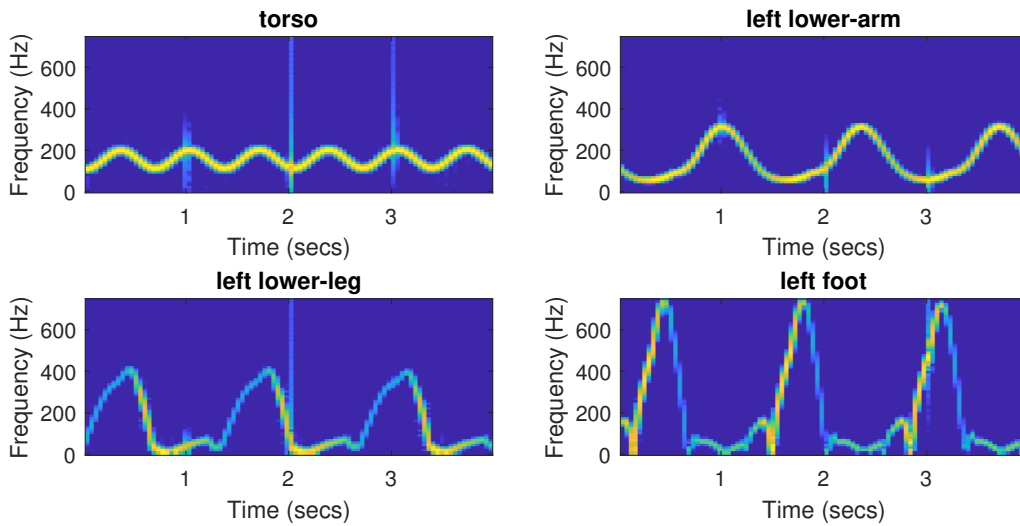


Figure 4.12: The spectrogram generated by four body parts in low resolution FMCW radar.



Figure 4.13: The spectrogram of a walking human with high resolution radar settings $S1$.



Figure 4.14: The spectrogram of a human walking with low resolution radar settings $S2$.

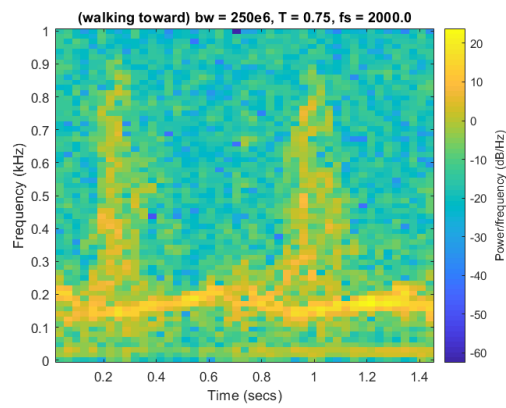


Figure 4.15: Micro-Doppler measurement for human walking towards the radar with swinging arms.

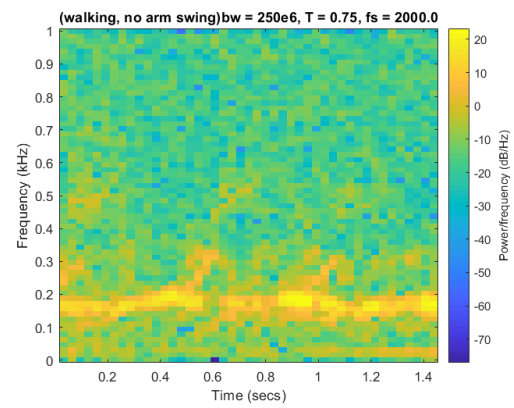


Figure 4.16: Micro-Doppler measurement for human walking towards the radar without swinging arms.

5

Classification With Neural Networks

5.1 Data measurement campaign

Four different human activities are considered in the classification problem: walking while swinging arms (0), walking without swinging arms (1), boxing while standing still (2) and standing still (3). As discussed in chapter 4, the micro-Doppler signature changes significantly with the radial velocity of the target. For a target or a body composed of many rigid parts, we assume that the motion is mainly in the direction of translation (or walking for a human). The angle between the direction of translation and the direction towards the radar is referred to as the aspect angle below. In order to explore the potential of deep learning algorithms to distinguish aspect angles, we measured these four activities at different aspect angles by changing the orientation of the person: facing the radar with the front (0), with the back (1) and with the side (2). There are 12 classes (shown in table 5.1) of data measured in total. Besides, the data is evenly collected from two people.

In the measurement campaign, the radar is configured to use settings $S3$ in table 4.1, i.e., the bandwidth is set to 250 MHz and each chirp is 750 ms. Each chirp has 1501 samples. Therefore, the sampling frequency of the radar is 2000 Hz. A MATLAB script is used to measure the data. The data is measured in a 50-chirp batch, which means the person is doing the same activity during each batch measurement.

Class name	label (0-11)	activities (0-3)	direction (0-2)	# of samples
walk_wa_side	0	0	2	350
walk_wa_front	1	0	0	350
walk_wa_back	2	0	1	350
walk_woa_front	3	1	0	350
walk_woa_side	4	1	2	350
walk_woa_back	5	1	1	350
boxing_front	6	2	0	450
boxing_back	7	2	1	450
boxing_side	8	2	2	350
standing_front	9	3	0	350
standing_back	10	3	1	350
standing_side	11	3	2	350

Table 5.1: Class description and the number of samples.

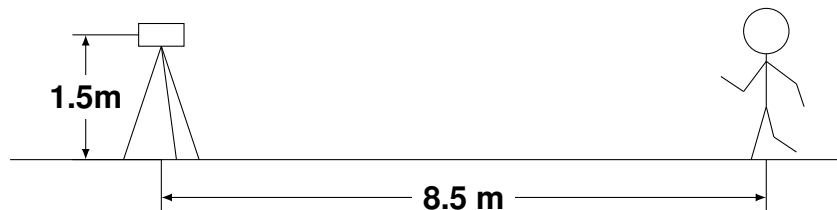


Figure 5.1: Radar setup model for data measurement campaign.

The measurement campaign is carried out in a laboratory shown in figure 5.3. This picture is taken while the human is walking without arms swinging and the radar is pointing at the back of the human, corresponding to label 5 in the table 5.1. A special 3D printed box, as shown in figure 5.2, is built to hold the radar on a 1.5 meters high tripod. The aperture of the horn antenna is oriented horizontally such that the main lobe is parallel to the floor of the laboratory. A person executes the activities around 8.5 meters away from the radar, which makes the main lobe illuminates the entire body, as shown in figure 5.1. There is no object blocking the line-of-sight path between the radar and the person. The indoor environment contains multiple large objects such as the walls surrounding the person, but they are

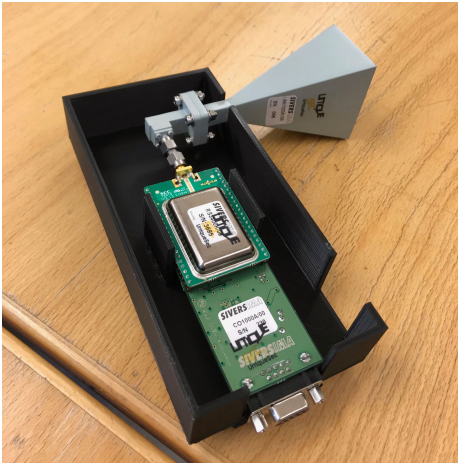


Figure 5.2: Radar kit assembly with a 3D printed box.



