Complex-Valued Neural Networks for Radar-based Human-Motion Classification

Master Thesis Report

Ximei Yang

Complex-Valued Neural Networks for Radar-based Human-Motion Classification

Thesis

to obtain the degree of Master of Science
in Electrical Engineering
at Delft University of Technology,
to be defended publicly on Monday October 4, 2021 at 1:00 PM.

by

Ximei Yang

Faculty of Electrical Engineering, Mathematics Computer Science,
Delft University of Technology, Delft, Netherlands,
born in Hubei, China.

This thesis has been approved by the

Responsible supervisor: Prof. DSc. Alexander Yarovoy
Daily supervisor: Dr. Francesco Fioranelli

Thesis committee:

Professor DSc. Alexander Yarovoy,	TU Delft, Chair MS3
Associate Professor M.A. Zuñiga Zamalloa,	TU Delft, Embedded and Networked Systems
Assistant Professor Dr. Francesco Fioranelli	TU Delft, MS3

Keywords: Complex-valued neural network (CVNN), deep learning, human-motion classification, pulse-Doppler radar

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Nowadays, radar has been applied to human activity classification in the aging-in-place for health monitoring. The complex-valued neural networks (CVNNs) have been only minimally explored, especially on complex-valued radar signals, and there is an outstanding question on whether CVNNs can contribute to improving classification performance. This thesis proposes three complex-valued convolutional neural networks (CNNs) for human-motion classification based on monostatic radar. The range-time, range-Doppler, range-spectrum-time, and time-frequency spectrograms of micro-Doppler signatures are adopted as the input to CVNNs with different plural-handled approaches. A series of experiments determine the optimal approach and data format that achieves the highest classification accuracy. Experimental results on measured data show that 1) the accuracy of classification using CVNNs on range-Doppler and range-spectrum-time radar formats is significantly higher than the real-valued counterpart, and that 2)Deep neural networks achieve the best classification accuracy on CVNNs while shallow neural networks do not.

Acknowledgements

Time flies like an arrow, and two years are transitory. It is time to end my journey as a 'student' and again say goodbye to my university life. I appreciate having this priceless opportunity. I want to thank so many people that I don't know how to write for a moment. The two years in the Netherlands are fleeting. All this will be an unforgettable experience in my life.

First of all, I would like to acknowledge my daily supervisor, Dr. Francesco Fioranelli, for his careful guidance. He is very professional and patient. He has a serious and responsible attitude and can reply and solve problems I meet quickly. He is guiding me step by step until the end. Whenever I am confused or feel worried, he can point out the direction and encourage me. I also want to thank my responsible supervisor, Prof. Alexander Yarovoy, who always thinks of the students. Also, thank M.A. Zuñiga Zamalloa for taking time out of his busy schedule to join my thesis committee. It is my great honor to have you, three professors, as members of my thesis committee.

Everyone is kind, friendly, and enthusiastic in the MS3 group, from professors, the secretary, doctoral candidates to students. Thank Ronny Guendel, who handled the radar experiment and provided the data for my thesis, and all volunteers who joined in the experiment. Similarly, I would like to thank Simin, Yichuang, Yubin, and Yuanyuan of the same research group. They always spare no effort to help me and give me valuable opinions.

It is worth mentioning that I would like to acknowledge Dr. Zhang's and Dr. Cao's medical teams of Shanghai Huashan Hospital in particular. Six years ago, God seemed to play a huge joke on me and let me suffer from the disease and quit school. The doctors brought me back to life at this critical point and cured my disease with exquisite medical techniques. They also helped me regain confidence with warm comfort and an optimistic attitude when I experienced great pain and felt hopeless in my life. What was more important, they firmly believed that I could continue studying and striving for my idea. With their encouragement, I returned to the campus. The road to recovery is slow, long, and full of difficulties, but I am no longer afraid. The haze will eventually dissipate over time. I think I am lucky enough because I met such excellent and responsible doctors. Without them, I would never have today achievements.

When I first came to Holland, I was anxious and worried. Thank all the students and teachers I met in TUDelft. They are warm and friendly. With their help, I quickly adapted to my study and life in Delft, and I gradually have had confidence. I thank Bichi, Yujin, Xiaoyao and Yuanyuan. My life is full of joy with them, and I am no longer lonely in a foreign country. The Netherlands is an open and tolerant country with love and peace, and that is why I chose to come here. I am happy here, and I will cherish this journey.

Finally, thank my parents, who have always supported me, both financially and emotionally. Looking back on the road, I was also amazed at the master's period, which I completely dared not think of six years ago. Thank my doctors, teachers, classmates,

friends, and family again. I can't achieve these without them. John B. Goodenough, a Nobel prize laureate, said, "Some of us are like turtles. We walk slowly and struggle all the way. We can't find a way out at the age of thirty. But the turtle knows that he must go on." I am also a turtle. I thank myself for my persistence and must go on.

Ximei Yang
Delft, The Netherlands
September 28 2021

Contents

List of Figures

List of Tables

Nomenclature

AE	Autoencoder
ANN	Artificial Neural Network
BN	Batch Normalization
CAE	Convolutional Autoencoder
CM	Confusion Matrix
$CNN/ConvNet$	Convolutional Neural Network
$CONV$	Convolutional
CVD	Cadence Velocity Diagram
$CVNN$	Complex-Valued Neural Network
CW	Continuous Wave
CWT	Continuous Wavelet Transform
DCN	Deep Complex Network
DFT	Discrete Fourier Transform
DL	Deep Learning
DNN	Deep Neural Network
EM	Electromagnetic
FFT	Fast Fourier Transform
$FMCW$	Frequency-Modulated Continuous Wave
FT	Fourier Transform
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
HAR	Human Activity Recognition
HOG	Histogram of Orientated Diagram
KLD	Kullback-Leibler Divergence

KNN K-Nearest Neighbours

LDA Linear Discriminant Analysis

LPC Linear Predictive Coding

$LSTM$ Long-short Time Memory

ML Machine Learning

$MOCAP$ Motion Capture

MSE Mean Square Error

MTI Moving Target Indication

PCA Principal Components Analysis

PRF Pulse Repetition Frequency

PRI Pulse Repetition Interval

$Radar$ Radio Detection and Ranging

RCS Radar Cross Section

$ReLU$ Rectified Linear Unit

REM Rapid Eye Movement

RFC Random Forest Classifier

RNN Recurrent Neural Network

SNR Signal-to-Noise Ratio

$STFT$ Short-Time Fourier Transform

SVD Singular Value Decomposition

SVM Support Vector Machine

TF Time-Frequency

UAV Unmanned Aerial Vehicle

UWB Ultra-Wideband

wFM Weighted Fréchet Mean

WSS Wide Sense Stationary

WVD Wigner-Ville Distributions

1

INTRODUCTION

Radio detection and ranging (Radar) is widely used in the military and civil. Earlier developments in radar technology were limited to military applications such as aircraft/ship surveillance, navigation, and weapons guidance [8]. Considering radar's ability to detect objects and obstacles, positioning, and velocity measurement, radar is now used in many other civil applications, including automotive, human-computer interfaces, and health monitoring. Radar has its advantages. For example, radar-based indoor monitoring provides a non-obstructive passive motion sensing technology. In contrast, cameras may cause privacy concerns, while wearable devices need battery operation. In this chapter, the main motivation and contributions of this thesis using radar data for human activities monitoring are summarized.

1.1. MOTIVATION

Fundamentals in signal phenomenology have driven the development of unique approaches to machine learning (ML) architecture design [7]. ML algorithms have been applied in computer vision or speech domain and to radar, such as classifying human activities based on micro-Doppler, which enables assisted living in aging-in-place. For example, radar-based applications in the context of assisted living include fall detection and monitoring of activities patterns, gait analysis, and monitoring of vital signs such as respiration and heartbeat [9][10][11]. The attractiveness of radar comes from its contactless sensing nature, whereby no wearable sensors need to be attached to or worn by the users. Compared to optical cameras, no plain images of environments and people are recorded.

The general procedures of radar-based classification are as follows: radar data acquisition, signal pre-processing, feature extraction, and supervised classification. It is necessary to find an appropriate technique to extract and make full use of the characteristics of radar data to perform various tasks. When processing the radar data, the complex-valued nature of radar data is often ignored, which means no information about the phase of the radar data. However, the phase in the electromagnetic (EM) world often has many messages. Complex numbers could have a richer representational capacity, so an emerging learning network, COMPLEX-VALUED NEURAL NETWORK (CVNN), is introduced for the classification algorithm [5]. At present, the progress of deep learning in radar-based human monitoring is mainly based on real-valued operations and representations. Based on traditional architecture, the advantages offered by complex representations will be exploited.