Figure 5.3: Data measurement campaign environment.

stationary and, thus, they do not induce any contribution to the Doppler frequency. They are not likely to produce high range beat frequency due to the long duration of the radar chirp.

5.2 Data preprocessing

Inspired by [23] and natural language processing algorithms, the radar signal in time domain is converted to joint time-frequency domain by STFT to produce a spectrogram. In addition, we use Python as programming language instead of MATLAB for the deep learning part of the project because of two reasons: first, powerful deep learning frameworks such as Tensorflow and Keras are available in Python; second, small embedded devices such as Raspberry Pi are able to run light-weight deep learning models written in Python. Therefore, the data measured in the previous section is converted from MATLAB format to Numpy archive which is a Python package for the manipulation of arrays and matrices. The Numpy archive contains four data entities for each sample: the time-domain raw data, two spectrograms and one class label.

The three-step preprocessing is:

1. The DC component is suppressed by subtracting each chirp's mean. As shown in chapter 3 where the DC component is removed, the SNR of Doppler signature is significantly higher and, thus, it is easier to distinguish.
2. Two spectrograms with different STFT window lengths are calculated for each chirp.
3. The samples are shuffled and divided into two sets. The training set contained 85% of the samples and the test set contained the rest.

The first kind of spectrogram is generated by the STFT with the window length of 64 and the sample overlap of 16, and its dimension is 33×33 . The other kind of spectrogram has the window length of 128 and the window overlaps of 32, which results in a spectrogram of dimension 17×65 . The window is a Hanning window. Figure A.1 and A.2 in the appendix A show examples from each class. The spectrogram with a window length of 128 has a high frequency resolution and a low time resolution, whereas the other one has high time resolution and low frequency resolution. We will show that using two kinds of spectrograms together can improve the classification accuracy in a CNN based neural network.

5.3 Neural networks structure and classification results

We use five neural networks to solve the classification problem, three are CNN based and two are RNN based. Among the CNN based neural networks, three kinds of networks are tested. The first two use only one kind of spectrogram (either a window length of 64 or 128). The third one uses both these spectrograms simultaneously and the accuracy is significantly higher. Among the RNN based neural networks, bidirectional GRU (Gated Recurrent Unit) [24] and bidirectional LSTM (Long Short-term Memory) [24] are utilized. Among all neural network models, we use the CNN called *Double input CNN* which uses two spectrograms and it achieves

the highest test accuracy (90.15%), followed by bidirectional RNN with GRU cell (89.55%) or LSTM (89.39%) cell.

For all neural networks, batch normalization is used to reduce covariance shift [25] and dropout layers and early stopping are used to prevent overfitting. Adadelta optimization [26] is used in a stochastic gradient descent. Generally, each model is trained for 200 to 700 epochs. The weights of the model are saved after each epoch only if the loss on the test set is reduced. The model with the highest test accuracy is selected to represent the best performance of the model.

5.3.1 CNN based models

5.3.1.1 Neural network structure

In this part, we start with a shallow CNN neural network shown in figure 5.4, which has three convolution layers and one fully connected layer. Note that the batch normalization layers and dropout layers are omitted in figure. This shallow model is able to achieve 97% test accuracy on MNIST handwritten digit database [27], and the database has similar input and output shape as our problem. Then, we extend this model to give it two inputs to utilize two kinds of spectrograms (shown in figure 5.5). All the detailed model structures are listed in appendix B.

5.3.1.2 Performance

model	win. length	train acc.	test acc.	train loss	test loss	epochs
small CNN	64	0.8604	0.8727	0.3950	0.3318	240
small CNN	128	0.8543	0.8606	0.4267	0.3884	400
Double input CNN	both	0.9115	0.9015	0.2296	0.3112	170

Table 5.2: Performance comparison between *small CNN* unit and *Double input CNN* (12 class problem).

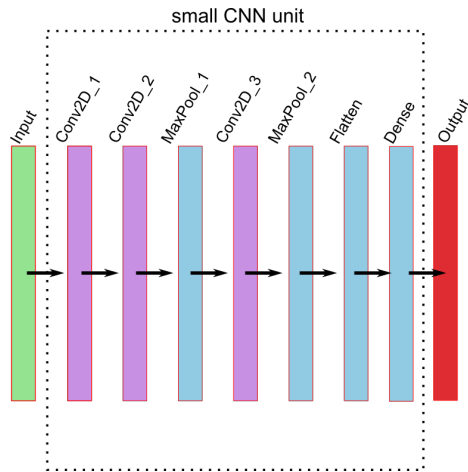


Figure 5.4: The structure of the *small CNN* unit.

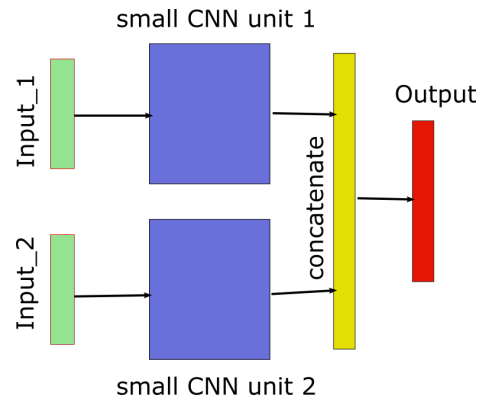


Figure 5.5: The structure of the *Double input CNN* which is built by two *small CNN* units.

Table 5.2 shows the performance of the three CNN based models. In figure 5.6, the accuracy and loss of the model are plotted. When the epoch reaches 200, the loss and test accuracy stop improving while the performance on the training set is still becoming better, which is a sign of overfitting. The model trained after 170 epochs is selected at last. The test accuracy of *Double input CNN* is 3 to 4 percent higher than the other two models that only use one type of spectrogram.

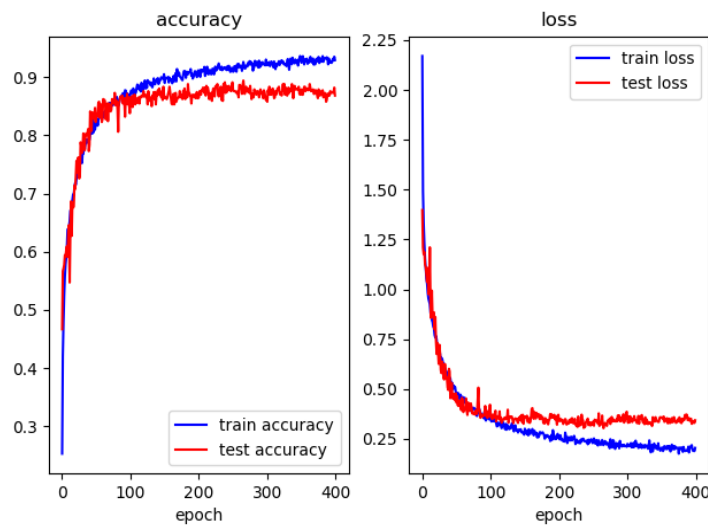


Figure 5.6: The plot of training history of *Double input CNN*.

5.3.1.3 Error analysis

The error analysis of the model is carried out by calculating the confusion matrix from the prediction of the test set. We also pick some mislabeled examples to show why the model makes mistakes. In the confusion matrix of the *Double input CNN* shown in figure 5.7, it is apparent that the accuracy on the diagonal is either way better than the average accuracy (90.15%) or far worse than the average. For example, the *Double input CNN* is very likely to mistake the activity "walk_woa_side" with "walk_wa_side" or "standing_front".