1.2. RESEARCH GOALS

The research goals of this thesis are to make use of complex-valued radar data to classify human activities and build the architecture to fit the data, including measured data pre-processing and classification methods. In addition, the effect of this new pipeline on the accuracy of human motion classification is also studied. Based on this objective, the question is whether a whole system can effectively classify human activities from raw radar data and CVNNs to achieve better performance than the current real-valued models.

In order to answer this question, a nine-class motion experimental model is developed, and the algorithms of data pre-processing and CVNN are designed. Input data used is the micro-Doppler signature and other formats of the same data as input to ML algorithm, for example, range-Doppler and range-time. In the classic approach to radar-based human activities monitoring, the input to a classical convolutional neural network (CNN or ConvNet) is the absolute value. Not much phase information has been used in ML communities or libraries because of no phase of optical images in computer vision. Radar data is inherently complex-valued, but very few algorithms exploit it due to the absence of the complex building blocks in mainstream ML frameworks. For example, Caffe, Keras, Scikit-learn, and Theano documentation do not mention complex numbers, while TensorFlow or PyTorch have no complex-valued blocks. Hence, within the work of this thesis, different strategies to implement complex-valued neural networks are developed, with a focus on the specific application of human activities monitoring.

It is assumed that the existing data acquisition strategy remains unchanged from the MS3 group, TUDelft. The focus is radar data processing, including data pre-processing and machine learning classification. The effort is not just using the ML architectures from image processing but also try to modify and define new architectures that account for the nature of EM data. Then we train and test to evaluate complex-valued neural networks and compare them with real-valued networks on radar-based tasks to analyze their performance and determine which is more suitable.

1.3. CONTRIBUTIONS

The contributions of this thesis are as follows (in chronological order):

- Implementation of a MATLAB package for the signal pre-processing pipeline to generate radar dataset.
- Formulation of a data model tailored toward pulse-Doppler data, integrating the processed results of all data into a npy file. The model describes the output signal of PulsON P410 Ultra-Wideband (UWB) radar, including the Doppler characteristics of the range-time-channel matrix. The goal of the data model is to serve as a

starting point for developing more effective data simplification.
- Implementation of a CVNNs package for radar data classification in PyTorch framework. The input of the neural network can be complex-valued.
- Construction of small neural networks designed specifically for radar data. Rather than predefined networks in PyTorch framework, all the pieces of codes of networks are rewritten.
- Experimentation validating radar data format choices specific to the constructed CNN models. The performance of models under various conditions is analyzed. The results show that CVNNs can decrease the error rate by 5% in range-Doppler representation, and the accuracy is up to 92.6% in the whole radar data representations.
- Contribution of results arising from this thesis to the writing up of a conference paper aiming at the IEEE Radar Conference 2022.

1.4. OUTLINE

This thesis is structured as follows. Chapter 2 discusses the related literature concerning radar-based motion classification utilizing traditional and new models. Then, in Chapters 3 and 4, the related theoretical background is provided, followed by a description of detailed methodology and extensions, including data processing pipeline and signal processing techniques characterizing data. Next, Chapter 5 presents an overview of the dataset related to implementing the pre-processing approaches. Chapter 6 provides corresponding classification strategies, presenting experiment outcomes and the comparative evaluation of different models. Finally, Chapter 7 concludes this thesis and also provides directional suggestions for future research.

2

Literature Study and Research Questions

As radar data increases, research over radar signal processing combined with machine learning is becoming more critical. Radar is more widely used in sensing in the assisted living domain because it is not affected by light and has the advantage of no contact with the end-users, increasing the performance of recognizing human activities and monitoring vital signs. The main topics discussed in the literature about the radar classification system are the traditional radar signal processing and classification approaches, which are reviewed in this chapter and used to formulate the research questions of this thesis. From the current literature, it is clear that complex-valued networks have been only minimally explored, and there is an outstanding question on whether they can contribute to improving classification performances.

2.1. Radar for Sensing in Assisted Living

Radar has defense and military applications, e.g., air-traffic control to scan the surrounding space and discover airplanes. There are many fields beyond defense and military in radar application, including healthcare applications of radar for human activities monitoring, gait analysis, and vital signs. Table 2.1 lists a selection of the most representative papers for radar sensing in assisted living.

Paper [9] focuses on radar application in the healthcare domain. One is to estimate vital signs, such as respiration and heartbeat, the other is to monitor people's activities at home. The objective is to characterize these movements in three domains, range, time, and velocity, and explain the theories. The range is the distance at which the subject and his or her body parts are located with respect to the radar. Radar systems typically measure velocity through the Doppler effect. The radars utilized include continuous wave (CW) 24-GHz radar, UWB X-band radar, frequency-modulated continuous wave (FMCW) radars with 60-GHz, 9.8-GHz, 9.8-GHz. Paper [9] also mentioned that CW, FMCW, and UWB are related to the specific type of waveform transmitted and received

Table 2.1: Papers for radar sensing in assisted living, including specific tasks, radar types, and data formats

Paper	Domain	Tasks	Radar Type	Data Pre-processing
[9]	health-care	estimate vital signs, monitor activities at home	CW, FMCW, UWB	range-Doppler–time
[7]	HAR	security, remote health	CW	micro-Doppler signatures
[10]	health-care	heart/respiration monitoring	CW, FMCW	range-Doppler–time
[12]	human-machine interaction	micro-gesture recognition	UWB + 5 pressure sensors	micro-Doppler signatures
[13]	human-machine interaction	dynamic hand gesture classification	multistatic FMCW	micro-Doppler signatures
[14]	human-machine interaction	ASL recognition	UWB, FMCW	micro-Doppler signatures
[15]	human-machine interaction	ASL recognition, daily activities	FMCW	micro-Doppler signatures, range-Doppler

by the radar. After radar receiving sequences of EM waves, the next step is extracting information on the targets of interest.

Paper [7] is an overview of radar-based human activity recognition (HAR) for indoor monitoring, including security, remote health/telemedicine, health-computer interaction. In the research, the signal radar transmits is 4-GHz CW. Doppler–time pattern, also called spectrogram and micro-Doppler signature, is the most commonly used input data representation for motion recognition. The micro-Doppler signal is handled by short-time Fourier transform (STFT). It also refers to the cadence-velocity diagram (CVD) due to the STFT resolution tradeoff.

Paper [10] talks about heart/respiration monitoring for respiratory disease and sleep cycle classification and corresponding approaches in the radar domain. Time-varying respiration rate and heart rate can be estimated, which helps breathing disorders and sleep stages recognition. Breathing disorders have six breathing patterns while sleep stages have four classes (wake, rapid eye movement (REM), light sleep, and deep sleep) and 11 features from respiration rate and heart rate. Paper [10] discusses multimodal fusion, such as range-Doppler-time cube, and combines multimodal fusion with a machine learning algorithm. Fusion generally improves the overall classification performance, compensates for the shortcomings of a single sensor, and improves the sensitivity and specificity related to a specific category of interest.

Paper [12][13][14][15] specialize in hand gestures composed of the static part (when hand and finger are still) and dynamic conversion between static part. The process flow is almost the same as human motion and sign measurements. Different from the other three papers, hierarchical sensor fusion is emphasized in Paper [12], which means that not only UWB pulse-Doppler radar signal but also five wearable pressure sensors. Radar was 7.29-GHz UWB with approximately 1.5 GHz useful bandwidth. Pulse repetition frequency (PRF), which also impacts classification accuracy, was equal to 200 Hz. Paper [13] copes with multimodal sensing so that the emphasis is fusion strategy from different receivers. Fusion methods include soft fusion using confidence level and hard fusion using prediction label. Paper [13] also explored data fusion of different receivers. The radar system was a multi-static FMCW radar with carried frequency 24-GHz, a bandwidth of

500 MHz, and four-receiver antennas. The pre-processing method of the noncontact radar approach was multi-static radar micro-Doppler signatures.

Both Paper [14] and [15] tackled American sign language (ASL) recognition. This technology is designed for deaf communities to linguistic analysis. Deaf people rely heavily on technology as assistive devices. For gesture recognition, Paper [14] utilized UWB impulse radar with a transmission frequency range of 7.25-10.2 GHz as well as FMCW radars at 24 GHz and 77 GHz with a bandwidth of 1.5 GHz and 750 MHz, respectively. Pre-processing and representation of data was the micro-Doppler signature. Paper [15] explored measurements of not only hand but also arm and upper body kinematics for daily activity. The sensor was Texas Instruments 77-GHz FMCW with a bandwidth of 750 MHz. Data representations were micro-Doppler signatures and range-Doppler.

In summary, radar has various applications for assisted living, mainly including estimation of vital signs, hand-gesture recognition, and daily human-motion classification. There is a wide range of radar waveforms with different frequencies. Research in radar-based classification has explored the use of a wide range of radar waveforms for this purpose at transmit frequencies of 2.4, 4, 5.8, 7.29, 9.8, 24, 60, 77, and even 94 GHz, including CW, FMCW, and pulsed Doppler radar; UWB, ultrashort pulse, interferometric, and multi-static radar; dual-polarized radar [7]. Collected radar data can be processed to micro-Doppler signatures or range-Doppler-time for classification. In this thesis, the emphasis is human-motion classification on the UWB pulse radar.

2.2. HUMAN-MOTION CLASSIFICATION

After collecting radar data, it is necessary to generate the radar data patterns and use machine learning to automatically teach an algorithm to classify patterns related to different human-motion activities. From radar-specific signal processing to machine learning algorithms, classification algorithms includes quadratic-kernel support vector machines (SVM), k-nearest neighbors (KNN), random forest classifier (RFC), and linear discriminant analysis (LDA), and neural networks. Neural networks have also developed many different algorithms, such as CNN and recurrent neural network (RNN), auto-encoder network (AEN), fully convolutional network (FCN). Table 2.2 summarises the pre-processing and ML approaches of the main papers on human-motion classification. Dataset size is the number of samples/the number of classes, and the highest accuracy is indicated as taken from the results of each paper.

Table 2.2: Papers about human-motion classification, including related dataset and ML algorithms

Paper	Dataset Source	Data Pre-processing	Dataset Size	ML Algorithm	Accuracy
[16]	pulsed radar	range-time, HOG	862/16	SVM, KNN	97.7%
[17]	UWB	range-Doppler	1000/4	SVM	98%
[18]	UWB TWR	range-time	864/4	AEN, RNN	93%
[19]	CW TWR	-	1400/2	FCN	89%
[20]	FMCW	micro-Doppler signatures	24000/10	ResNet-18	97%
[21]	multistatic pulsed radar	micro-Doppler signatures	200/2	CNN	99%

Paper [16] is aimed at gross-motor activities and uses the histogram of oriented gradients (HOG), a kind approach from image processing to deal with radar range-time data. Paper [17] with the purpose fall motion detection exploits Doppler and range information (range-Doppler format). Both of them utilize classical ML methods, while the other four papers explore neural networks for classification. Paper [18] and Paper [19] adopt through-the-wall radar (TWR). Paper [20], and Paper [21] focus on human gait identification and classification and employ the same processing mode (micro-Doppler and CNN, ResNet is a sort of CNN architecture). Paper [21] also involves merging the fusion step. From Table 2.2, the main pre-processing approaches are range-time, range-Doppler, and micro-Doppler. It is hard to recognize which processing methods or ML algorithms are better due to inconsistent datasets.

Paper [7] shows an excellent example to evaluate these various ML approaches. It has discussed all kinds of knowledge-aided radar-signal processing techniques, such as handcrafted features, data-driven approaches, unsupervised pre-training and transfer learning, and 3D input representations. It explored several neural networks, such as CNN, autoencoders (AE), convolutional AE (CAE). An experiment was implemented in [7]. The radar emitted CW signal at 4 GHz. For human-motion recognition, there are 12 labels for training: walking, jogging, limping, sitting, walking with a cane, walking with a walker, walking with crutches, crawling on hands and knees, creeping while dragging the abdomen on the floor, using a wheelchair, falling after tripping, and falling off a chair. Each of the 11 participants performed 12 daily activities, and a total of 1007 measurements were collected. The pre-processing method is micro-Doppler signatures (spectrogram). It has assessed the performance of each model according to the experimental results.

Table 2.3 describes the test accuracy (ACC) of the various models on the same dataset on Paper [7]. Though 127 features are extracted, only 50 features are selected by sequential backward elimination. For model 2, principal components are performed from the 128*128 matrix of the spectrogram, and the size of testing sets is 10%. It can be seen that the performances of the deep neural network is better than that of traditional ML, and the last method, DivNet-15, has the highest accuracy.

Table 2.3: Accuracy comparison between ML algorithms in Paper [7]

	MODEL	ACC	REMARKS
1	multi-class SVM	76.9%	127 handcrafted features extracted
2	2D PCA + kNN	88.7%	15 principal components, k=3
3	three-layer AE	84.1%	-
4	three-layer CNN	90.1%	2*2 max-pooling, 50% dropout
5	CAE	94.2%	two fully connected layers, softmax classifier
6	VGGnet	90.8%	pre-trained on the ImageNet database
7	DivNet-15	**95.1**%	15-convolutional-layer residual neural network

In conclusion, the neural network performs better than other ML classical arts in terms of human-motion classification. Better in this context means that the value of the performance metric such as accuracy is higher compared to traditional ML algorithms, but also that the networks can perform implicitly feature extraction without explicit guidance from the human operator. Therefore, this thesis will focus on different radar data formats and neural networks.

2.3. Complex-Valued Neural Networks

Deep neural networks have broad prospects in the radar area. Nowadays, most neural networks are based on real-valued operations, drawing from the field of image processing where each pixel is associated to a real value or a vector of real values. However, complex numbers have more abundant representation ability, i.e. they also include besides absolute value the phase information, which in the electromagnetic world typically carries important information. CVNN architectures have been partially explored and implemented in fields dealing with complex numbers, such as speech, telecommunications and image [5][22][23]. This network should also be able to be applied to radar data processing and classification.

CVNN is based on the classical multi-layered type back-propagation network. A paper [5] in 2018 made the main contributions to complex batch normalization (BN), complex weight initialization, and complex convolution. More specifically, this paper has used complex blocks, weight initialization strategy, and current neural network algorithms to realize CVNNs, and was practiced in the experiment of end-to-end training scheme to prove that the performance of CVNNs is comparable to or better than the real-valued models. This technique involving complex blocks is called deep complex network (DCN). This paper tested the complex-valued CNNs on multiple visual tasks (CIFAR-10, CIFAR-100, and SVHN* dataset for image classification), music transcription (MusicNet dataset), and speech spectrum prediction (TIMIT dataset). For image classification, CVNNs only benefited on the CIFAR-100 dataset (0.86% decline in classification error) while the other two datasets remain almost constant on error, even became a little higher on error. As for the MusicNet dataset, the reported average precision increased from 69.6% to 72.9% by CVNNs based on deep plain CNN models while shallow CNN maintained the same precision. Paper [5] also constructed complex convolutional long-short time memory (LSTM) for speech spectrum prediction. Mean square error (MSE) on the TIMIT dataset decreased 0.28% on CVNNs. The problem Paper [23] analyses is that CVNNs are almost equivalent to a double-dimensional real-valued neural network, so it compares the generalization characteristics of complex-valued and real-valued feedforward neural networks. CVNNs seemed not to make the model's accuracy higher apparently in the speech and image domain.

Few algorithms exploit the complex-valued nature of radar data, so CVNNs for radar are potentially a good breakthrough to orient research towards. Table 2.4 illustrates some literature about CVNNs applied on radar data.

Paper [24] with the purpose of classifying the through-Wall human activities, have built complex-valued CNN models, which are the same structures as several classical CNN. The data was collected by UWB stepped-frequency continuous wave (SFCW) radar, and the radar data representation is range profiles (range-time) which contain amplitude and phase information. A moving target indication (MTI) filter has been applied. Dataset has 2750 samples with 11 classes. VGG16, the AlexNet, and the ZFNet are chosen

Table 2.4: Papers about CVNNs based on radar data and applications

Paper	Radar Type	Architecture of Neural Networks	CVNN Techniques
[24]	UWB SFCW radar	VGG16, AlexNet, ZFNet	DCN
[25]	SAR	ResNet	SurReal
[26] [27]	SAR	the plain CNN	DCN
[28]	SAR	ResNet	DCN, SurReal, multi-channel
[29]	FMCW radar	the plain CNN	DCN

as basic models, and the DCN approach from Paper [5] is applied to CVNNs. It shows an improvement of approximately 5.56%6.31% with excellent performance. The result of Paper[24] is encouraging with the reference value.

Paper [25] [26] [27] [28] adopt synthetic aperture radar (SAR) images. Paper [25] puts forward a novel convolution operator using weighted Fréchet mean (wFM) on a Riemannian manifold and a fully connected layer operator using the distance to the wFM for CVNNs. This approach is called SurReal. ResNet model is applied on MSTAR dataset on SAR with accuracy from 94% to 98%. Paper [26] is aimed at the classification of three different classes of drones. It exploits the raw signal and the spectrogram from micro-Doppler analysis on CNN models. Paper [26] indicates that CVNNs by DCN overpowered real networks when presented with a large yet complicated dataset. Paper [27] has almost the same structure except for the dataset. Paper [27] utilizes airborne SAR (AIRSAR) data and shows the classification error can be further dropped if employing complex-valued CNN.