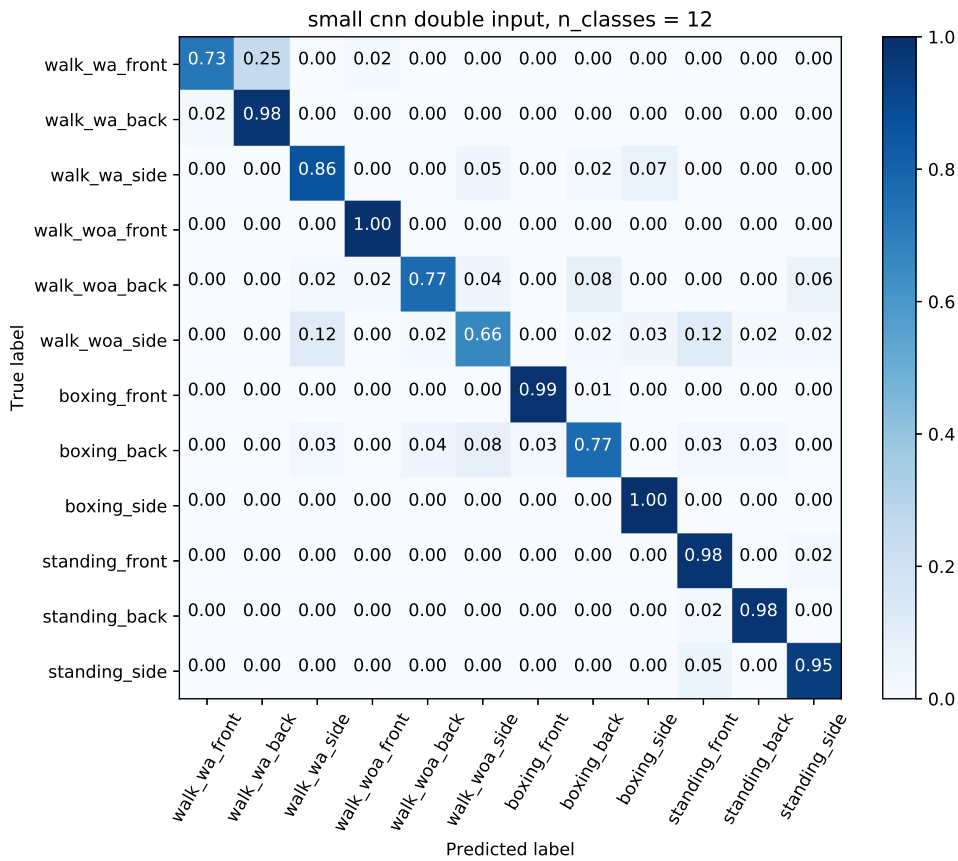


Figure 5.7: Confusion matrix of *Double input CNN* (12-class problem).

Table 5.3 lists some commonly mislabeled classes with the error ratio taken from the confusion matrix. We also summarize our findings as follows:

True label	Classified as	Error ratio (%)
walk_wa_front	walk_wa_back	25
walk_woa_side	walk_wa_side	12
walk_woa_side	standing_front	12
boxing_back	walk_woa_side	8
walk_woa_back	boxing_back	8

Table 5.3: Some common mistakes made by *Double input CNN*.

1. When the person is standing still, the model has a surprisingly high accuracy in detecting the direction despite the absence of Doppler frequency.
2. Among the activities that are *not* standing still, the model is also able to correctly identify the activities when the person faces the radar with the front, except for walk_wa_front.
3. When the person faces the radar with the side and s/he is *not* standing still, the model is prone to make mistakes except for boxing_side. The reason is that the radial velocity of the movement is tangential and very little Doppler frequency can be observed.
4. While the person faces the radar with the back and s/he is *not* standing still, the model has high accuracy only in walk_wa_back.

Next, we take a closer look at the mislabeled samples. On the left of figure 5.8, a mislabeled example belonging to "walk_wa_front" is shown, but it is classified as "walk_wa_back" (most frequent error listed in table 5.3). On the right side, there are four correctly labeled examples from the test set. Because of low frequency resolution and the very slight difference between the two activities, the micro-Doppler signatures look very similar. It may be difficult for the neural network to learn the minor difference given limited training samples.

From the confusion matrix, we can infer what kinds of physical information the model makes use of. First, the model uses the variation in radial speed when the

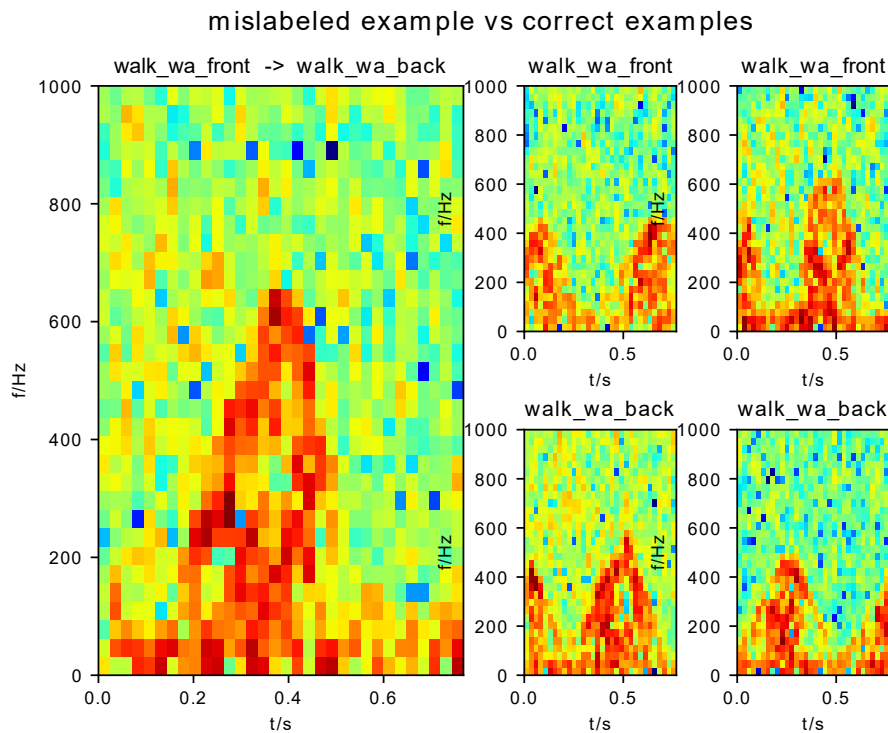


Figure 5.8: The example of a mislabeled sample.

person faces the radar with the front. Second, the model uses the change in RCS. Although "boxing_side" has little radial speed variation because the movement is perpendicular to the radial direction as viewed from the radar, the model correctly labels every example of this class in the test set. We examine some samples from the activities from "boxing_side" and they look very similar to standing still. Because the RCS is changing when boxing, the model can distinguish it from standing still.

5.3.1.4 The influence of DC component

The same structure of *Double input CNN* is trained on spectrograms of which the DC component is not subtracted. The model easily overfits the data and it achieves 90% in training accuracy but only 85% in test accuracy. Thus, suppression of the DC component improves the performance of neural network significantly.

5.3.2 RNN based neural networks

RNN based neural networks are widely applied to sequential data such as speech and other types of sound. The micro-Doppler signature is presented by a spectrogram, which is essentially sequential data that describes how the spectrum of the signal varies with time.

We test two RNN neural networks and their structures are shown in figure 5.9. The dropout layers and batch normalization layers are omitted in this figure. They have the same structure except for the RNN unit.

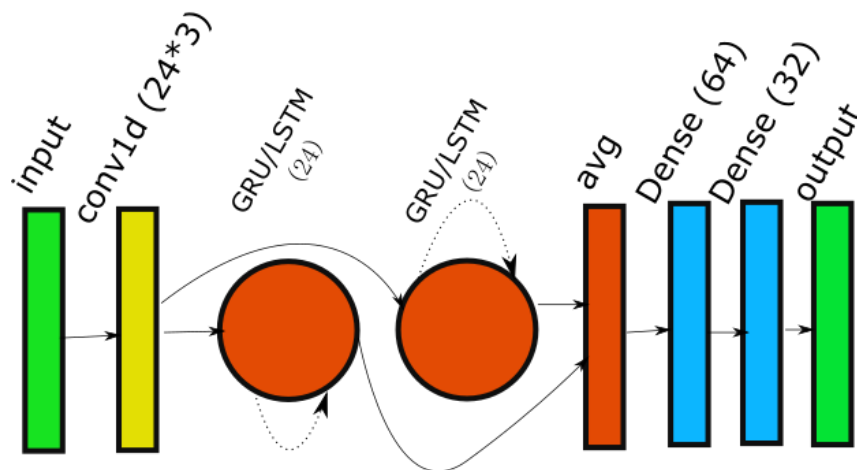


Figure 5.9: RNN based neural networks' structure. *Input*: spectrogram of window length 64 or 128; *conv1d*: 24 one dimensional convolutions with filter size 3; *RNN unit*: either GRU or LSTM unit with output size 24; *avg*: the forward output and backward output of bidirectional RNN units are averaged; *Dense*: two fully connected layers. Dashed lines represent recursive connection.

5.3.2.1 Performance

The accuracy of RNN based neural networks shown in table 5.4 is very close to *Double input CNN*. However, the loss is higher than *Double input CNN*, which means RNN based neural networks has a lower confidence level. We also implement the RNN with two inputs and tried to benefit from both the high time resolution

and the high frequency resolution as *Double input CNN*, but the accuracy is only around 80%.

RNN unit	train acc.	test acc.	train loss	test loss	epochs
GRU	0.8816	0.8955	0.4112	0.4054	633
LSTM	0.8701	0.8939	0.4342	0.4259	466

Table 5.4: RNN based neural networks performance.

5.3.2.2 Error analysis

The confusion matrices of the RNN based neural networks are similar to the one of *double input CNN*. However, the most likely error made by *double input CNN* does not happen to RNN models. The confusion matrices are shown in figure 5.10 and figure 5.11.

5.4 Predict only the activity (4-class problem)

In previous discussion we try to solve a 12-class problem, with around 300 training samples for each class. However, in some applications such as fall detection, people are only interested in predicting the type of activities (such as walking normally or falling down) rather than the aspect angle of the target. Therefore, we also modify the output layer's size of the neural network to four, and the output is then the prediction of the activity type. In this case, the number of training samples is increased to 900 for each class. We trained a *small CNN* unit and a *Double input CNN* to predict the activity type, but the accuracy is not significantly increased even if the training set size is tripled for each class. In Table 5.5, the loss and accuracy are listed.

Comparing the result of 12-class problem (table 5.2) with the result of the 4-class problem (table 5.5), we can notice an improvement in the accuracy of the *small CNN* unit. However, the overall accuracy is still around 90%.

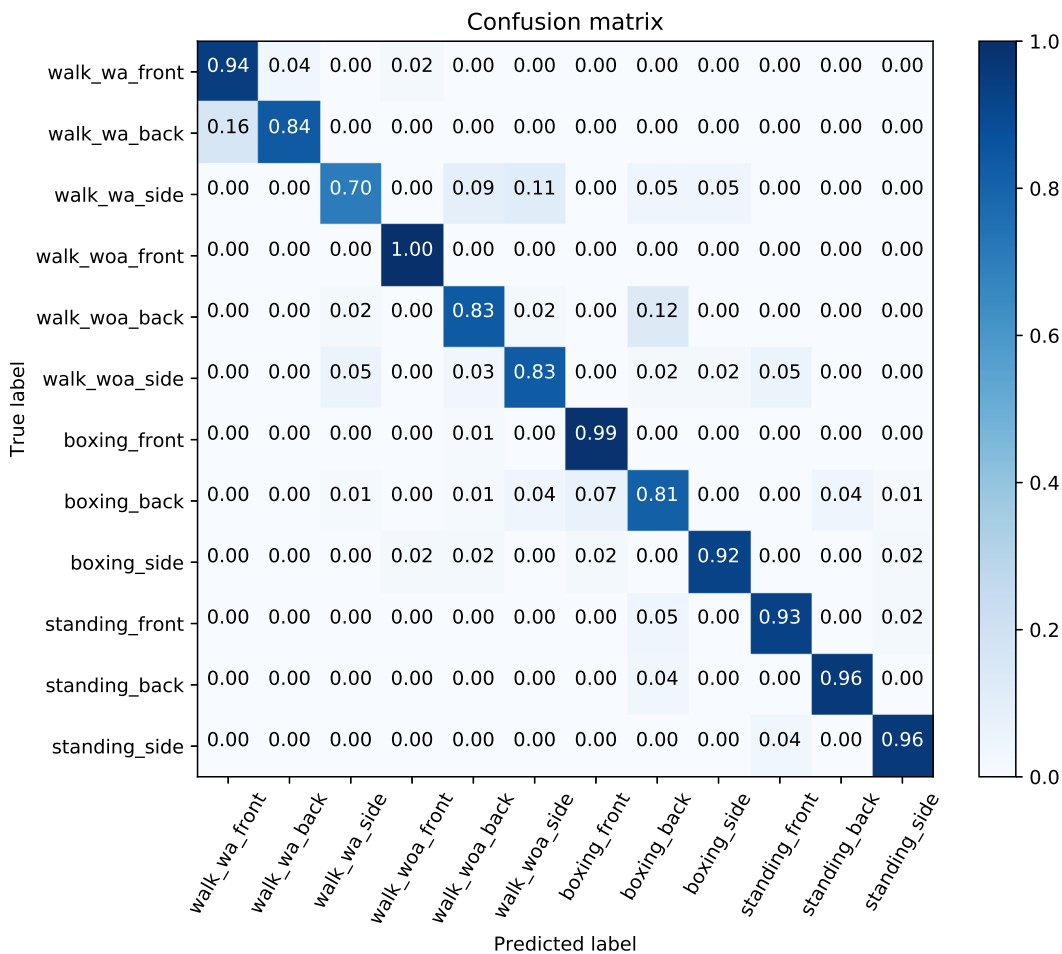


Figure 5.10: Confusion matrix of bidirectional GRU model.

The confusion matrix of *Double input CNN* on the 4-class problem is shown in figure 5.12. The accuracy on prediction the activity "standing_still" is much higher than the other three activities. So the model still has bottlenecks in certain activities.

5.5 Tuning parameters of neural networks

We mentioned in section 5.3 that the dropout layers and the early stopping are used for preventing the model from overfitting. The goal of tuning the dropout factor and epoch number is to reduce the gap between the training accuracy and test accuracy.