All three CVNN techniques are explored in Paper [28]. Multi-channel architecture is actually a "trick" where complex numbers are treated as two-channel real numbers. It deals with ResNet 50-layer on MSTAR and SAMPLE datasets. SAMPLE dataset is for pre-training, and accuracy is about 83%. This paper presents that no CVNN substantially outperformed the magnitude only baseline, and the SurReal performs best among the three techniques. Paper [29] in autonomous driving domain obtained range-Doppler maps from FMCW radar and implemented the plain CNN on DCN, showing that computational complexity substantially improves all considered metrics.

In conclusion, CVNNs cannot always provide better performance than their conventional, real-valued counterparts when processing radar data. The success of each CVNN model is highly dependent on the nature of the dataset used, application addressed, as well as the architecture of the CVNNs themselves [30]. This leads to the research questions outlined in the following section.

2.4. RESEARCH QUESTIONS

This chapter briefly introduced radar application in assisted living, radar types, preprocessing methods, and classification algorithms related to radar. There is considerable interest in exploring radar domains for human-motion classification and outstanding questions, for example, how to select algorithms given its different operational conditions. The research problems formed in Section 2.1-2.3 can be refined and divided into several sub-questions using terms and techniques described in this chapter. Firstly, the main research question is:

Do CVNNs improve the performance of radar-based human-motion classification on CNN models?

In other words, is there any value in this specific radar application to also consider the phase information together with absolute value data? Then, this can be divided into the following sub-questions to be answered in this thesis.

- The nature of the dataset is crucial for the success of each CVNNs. Based on one specific type of radar, which kind of radar data representations are most suitable for CVNNs, range-time, range-Doppler, or micro-Doppler?
- CNN has a wide range of network architectures, from basic to complex ones. Which CNN structure with a certain characteristic makes CVNNs improve the performance of the model?
- There are several CVNN techniques to implement operations with complex numbers in networks, such as DCN, SurReal and multi-channel. Which is most proper for the CNN model and human-motion dataset?
- How to evaluate the performance of CVNN models, in addition to classification accuracy?

3

Theoretical Background

This chapter explains the background and technology of radar classification. Firstly, the properties and working principles of radar are discussed. The following section discusses the data formats generated by the radar and the pre-processing techniques. Then, the classification methods suitable for radar data in the field of machine learning are discussed. Finally, the most advanced classification method, neural networks, is explained, which sets a performance standard. Therefore, it is the baseline with which the new approaches explored in this thesis must compete.

3.1. Basic Radar Principle

Radar is an electromagnetic sensor. Despite the significant contributions of many others, it was Christian Hülsmeyer who designed, built, demonstrated, and patented the first such system [31]. There are many types of radars, such as CW, UWB and FMCW. One example is pulsed radar (also called pulse radar), which works by transmitting and receiving pulse sequences in a short time. During transmission, the receiver is turned off to prevent damage caused by high transmission power.

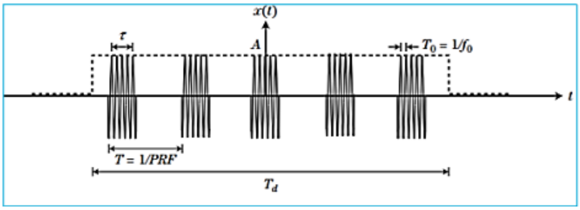
3.1.1. The Doppler Effect

If the radar and the target move relative, the echo frequency f_r received by the radar will be different from the radar transmission frequency f due to the Doppler effect. The equation 3.1 represents their relationship, which is also applicable to sound waves and other waves in the electromagnetic spectrum.

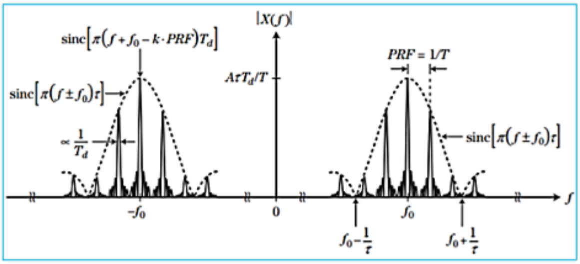
$$f_r = \left(\frac{1 + v/c}{1 - v/c}\right) f \tag{3.1}$$

The difference of f_r and f is the Doppler frequency f_d, also called the Doppler shift. To calculate f_d, the equation can be simplified to Equation 3.2, the sign "≈" holds on the

3



(a) time-domain representation



(b) frequency-domain representation

Figure 3.1: A finite pulse modulation sequence before and after Fourier transform [1]

condition that the moving speed of the object is far less than the speed of light.

$$\begin{aligned} f_d &= f_r - f \\ &= \frac{2v/c}{1-v/c} f = 2\left[v/c + (v/c)^2 + (vlc)^3 + \cdots\right] f \\ &\approx \frac{2v}{c} f \end{aligned} \tag{3.2}$$

Doppler shift also represents velocity. If we can estimate the Doppler shift f_d from the radar data, then the radial velocity v of the target can be also estimated. One way to measure the Doppler shift is Fourier transform (FT). There are different radar waveforms, including continuous wave, single pulse wave, infinite pulse train, finite pulse train, modulated finite pulse train. Figure 3.1 is a time-domain representation and a frequency-domain representation of a finite pulse modulation sequence. Figure 3.2 clearly shows the frequency shift of radar echo reflected due to the Doppler effect. Doppler resolution and Doppler ambiguity should also be considered in radar design.

3.1.2. THE MICRO-DOPPLER EFFECT

Besides the Doppler effect, there is also the micro-Doppler effect. When, in addition to the constant Doppler frequency shift induced by the bulk motion of a radar target, the target or any structure on the target undergoes micro-motion dynamics, such as mechanical vibrations or rotations. The micro-motion dynamics induce Doppler modulations on the returned signal, referred to as the micro-Doppler effect [32]. The bulk motion is responsible for the Doppler frequency shift mentioned in the last subsection,

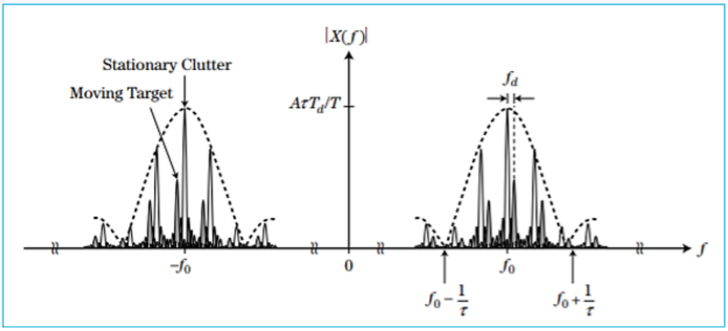


Figure 3.2: Frequency shift f_d from the radar echo [1]

while the micro-Doppler is due to micro-motion dynamics in addition to the normal movement of the target.

Examples of applications are the rotor blades of helicopters and unmanned aerial vehicles (UAVs), daily human activities, human hand gestures and sign language. The micro-Doppler is a new approach for analyzing target signatures to determine the dynamic properties of the target, so it is called the micro-Doppler signatures. The micro-Doppler features serve as additional target features, which is complementary to existing methods.

Micro-Doppler modeling is that object can be represented by one or more point scatterers, e.g., a human body decomposed into a series of triangular facets [32]. Models classify into rigid bodies and non-rigid bodies. Rigid bodies mean no deformation, such as helicopters, while the non-rigid human body is the deformable body. Therefore certain parts of non-rigid bodies can be modeled by two point scatterers connected by a rigid segment. The micro-Doppler frequency is added on top of the bulk Doppler. Our thesis focuses on human movements, a classic non-rigid body motion. The scattering from the entire human body may be approximated using superposition of the returns from all body parts. This rich information helps gain much understanding of what the "target" is doing. One example is human waling with the movement of the main body and torso, with additional back-and-forth oscillating movements of the arms. Different human will exhibit different walking patterns. By working on the human body walking gait, it can infer medically relevant information of the target, such as falls, seizures and staggering.

3.2. RADAR DATA FORMAT FOR PULSE RADAR

Different from the images captured by the camera, the signal received by radar is an inherently complex time series that is, in general, a time-delayed, frequency-shifted version of the transmitted signal [7]. For radar data domain representations, preprocessing data includes Fourier transform, time-frequency analysis, filtering, formatting, and generating 1D, 2D, or 3D images as input.

3.2.1. RANGE-TIME AND RANGE-DOPPLER MATRIX

To capture the Doppler frequency, we sample multiple pulses every pulse repetition interval. Pulse repetition frequency (PRF, $PRF = 1/PRT$) is much lower than the sampling time of the range bins. The raw pulsed radar data is seen as a 2D range-time matrix with column range profile and row the same range bin over consecutive pulses. Each row uses Fourier transform to calculate its spectrum to obtain the range-Doppler matrix, which has moving targets and static clutter and noise. Static clutter is the dominant return due to static targets. MTI filter can remove clutter. This processing method does not apply to all types of radars. For example, CW radar cannot measure range, and FMCW radar produces raw data needing the first fast Fourier transform (FFT) to extract range information.