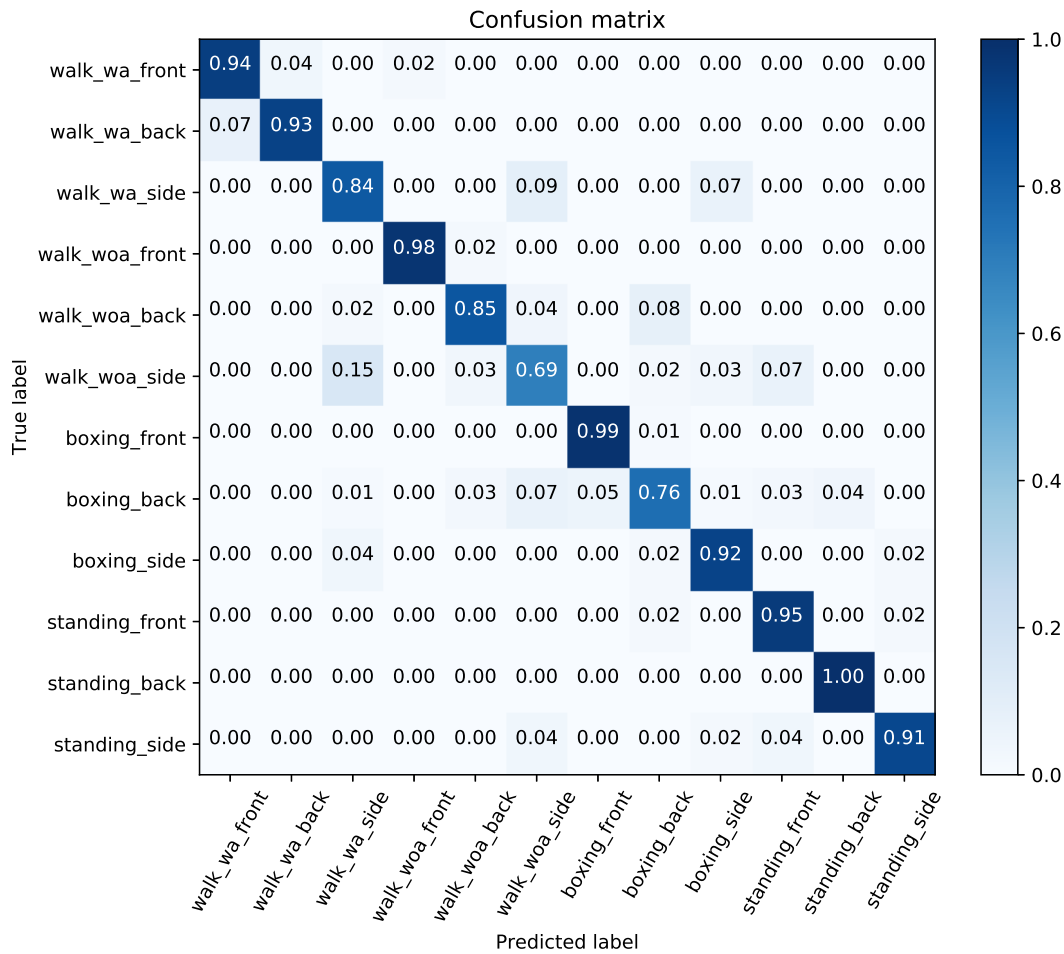


Figure 5.11: Confusion matrix of bidirectional LSTM model.

The reason is discussed below.

In a supervised machine learning problem, the *Bayes optimal accuracy* is the theoretical upper bound that any machine learning model can possibly achieve [28]. The *Bayes optimal accuracy* basically uses the conditional probability density of the target (or label) given the observation. In our thesis, the target is the type of the activity whose the observation is a spectrogram. However, in most real-world problems like ours, this conditional probability is unknown so that the accuracy upper bound is also unknown. It is possible to use human-level accuracy as an estimation of the Bayes optimal accuracy in some problem, e.x. image classification. Usually

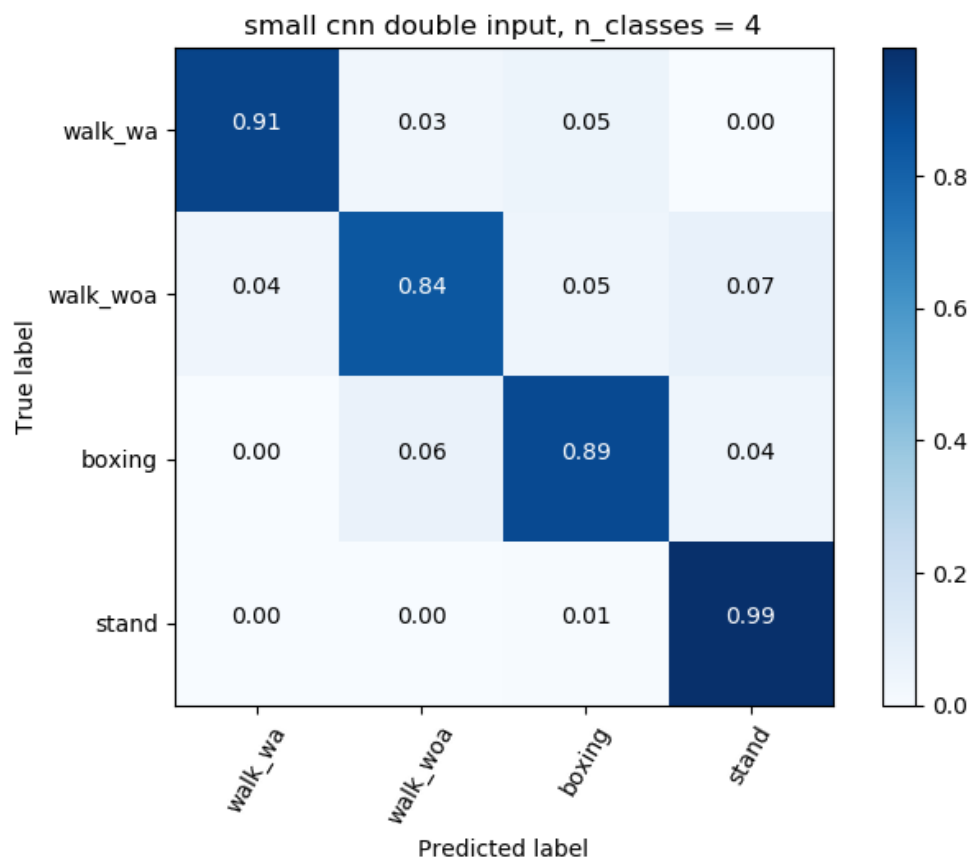


Figure 5.12: Confusion matrix of *Double input CNN* (4-class problem).

model	win. length	train acc.	test acc.	train loss	test loss	epochs
small CNN	64	0.8963	0.9045	0.2749	0.2807	260
small CNN	128	0.9160	0.8984	0.2379	0.2573	240
Double input CNN	both	0.9147	0.9090	0.2313	0.2637	100

Table 5.5: Performance comparison between *small CNN* unit and *Double input CNN* (4-class problem).

the training accuracy is higher than test accuracy. A special case occurs if dropout layers are used. In our results shown in table 5.5, the test accuracy is higher because dropout layers are active during training but they are disabled during the evaluation on the test set. Thus, should the dropout layers be disabled on the training set, training accuracy would be higher. A model underfits the data when the avoidable bias is high, while it is overfits the data when the variance is high. In the table 5.6, we list some examples of the status of a machine learning model. We also show an example of an overfitting model in figure 5.13.

human-level acc.			0.91		
model	train acc.	test acc.	bias	variance	status
1	0.80	0.80	high	low	underfitting
2	0.99	0.80	low	high	overfitting
3	0.90	0.89	low	low	acceptable

Table 5.6: Examples of the status of a machine learning model.