3.2.2. TIME-FREQUENCY DISTRIBUTION

To extract micro-Doppler signatures, Fourier transform is the most common tool for relevant signal processing. Equation 3.3 transforms time domain $s(t)$ into its frequency domain $S(\omega)$.

$$S(\omega) = \int_{-\infty}^{+\infty} s(t) \exp(-j\omega t) dt \quad s(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S(\omega) \exp(j\omega t) d\omega \tag{3.3}$$

However, many real signals are non-stationary signals, which means they change their frequency within the observation time, for example, music signal. For human motion, each moving part is like a "note" in the Doppler spectrum because the change of velocity (the Doppler frequency) also changes with time. In addition, the Fourier transform does not tell "when" a specific frequency occurs.

Time-frequency (TF) distributions are necessary to process non-stationary signals, just like speech and audio analysis, to measure and highlight different Doppler shift frequencies that change over time. TF distributions include STFT, Gabor transforms, fractional Fourier transforms, wavelet transforms, and so on. The most used way is STFT, with its expression as follows:

$$STFT(t,\omega) = \int s\left(t'\right) w\left(t' - t\right) \exp\left\{-j\omega t'\right\} dt' \tag{3.4}$$

where $s(t)$ is wide sense stationary (WSS) signal in the time domain, $w(t)$ is a short-time sliding window function to limit the transformation to a short segment of the signal. Therefore changes in the frequency content over time can be characterized. If the input is a 1D vector with time, it generates a 2D matrix (joint function of time and frequency), in other word, a 2D image. The magnitude of STFT, or the square magnitude sometimes is spectrogram, the most commonly used TF representation in micro-Doppler analysis. Each row (Doppler bin) represents how a certain amount of Doppler frequency of the signal changes over time, while each column (time bin) represents what Doppler frequencies are recorded at a given time.

There are three hyperparameters of STFT: the duration of the window, overlap between two consecutive windows, and the type of window. Types of windows include Hamming, Hann, Gaussian, Kaiser-Bessel, Gaussian. A good window can reduce sidelobes and widen the main lob. There is a trade-off for the duration of the window between temporal resolution and spectral resolution in Doppler. The window duration is fixed. Short windows can grasp fast signals in time and movement of individual components, but frequency resolution (wideband) is poor, which blurs the frequency. Finding a compromise depends on the application and objects to observe to match the window duration to the target dynamic. For example, the required window length is very short to observe the time-varying characteristics of vital signs.

Figure 3.3 is an example of spectrogram image showing one person walking toward and away from the radar system. Beyond STFT, there are continuous wavelet transform (CWT) and Wigner-Ville distributions (WVD) [33]. These alternatives help break such trade-offs at the price of more complexity. Wavelet function, also called the scalogram, can be scaled in frequency while shifted in time. The WVD, which is one of Cohen's classes, is a time-dependent autocorrelation function. Considering the WVD has nasty cross-terms and does not have any physical meaning, pseudo-WVD and smoothed pseudo-WVD are emergent techniques to solve this problem. However, it is always real value, so it is not suitable for CVNNs. Therefore, this is not the focus of this thesis and will not be discussed and expanded.

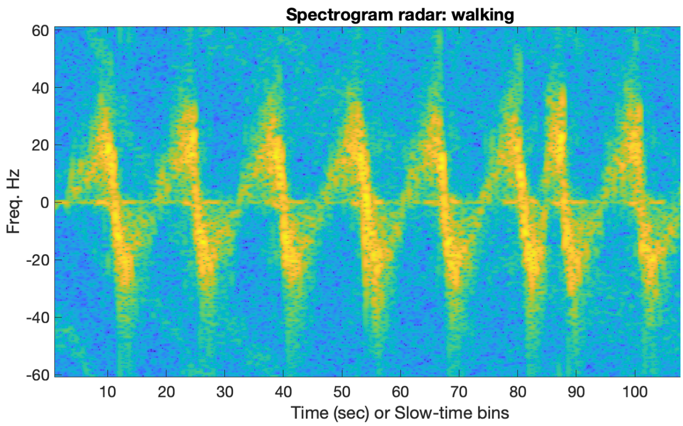


Figure 3.3: Spectrogram from radar data representing human walking motion in two minutes

3.3. SUPERVISED CLASSIFICATION

3.3.1. SUPERVISED CLASSIFICATION PIPELINE

In this subsection, the focus is on the usage of radar-based object classification. Feature extraction and classification problem can be tackled by machine learning techniques in the supervised learning framework. In object-based image analysis process, there are segments from the images of objects and then training data is collected. Then a classifier is instantiated, (e.g.based on spectral information, texture, shape) and is trained using the training data. Finally, performance of the classifier is assessed by test set. Figure 3.4 illustrates ML model training and evaluation process.

The more theoretical steps are explained in detail. The first step of general pipeline is still inputting processed data samples with labels. The second step is parametric function or model decision to get expected outputs. The third step is loss function decision

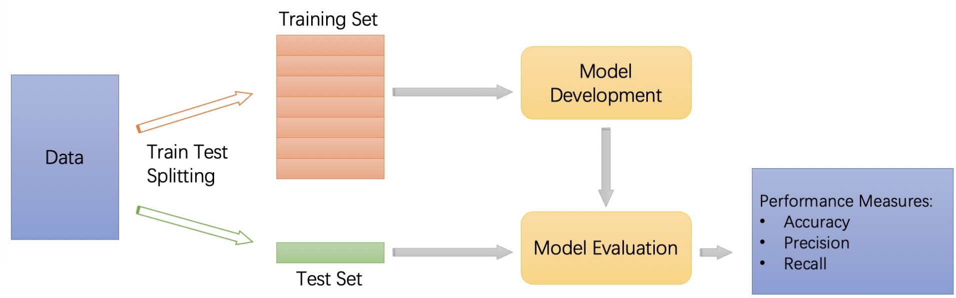


Figure 3.4: Model training and evaluation process based on supervised classification

to evaluate how well the function or model is learning. The forth step is training to find the best parameters of function or model using optimizer, such as stochastic gradient descent and finally testing, which means to assign an unknown test data to a certain class of interest and doing evaluation. As for the input, micro-Doppler (Doppler-time patterns) is a good data format for object classification since it captures the specific movement of individual parts of the object to identify while other formats can also be tried as inputs to a classification algorithm utilising radar data.

3.3.2. FEATURE EXTRACTION

Features extracted from TF distributions are discriminating properties of the object signatures. By features, machine learning is an efficient way to classify objects of interest. Handcraft features are features chosen with the physical or kinematic meaning of the micro-Doppler spectrogram or the raw, complex radar data as the input of the machine learning model, such as mean period and bandwidth. There are also image-inspired and speech-inspired features, e.g., textural features and HOG [16]. Good features must be relevant to the classification problem and suitable features can mitigate the issues of the "curse of dimensionality" and overfitting. These features can be divided into three categories:

- Physical features: These have physical or kinematic meaning, such as velocity and stride rate, mean period, Doppler offset, radar cross-section (RCS) limbs and body ratio.
- Transform-based and decomposition-based features: These are calculated from the micro-Doppler spectrogram or on the raw complex radar data, such as centroid f_c, bandwidth B_c, singular value decomposition (SVD) of the spectrogram, discrete Fourier transform (DCT) coefficients.
- Speech-inspired and image-inspired features: These features originally designed for speech or image processing. Features arise from moments and texture of the radar data image, such as mel-frequency cepstrum coefficients, linear predictive coding (LPC), Pseudo-Zernike and Krawtchouk moments [34].

Data-driven feature-learning methods are extracting features from the training data.

To reduce dimensionality, it compresses input data that are highly dimensional into a reduced-size representation as features. One common method is principal components analysis (PCA), a linear transformation maximising variation in the training dataset [35]. It is generally used to represent a given input using less number of dimensions than originally present. The strategy is to find the direction with the most variance and the resulting first N vectors to lower dimensionality. This technique projects the N*M spectrogram into a smaller data structure. There are also some pre-class activities to increase classification accuracy, such as feature scaling and standardization.

There are many features, so features chosen must be suitable for a specific classification problem. Interpreted as matrices of pixels, the input data format is a kind of image, so there are many mature approaches from image classification as a reference. Another method is neural networks inspired by image and speech processing research. Neural networks have advantages to perform inherently feature extraction together with classification, so it has no explicit feature extraction block. In other words, deep supervised learning with neural networks enables to "skip" some of the processing steps above, such as feature extraction, so that typically the network takes care of them. We will talk about this method in detail later.

3

3.3.3. MACHINE LEARNING ALGORITHM

ML algorithm for classification is also called classifier. Classic classification algorithm has two categories: generative models and discriminative models. Generative models include naive Bayes, KNN, nearest mean, while typical discriminative models are decision trees, SVM, and logistic regression. We take two simple and easy classifiers, SVM and KNN, for examples. SVM is a two-class classification model with the primary thought to find a linear classifier with maximum separation hyperplane in feature space [36]. It also can be extended to the multi-class classifier. The strategy of KNN is that an object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small) [37]. After extracting handcrafted features or PCA, we can apply the features to these classifiers and get the final results by training.

Beyond classic ML algorithms, neural networks belong to a broader family of machine learning algorithms that progressively use multiple layers to extract higher-level features from the raw input [38]. It maps the input to a high dimensional feature space which is the learnt representation. Interested in skipping some steps automatically, neural networks is implemented in radar areas. This powerful classifier has a crucial advantage with feature extraction, or selection inherently performed together with the classification without human operator input, sometimes called "feature engineering".

3.4. NEURAL NETWORKS

This section presents neural networks as alternative classification algorithms to the conventional supervised learning algorithms described in the previous section, such as SVM or KNN. Neural networks are attractive because of their capabilities to capture distinctive patterns in the data without explicit, manual feature extraction. This section provides an overview of basic concepts and architectures of neural networks, whereas a more detailed discussion of the complex-valued neural networks developed in this thesis is given in Chapter 4.

3.4.1. ANN MODEL

There are different structures of neural networks, but they always consist of the same components: neurons, connections, weights, biases, and functions. The neural network can be interpreted as a nonlinear extension to PCA, which could be implemented using a one-layer artificial neural network with hidden linear units. Since the expressiveness of linear function is limited, activation units (nonlinear function) are necessary. The ordinary neural network is an artificial neural network (ANN) shown in Figure 3.5.

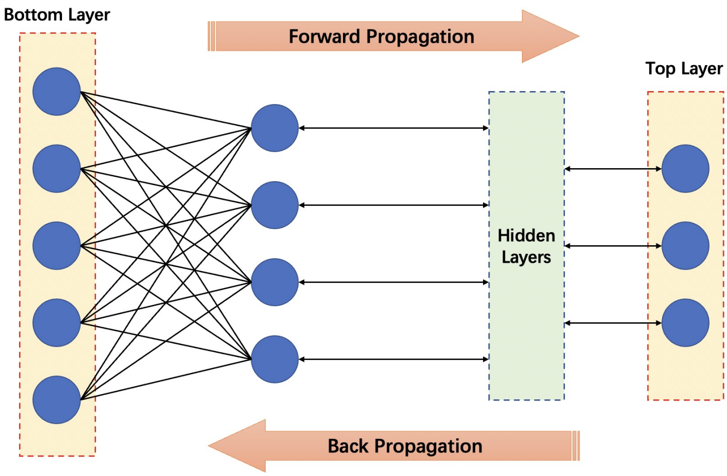


Figure 3.5: Regular ANN architecture and propagation mechanism

The basic block of the neural network is parametrized linear functions (neurons) and non-linear activation functions. Parameters of the linear functions are optimized with respect to a loss function. The loss function is differentiable, so gradient-based optimizers, e.g., stochastic gradient descent, are computed to minimize loss. Forward propagation is moving from the bottom/input layer (left) to the top/output layer (right) to update neurons in the neural network. The process of moving backward from the top to the bottom layer is called backward propagation. During training, weights and bias update every epoch by gradient descent is backward propagation process.

Deep learning (DL) is part of a broader family of ML methods based on ANNs [39], also called deep neural network (DNN). The adjective "deep" in deep learning refers to the use of multiple layers in the network shown in Figure 3.6. It can extract features and patterns in the data automatically with high accuracy and is very promising in many other fields, such as images and sounds. The drawback is needing much data to train and problems of overfitting. Besides it is hard to understand and explain their internal decision process.

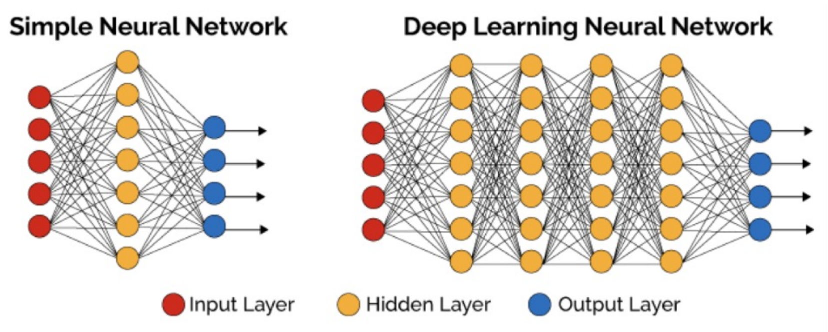


Figure 3.6: Artificial neural network vs Deep learning neural network [2]

3.4.2. TYPES

Several neural network architectures for radar signals are as follows:

- Convolutional Neural Network: CNN is primarily used for image processing but can also be used for other input types. This input data is fed through convolutional layers and pooling layers instead of normal layers. Pooling is a way to filter out details, such as max-pooling for local maxima. When features have been extracted from previous layers, the next step is fully-connected and get outputs. Famous CNNs from the image processing community, such as Alexnet, VGG, GoogleNet, ResNet, can be reused for radar [7]. One disadvantage of CNN is that a large training database is required.

- Autoencoders: Autoencoders are neural networks designed for unsupervised learning, meaning the data is not labeled. AE can minimize datasets. As a data-compression model, they can encode a given input into a representation of a smaller dimension. The work they do is very similar to PCA. AE aims to approximate the identity operation using a symmetric encoder-decoder structure. The cost function in the regularizer is Kullback-Leibler divergence (KLD).

- Convolutional AE: CAE is a two-stage process combining convolutional filtering in CNNs and unsupervised pretraining of AE [40]. AE is used first to initialize network parameters, and then supervised fine-tuning with labeled measured data is used to optimize final values [7].

- Recurrent Neural Network: RNNs have connections between passes, connections through time, which is suitable for the continuous stream of motions. One big problem with RNNs is the vanishing or exploding gradient problem. LSTM and gated recurrent unit (GRU) can solve this problem [41].

It is worth mentioning that when the input is a matrix with M*N pixels, some pixels are not conveying information or are noisy. Clutter-suppression and noise-reduction techniques are needed for the classic classification algorithm, for example, MTI filtering or high-pass filtering can remove the contribution of the static clutter to generate spectrograms with the best signal-to-noise ratio (SNR). However, filtering to remove clutter weakens neural networks' performance because neural networks can extract information from signal components not masked by clutter.

3.4.3. EVALUATION AND CHALLENGES

Accuracy, speed of training, complexity, and interpretability should be considered to judge these methods. Complexity is how large the classifier is in memory, and interpretability is a descriptor of how easy it is to explain for humans how the classifier works, which is a crucial aspect for many industrial applications. Confusion matrix (CM) in Figure 3.7 can assess the classifier's performance with accuracy, precision, and recall. Precision is the proportion of correct predictions over all predictions for a given class.

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive** (TP)	**False Positive** (FP)
	Negative	**False Negative** (FN)	**True Negative** (TN)

Figure 3.7: Confusion matrix for two-class classification problem

Recall is the proportion of all correctly predicted samples of class K over all true samples for such class. The expressions of four performance metrics are below, where F1 score is applied to solve the trade-off between precision and recall.

$$\begin{aligned}
\text{accuracy} &= \frac{TP+TN}{TP+FP+FN+TN} \\
\text{precision} &= \frac{TP}{TP+FP} \\
\text{recall} &= \frac{TP}{TP+FN} \\
F_1 &= 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}
\end{aligned} \tag{3.5}$$

As for performance metrics in the multiclass case, one way is the simple average, also called macro, which means all classes have equal importance. Another choice is a weighted average, accounting for the frequency of the classes in the data to correct potential imbalances of data [42]. There are some drawbacks of neural networks:

- Much data is needed for training neural networks when the network is deep.
- Interpretability is poor. Interpretability is a descriptor of how easy it is to explain for human how the classifier works, and why it makes certain predictions. The internal algorithmic steps are not easy to explain and justify.
- Overfitting and underfitting problems often exist. Overfitting is that the model performs well on the training data but poorly at the validation stage. If the model performs poorly on both training and validation, it is underfitting which means the classifier is too simple. It is not easy to find the best optimal neural layer numbers.
- Some information is rooted in known EM physics and does not need to be learned from the data, e.g., neural networks can not implement Fourier transform and

learn frequency from raw data. Therefore, it is necessary to add Fourier transform step as pre-processing artificially. It seems that the neural network is not so "smart" and intelligent.

3.4.4. ADDITIONAL TECHNIQUES FOR NEURAL NETWORKS AND RADAR DATA

In order to improve neural network's performance, there are several other techniques.