If human-level accuracy is available, the goal would be to make the model’s training accuracy approach or even surpass the human-level accuracy while keeping the variance small. However, the human-level accuracy is also unknown in this work. As a result, we first try to train some models with a rather large capacity (many layers and parameters) and get an overfitting model. Then, we estimate the highest test accuracy as a proxy to Bayes optimal accuracy. We find that the training accuracy can easily approach 100% because it is an overfitting model, but the test accuracy is always around 90% or lower. After that, we reduce the model capacity by removing

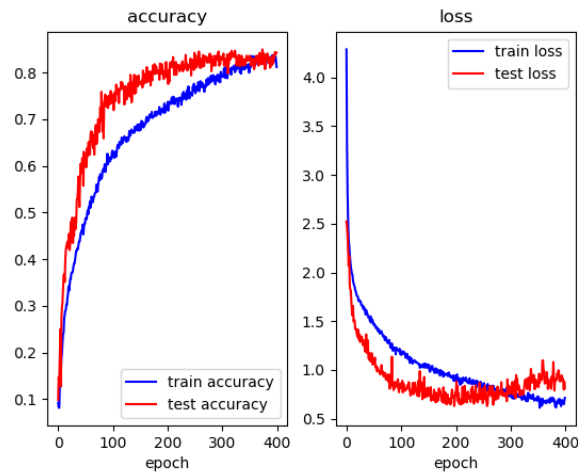


Figure 5.13: Training history of an overfitting model based on *Double input CNN* when the data is divided into four different classes.

layers or reducing the size of layers until the variance is reduced to an acceptable level. Next, we test different dropout factors until we find a model with a training accuracy and a test accuracy that are both close to 90% regardless of how long it is trained. Finally, the training stops at the epoch when the test accuracy is at its highest value. We mitigate problems associated with overfitting in this way.

6

Discussion

6.1 Limitations

Although the trained model achieves the test accuracy of around 90%, many problems remain unsolved because of some limitations. In this section, we discuss these problems and their cause.

6.1.1 Overfitting

It is possible that the overfitting occurs because the dataset is divided into only two sets, which are a training set and a test set. A better way to divide the dataset is to create yet another set, which is referred to as the development set. With three sets, we can train the neural network on the training set while observing the accuracy on the development set. The test set should not be seen by the neural network until it does well on the development set. This might avoid overfitting. However, the dataset only contains 4400 samples in total. Thus, we made a compromise such that the size of the training set is kept relatively large by not using any samples to create another set.

6.1.2 Not enough data

With more data, the neural network models may bring us closer to a better classification result. The performance of neural networks relies heavily on the number of training data. 4400 is a rather small data set for training neural networks, which forces us to use shallow neural networks to avoid overfitting. Thus, it is worth trying to acquire more data in order to get higher accuracy.

6.1.3 Changes of the environment

All the samples in the dataset are collected in the same environment. Several large metal objects are close to the measurement subject and it is possible for the radar to measure the second and third reflection from them. The propagation channel is rather complex and this may also affect the result. This leads us to the question: "if the environment changed, will our pre-trained models still maintain its current accuracy?" The answer is unknown since we lack samples collected in other environments.

6.2 Future work

We discovered in chapter 5 that the neural network models had difficulty in differentiating some activities and those activities generally had either little variation in radial speed or very similar spectrograms as compared to others. The future work should focus on improving the classification accuracy. We propose several methods that are worth trying: (i) changing radar setup, (ii) using practical activities, (iii) exploring other algorithms and (iv) using simulation data.

6.2.1 Changing radar setup

As discussed in chapter 3, the radar kit is set to have a very low range resolution because of its limitation. However, it is possible to implement our own control board to transmit successive chirps and replace the original one. In this way, the range resolution can be significantly increased and the accuracy may benefit from the high range resolution because each sample contains more information.

Some classes have similar distributions because they have little radial speed. To reduce the error rate on prediction for these classes, we propose to use two radars which are perpendicular to each other. Thus, there are two aspect angles for the target and its radial speed is measured significantly better as compared to a single radar. The data from the two radars can be combined either by averaging the measurements or use two spectrograms of the measurement as two channels. Another option is to use a radar with a beam that can be scanned mechanically or electrically. Given a multipath propagation environment, such a radar may illuminate the target from different directions by means of reflections from walls and other stationary objects.

6.2.2 Using practical activities

The 12 activities are designed for experimental purpose. Thus, they have little practical meaning. It is better to set the goal to solve some real-world problems. These problems need to involve activities that have very distinct distributions. Fall detection can be a good example. When a person falls down, the change in vertical speed is rather rapid and the change can be easily distinguished from normal activities. However, it is difficult to measure a balanced dataset with classes that have a similar number of samples. It is also important to change the environment in the data measurement campaign to get a more generalized model.

6.2.3 Exploring other algorithms

Neural networks are chosen because of their flexibility. The degree of freedom to control a neural network is higher than traditional machine learning algorithms. For example, one can control the capacity of a neural network by adding new layers and changing the sizes of the layers, whereas only a few hyperparameters can be changed in SVM. Because the focus of the project is on neural networks, other algorithms such as SVM are not used in the classification part. However, they can potentially have better performance than neural networks for some applications.

6.2.4 Using simulation data to train

Radar measurements of the type that is presented in this thesis are very time-consuming. In addition, disturbances come from the radar hardware and the environment. More importantly, the neural networks work as a black box. Thus, we cannot tell to what degree the disturbances affect the result. As an alternative, we propose to use simulation to generate the dataset. To generate the radar data for training, the motion data of the human is a prerequisite and it can be either found in a motion capture database [29] or measured by motion capture equipment such as Microsoft Kinect. After the motion data is available, we can apply a radar model to it and even add disturbances (noise, multipath effects and more) to the simulated radar data. A large number of samples can be generated in a well controlled manner in the simulation environment.

7

Conclusion

In conclusion, the structural movement and the radar model together define the micro-Doppler signature. Neural networks are capable to classify micro-Doppler signature of different human activities in a monotonous environment while demanding minimal preprocessing and little domain knowledge even if the training set is small. The neural networks can achieve a rather high accuracy when the target's movement produces distinct variation in the physical metrics such as the radial speed and the RCS. However, the neural networks are prone to mislabel activities that have very similar statistical distributions.

Bibliography

- [1] V. C. Chen, “Advances in applications of radar micro-Doppler signatures,” in *2014 IEEE Conference on Antenna Measurements Applications (CAMA)*, pp. 1–4, Nov 2014.
- [2] V. C. Chen, F. Li, S. S. Ho, and H. Wechsler, “Micro-Doppler effect in radar: phenomenon, model, and simulation study,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, pp. 2–21, Jan 2006.
- [3] D. Belgiovane and C. C. Chen, “Micro-Doppler characteristics of pedestrians and bicycles for automotive radar sensors at 77 GHz,” in *2017 11th European Conference on Antennas and Propagation (EUCAP)*, pp. 2912–2916, March 2017.
- [4] B. R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*. CRC Press, 3 ed., 2013.
- [5] A. G. Stove, “Linear FMCW radar techniques,” *IEEE Proceedings - Radar and Signal Processing*, vol. 139, pp. 343–350, Oct 1992.
- [6] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, “Short-range FMCW monopulse radar for hand-gesture sensing,” in *2015 IEEE Radar Conference (RadarCon)*, pp. 1491–1496, May 2015.
- [7] G. M. Brooker, “Understanding millimetre wave FMCW radars,” in *1st international Conference on Sensing Technology*, pp. 152–157, 2005.
- [8] C. C. Victor, *The micro-Doppler effect in radar*. Artech house, 2011.