1. During training, another evaluation strategy is cross validation, including one time split, K-fold cross validation and leave-out validation. The latter two are recommended, because the result of the first one might be biased. K-fold cross-validation is that training dataset is divided into K portions and one is used for validation of the classifiers. Testing set is to assess the generalisation performance of the classifier on new, unseen data. Performance metrics are averaged across the K validations performed.

2. Deep network has the problem of gradient vanish or explosion, which makes the deep learning model difficult to train. Batch normalization can alleviate this problem [43]. Batch normalization is achieved through a normalization step that fixes the means to zero and variances to one of each layer's inputs.

3. Tuning hyper-parameters can improve the performance. One way of hyperparameter optimization is grid search, which is simply an exhaustive searching through a manually specified subset of the hyperparameter space.

4. Training a model needs a set of labeled data. The more data the better, however, it is hard to get enough experimental data sometimes. One strategy is transfer learning, using a network that has been pre-trained with data from a different domain, e.g., optical images, then followed by fine tuning with a smaller amount of radar data.

5. Limited by amount of training data, another solution is micro-Doppler signatures simulated by Motion Capture (MOCAP). MOCAP data acquired from RGB-D (red, green, blue, depth) cameras as the kinect sensor, can get 3200 samples from 55 MOCAP (body parts) [7]. This kind of simulated data is good enough for training an algorithm even if not perfectly identical to experimental data, while the real radar data is applied to fine-tune and test.

6. Generative adversarial network (GAN) is another new approach that generates new data having two networks playing against each other, which is popular in image domain recent years [44]. The generator tries to make synthetic data from a small set of real data while the discriminator tries to establish if such data is real or not. However, GAN is unrelated to kinematics, and does not emphasis on properties of data. This approach may not be suitable for the micro-Doppler Signature data simulation.

3.5. CONCLUSION

This chapter illustrates theoretical background of radar data and ML classification in details. For example, for the pre-processing step, the general pattern is the micro-Doppler signature. For machine learning algorithms, classical models including SVM, KNN, and neural networks are introduced. Generally speaking, a complete process of classification problems is as follows: firstly, define the goal and what class to categorize; secondly, do experiments to collect enough data and label the collected samples; thirdly, clean the data because some of the data may be wrong and explore the collected data; and then, pre-process the data to find the appropriate features. The next step is to determine the ML model. There are many kinds of ML models, so it is necessary to find a suitable one according to the characteristics of the data. The last step is to train the model with the data and validate the model. The pipelines in Figure 3.8 is not only the radar-based classification problem but also the process of other ML problems such as regression problems and unsupervised learning.

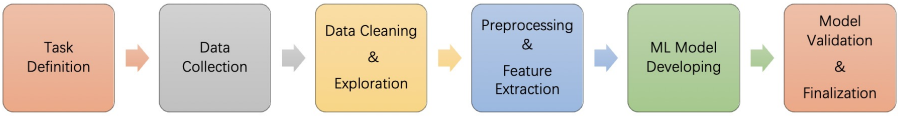


Figure 3.8: Machine learning pipeline

4
Methodology

This chapter pays attention to the techniques developed and investigated in this thesis project to classify human activities, mixing signal processing applied to radar sensors, and neural network algorithms. Specifically, a detailed discussion of the constituent blocks of CNNs is provided, as they are among the most successful network architectures used for pattern classification. Furthermore, the proposed implementations of complex-valued blocks are discussed.

4.1. Radar Data Representations

Based on the principles of radar systems, transmitting and receiving sequences of electromagnetic waves will extract information on the targets of interest. Different formats of the same data can be applied as input to neural network, and we will discuss and classify these data formats in detail below.

- Domain representations: The different domains of radar data can be exploited for objects classification. Dimensionality can be 1D as the high-resolution range profiles. The 2D data format can be a range-time matrix, range-Doppler matrix, or Doppler-time matrix. The input data representations can also learn the high dimensional mapping between the raw data and motion classes, such as 3D range-Doppler-time data cube. These joint-domain presents the time-varying range and Doppler frequency information. Micro-Doppler is typically observed as a 2D cut for Doppler-time. Doppler–time pattern is the most commonly used input data representation for motion recognition because human activities will exhibit different patterns in these frames.

- Different time-frequency distributions: Time-frequency analysis is necessary for micro-Doppler analysis. STFT is the most common, also called the micro-Doppler signature. Spectrograms highlight the variation in Doppler shift induced by human motion. There are other methods such as Gabor transforms, and wavelet transforms.

- Real-valued vs Complex-valued data: These are complex numbers in some data format, such as spectrograms. The absolute value is used to plot the data and the input to ML algorithms, but the phase still has important information to exploit for classification. Complex numbers as input to ML algorithms will be applied and compared in the thesis.
- Other formats: Another format exists. For example, the range-angle pattern is the distance and azimuth of objects used for automotive radar. The CVD takes a further FFT along with the Doppler frequency bins of spectrograms, highlighting the periodicity of micro-Doppler modulation over time [7].

We will implement three domain representations, range–time, range-Doppler, and Doppler–time by STFT in the experiment. Besides, the imaginary part of these complex data will be saved instead of taking the absolute value. Note that the resolution of the radar depends on parameters such as bandwidth and observation time. Even though as high resolution as possible, the hardware limitations and the computational burden should be considered.

4.2. CONVOLUTIONAL NEURAL NETWORKS

4.3. OVERVIEW

Inspired by the way the visual cortex of the human brain works, the convolutional neural network is firstly a tool to perform image processing with input data treated as a 2D matrix of pixels [45]. Now CNN is also one of the most used neural network architectures to process radar data. One unique block of ConvNet is convolutional (CONV) layer, followed by some form of non-linear activation function, e.g., rectified linear Unit (ReLU) layer and pooling layer. After these layers, the initial input data is compressed into a smaller-dimensional representation (flatten layer in 4.1), interpreting that as features for classification.

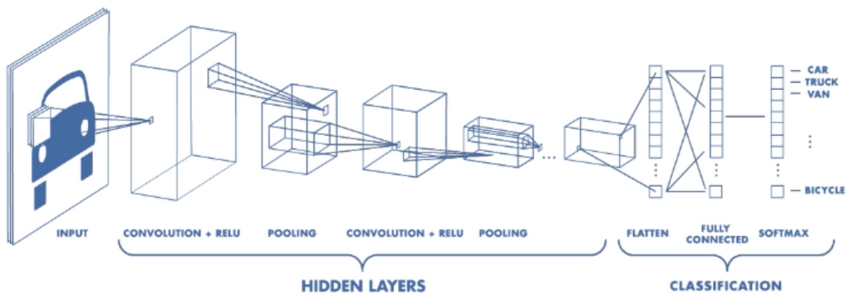


Figure 4.1: Example of architecture of a classic convolutional neural network [3]

4.3.1. ANN AND CNN

CNN is very similar to ANN except the neurons in the CONV layer connected only to a small local region shown in 4.2 instead of all of the neurons in a fully connected manner.

Neurons in both fully-connected (FC) layer and CONV layer compute dot products.

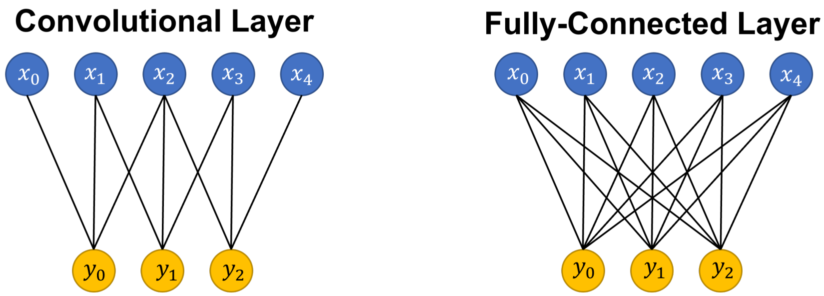
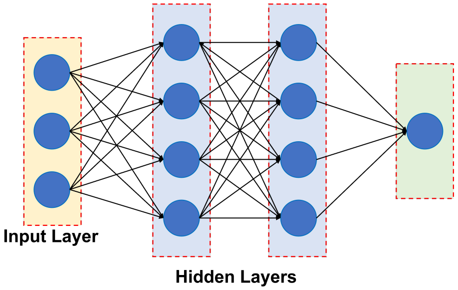
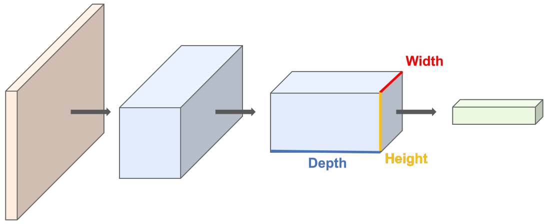


Figure 4.2: CONV layer VS FC layer

If the input holds the raw pixel values of the image, in this case, an image of width, height and with several channels, such as three color channels, R, G, B, it must be flattened as the input of ANN while CNN does not need it (Figure 4.3. CONV layer can handle this format of multi-dimension. Considering the format and complex nature of radar data, input data is seen as two-dimensional image with two channels, the real and imaginary parts, or amplitude and phase.