- [9] Y. Kim and H. Ling, “Human activity classification based on micro-Doppler signatures using a support vector machine,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 1328–1337, May 2009.
- [10] Y. Kim and B. Toomajian, “Hand gesture recognition using micro-Doppler signatures with convolutional neural network,” *IEEE Access*, vol. 4, pp. 7125–7130, 2016.
- [11] Y. Kim and T. Moon, “Human detection and activity classification based on micro-Doppler signatures using deep convolutional neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, pp. 8–12, Jan 2016.
- [12] V. C. Chen, “Doppler signatures of radar backscattering from objects with micro-motions,” *IET Signal Processing*, vol. 2, pp. 291–300, September 2008.
- [13] J. Li, S. L. Phung, F. H. C. Tivive, and A. Bouzerdoum, “Automatic classification of human motions using Doppler radar,” in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, June 2012.
- [14] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.
- [15] J. Lei and C. Lu, “Target classification based on micro-Doppler signatures,” in *IEEE International Radar Conference, 2005.*, pp. 179–183, May 2005.
- [16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [17] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *ArXiv e-prints*, Dec. 2014.
- [18] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *ArXiv e-prints*, July 2012.
- [19] B. Shaffer, “Why are automotive radar systems

- moving from 24GHz to 77GHz?." https://e2e.ti.com/blogs_/b/behind_the_wheel/archive/2017/10/25/why-are-automotive-radar-systems-moving-from-24GHz-to-77GHz, March 2018.
- [20] R. Boulic, N. M. Thalmann, and D. Thalmann, "A global human walking model with real-time kinematic personification," *The Visual Computer*, vol. 6, pp. 344–358, Nov 1990.
- [21] D. A. Winter, *Biomechanics and motor control of human movement*. Hoboken, N.J: John Wiley and Sons, 4. ed., 2009.
- [22] "Bipedal gait cycle." https://en.wikipedia.org/wiki/Bipedal_gait_cycle, last access March 2018.
- [23] J. Kwon and N. Kwak, "Human detection by neural networks using a low-cost short-range doppler radar sensor," in *2017 IEEE Radar Conference (Radar-Conf)*, pp. 0755–0760, May 2017.
- [24] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.
- [25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.
- [26] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.
- [27] Y. LeCun and C. Cortes, "MNIST handwritten digit database." <http://yann.lecun.com/exdb/mnist/>, 2010.
- [28] A. Ng, "Machine learning yearning." <http://www.mlyearning.org/>, last access Jan. 2019.
- [29] M. R. Laboratory, "CMU graphics lab motion capture database." <http://>

`mocap.cs.cmu.edu`, last access Jan. 2018.

A

Appendix 1 Spectrogram examples

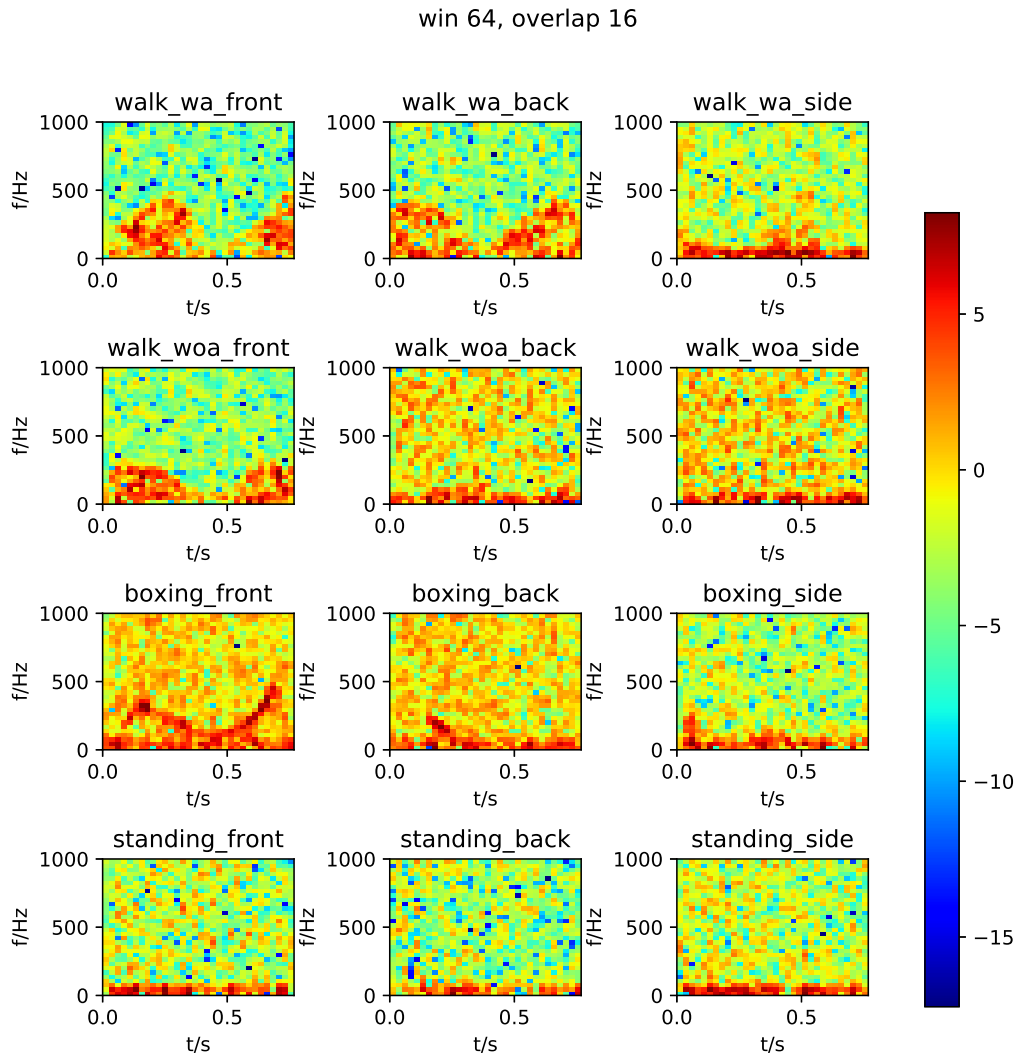


Figure A.1: Spectrogram samples of the data set, calculated by STFT with window length of 64 and an overlap of 16. Each row represents one posture. The order of the postures is: walking with swinging arms (walk_wa), walking without swinging arms (walk_woa), boxing while standing still (boxing) and standing still (standing).

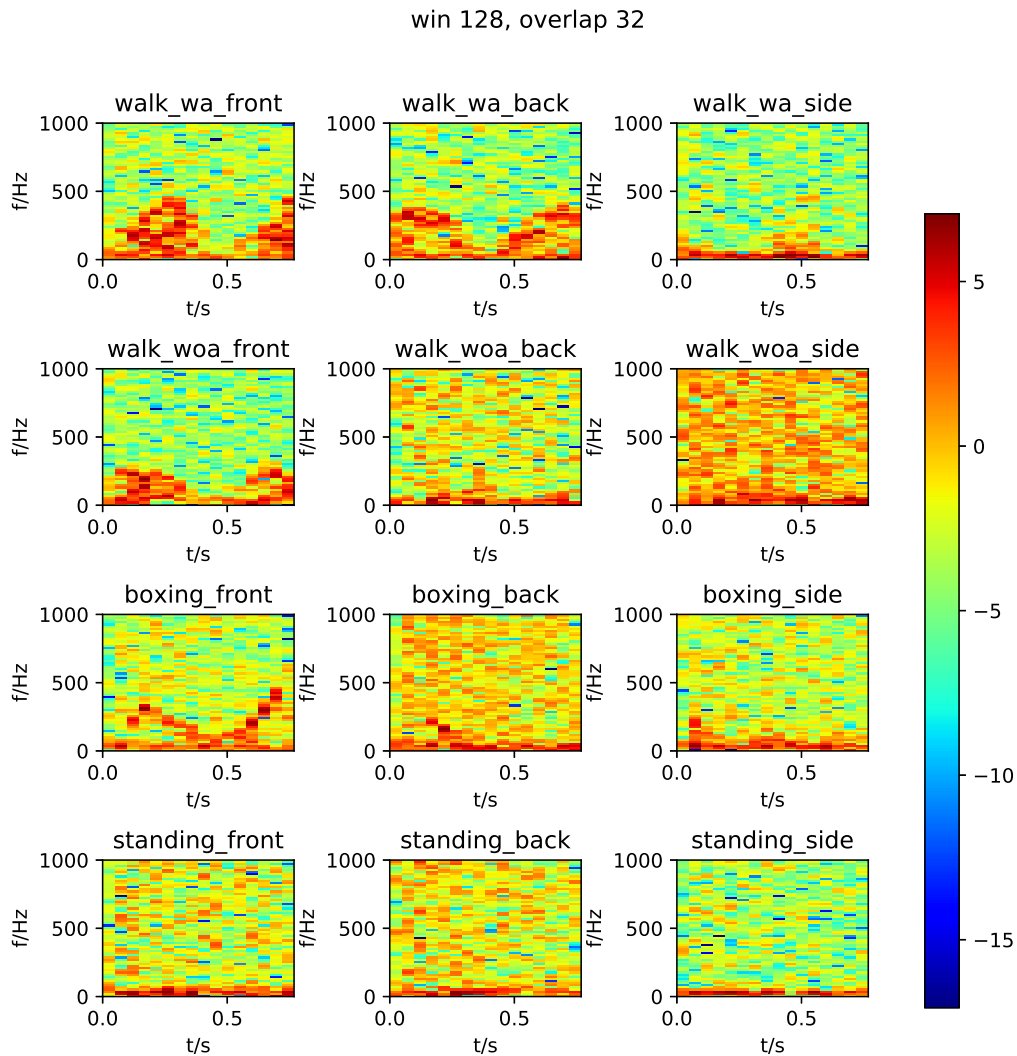


Figure A.2: The spectrograms of the same samples as in figure A.1, but calculated by STFT window length of 128 and an overlap of 32. Each row represents one posture. The order of the postures is: walking with swinging arms (walk_wa), walking without swinging arms (walk_woa), boxing while standing still (boxing) and standing still (standing).

B

Appendix 2 Neural Network model summary

B.1 Software used in this thesis

The software used in this project for the radar measurement and the neural network training is listed in table B.1.

Software	Version	For download
Python	3.7	https://www.python.org/downloads/
Tensorflow	1.4	https://www.tensorflow.org/versions/r1.4/
Keras	2.1.0	https://github.com/keras-team/keras/releases/tag/2.1.0
Numpy	1.4.0	https://github.com/numpy/numpy/releases/tag/v1.14.0
MATLAB	R2017a	https://se.mathworks.com/downloads/

Table B.1: Software used in this thesis

B.2 Model summary of Double input CNN

Double input CNN model summary

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 33, 33, 1)	0	
input_2 (InputLayer)	(None, 17, 65, 1)	0	

B. Appendix 2 Neural Network model summary

conv2d_1 (Conv2D)	(None, 31, 31, 32)	320	input_1[0][0]
conv2d_4 (Conv2D)	(None, 15, 63, 32)	320	input_2[0][0]
batch_normalization_1 (BatchNor	(None, 31, 31, 32)	128	conv2d_1[0][0]
batch_normalization_3 (BatchNor	(None, 15, 63, 32)	128	conv2d_4[0][0]
dropout_1 (Dropout)	(None, 31, 31, 32)	0	batch_normalization_1[0][0]
dropout_4 (Dropout)	(None, 15, 63, 32)	0	batch_normalization_3[0][0]
conv2d_2 (Conv2D)	(None, 29, 29, 64)	18496	dropout_1[0][0]
conv2d_5 (Conv2D)	(None, 13, 61, 64)	18496	dropout_4[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0	conv2d_2[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 6, 30, 64)	0	conv2d_5[0][0]
batch_normalization_2 (BatchNor	(None, 14, 14, 64)	256	max_pooling2d_1[0][0]
batch_normalization_4 (BatchNor	(None, 6, 30, 64)	256	max_pooling2d_3[0][0]
dropout_2 (Dropout)	(None, 14, 14, 64)	0	batch_normalization_2[0][0]
dropout_5 (Dropout)	(None, 6, 30, 64)	0	batch_normalization_4[0][0]
conv2d_3 (Conv2D)	(None, 14, 14, 24)	1560	dropout_2[0][0]
conv2d_6 (Conv2D)	(None, 6, 30, 24)	1560	dropout_5[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 7, 7, 24)	0	conv2d_3[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 3, 15, 24)	0	conv2d_6[0][0]
flatten_1 (Flatten)	(None, 1176)	0	max_pooling2d_2[0][0]
flatten_2 (Flatten)	(None, 1080)	0	max_pooling2d_4[0][0]
dropout_3 (Dropout)	(None, 1176)	0	flatten_1[0][0]
dropout_6 (Dropout)	(None, 1080)	0	flatten_2[0][0]
dense_1 (Dense)	(None, 64)	75328	dropout_3[0][0]
dense_2 (Dense)	(None, 64)	69184	dropout_6[0][0]
concatenate_1 (Concatenate)	(None, 128)	0	dense_1[0][0] dense_2[0][0]
batch_normalization_5 (BatchNor	(None, 128)	512	concatenate_1[0][0]
dropout_7 (Dropout)	(None, 128)	0	batch_normalization_5[0][0]
dense_3 (Dense)	(None, 12)	1548	dropout_7[0][0]
Total params: 188,092			
Trainable params: 187,452			
Non-trainable params: 640			

B.3 Model summary of RNN based neural networks

RNN models summary

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 17, 65)	0
conv1d_1 (Conv1D)	(None, 15, 24)	4704
batch_normalization_1	(None, 15, 24)	96
dropout_1 (Dropout)	(None, 15, 24)	0
bidirectional_1	(None, 24)	7056
dropout_2 (Dropout)	(None, 24)	0
batch_normalization_2	(None, 24)	96
dense_1 (Dense)	(None, 64)	1600
batch_normalization_3	(None, 64)	256
dropout_3 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dense_3 (Dense)	(None, 12)	396

Total params: 16,284

Trainable params: 16,060

Non-trainable params: 224