(a) ANN



(b) ConvNet

Figure 4.3: ANN model VS CNN model

4.3.2. CNN BLOCKS

The basic blocks of CNN are explained below in details.

- **CONV layer**: It computes the output of neurons that are connected to local regions in the input, each computing a dot product between their weights, seen as filters, and a small region connected to in the input volume. Four hyperparameters control the size of the output volume, including the depth (number of filters) K, stride (slide jump) S, and zero-padding convolution (number of zeros around the border) B, and filter extent (size of filters, eg 3*3) F. Another hyperparameter is bias. Parameter sharing of CONV layer is that if one feature is used to compute at some spatial position (x1,y1), it should also be useful to compute at a different position (x2,y2). If the CONV layer accepts a volume of size W1×H1×D1 as input (W1 is width, H1 is height, D1 is number of channels), a total of (F*F*D1)*K weights and K biases needs learning.

- **ReLU layer**: It applies an elementwise non-linear activation function. There are several kinds of activation functions, such as sigmoid, logistic. ReLU function is just one of them, and it is simple with the result thresholding at zero, leaving the size of the volume unchanged. The expression is below.

$$f(x) = x^+ = \max(0, x) \tag{4.1}$$

  Non-linear activation function is very important. Without nonlinear activation function, the output of each layer is always a linear function result of the upper layer. Therefore, no matter how many linear layers the neural network has, the output is still a linear combination of inputs that equals the most primitive perceptron. The nonlinear function is introduced as the activation function, so the deep neural network becomes meaningful (it is no longer a linear combination and can approximate any function). The earliest idea was the sigmoid or tanh function, but the amount of calculation was large and the gradient vanished [46]. If the ReLU function is activated, the calculation amount of the whole process is saved a lot. Besides, ReLU makes the output of some neurons 0, which leads to the sparsity of the network and alleviates the overfitting problem.

- **Pooling layer**: It performs a downsampling operation along the spatial dimensions, resulting in a smaller volume. It has two hyperparameters, the spatial extent F and the stride S. Pooling layers are uncommon to pad the input using zero-padding. Two common pooling operation are max pooling to calculate the maximum value for each patch, and average pooling to calculate the average value for each patch.

- **FC layer**: It may compute the class scores, resulting in a volume of size [1x1xN], where each of the N numbers corresponds to N categories. Each neuron in this layer will be connected to all the numbers in the previous volume.

Moreover, they still have a normalization function (e.g., SVM/softmax/sigmoid) on the last layer. Softmax and sigmoid are used not only in logistic regression, but also as the last activation function of a neural network to normalize the output of a network to a probability distribution over predicted output classes. When implementing the project, we chose the sigmoid function as the output layer, and its expression is as below. The output range is (0,1), representing the output probability, and the derivation is easy.

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} \tag{4.2}$$

4.3.3. CNN MODELS

Typical ConvNet architecture consists of CONV layer, ReLU layer, pooling layer and FC layer. The flow chart is 4.3, where N, M, and K determine the complexity of the model. The question mark '?' of pooling means pooling could be stacked or not. After flattening, the previous volume can be accepted to the FC layer.

$$Input \rightarrow [(Conv \rightarrow ReLu)*N \rightarrow Pooling?]*M \rightarrow Flatten \rightarrow (FC \rightarrow ReLU)*K \rightarrow Softmax \tag{4.3}$$

4

Figure 4.4 is an example of an actual ConvNet model. It is a plain CNN model with a straight pipeline. A basic CNN with few CONV layers, called shallow ConvNet. With stacking more layers, such as CONV layers and ReLU layers, it transforms to a deep ConvNet with more hyperparameters to tune, requiring many data for training.

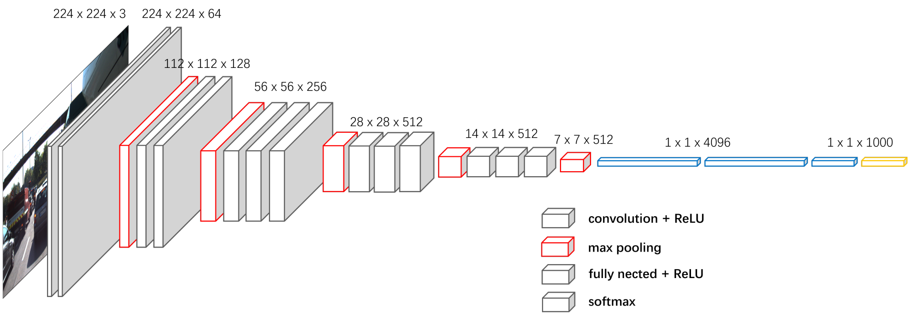


Figure 4.4: An example of ConvNet architecture

Theoretically, the deeper the network, the higher the accuracy. The reason is that after the neural network, adding a network layer that can be transformed identically will only keep the accuracy invariable or higher. However, experiments show that when the network is deeper, the accuracy is continuously improved, reaches the maximum value, and then decreases [4]. This phenomenon is also called "degradation", because it is difficult to realize "identity transformation (y = x)" due to nonlinear transformation. Besides, it is hard to acquire enough radar data, so more relatively shallow CNNs tailored for radar data. The complexity of these networks is relatively low and can avoid overfitting.

Besides the plain neural networks, there are more sophisticated architecture of CNN from the image processing community, such as AlexNet, VGG, GoogleNet, Residual Net (ResNet). One of the most famous CNN models is ResNet, proposed by Dr. He Kaiming in 2015. It is a milestone in CNN image history. It has "shortcut connection" to solve the degradation problem due to nonlinear activation function [4]. When the residual is 0, the stacking layer only makes identity mapping. At least, the network performance will not decline. The structure of residual learning is shown in Figure 4.7. ResNet is somewhat similar to the "short circuit" in the circuit. As shown in Figure 4.7, there is a

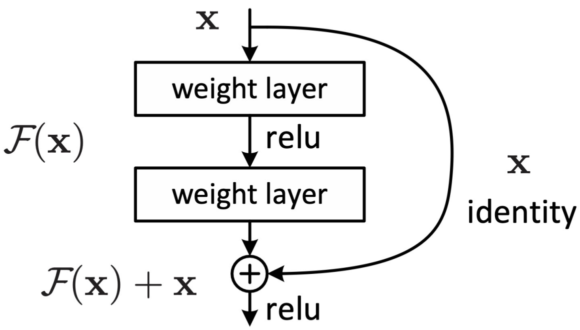


Figure 4.5: Residual learning: a building block in [4]

4

degradation problem in the depth network, that is, when the network depth increases, the network accuracy saturates or even decreases. This is not an overfitting problem, but a degradation problem of deep network. When a new layer is stacked up to establish a

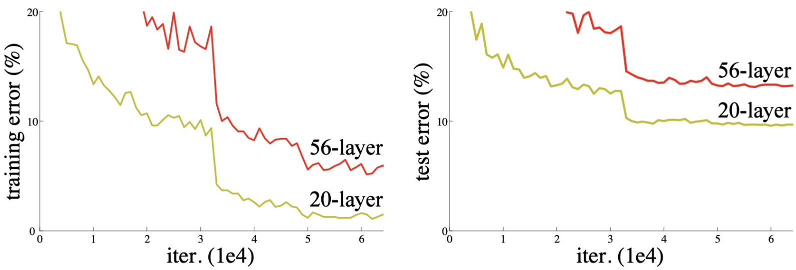


Figure 4.6: Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks [4]. The deeper network has higher training error, and thus test error.

deep network, an extreme case is that the added layer learns nothing and only copies the characteristics of the shallow network. The new layer is identity mapping. When the input is x and the learned feature is recorded as H(x), so the residual F(x) = H (x) - x. When the residual is 0, the stacking layer only makes identity mapping; at least, the network performance will not decline. In fact, the residual will not be 0, which will also make the stacking layer learn new features based on the input features to have better performance. General pre-defined ResNet models include ResNet18, ResNet34, ResNet50, ResNet101 and ResNet152. The numbers in the models represent the number of CONV layers in the network. The larger the number, the higher the complexity of the network. Hyper-parameters of building blocks are in brackets with the number of blocks stacked in Figure 4.7.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 4.7: Various ResNet architectures in [4]

4.4. COMPLEX-VALUED CNNS

4.4.1. DEFINITION

As we mentioned in the last section, the neural network is a supervised learning algorithm for classification, and it is common and efficient. Usage of neural networks in radar based human monitoring is usually based on real-valued operations and representations. Actually complex numbers could have a richer representational capacity, but have been marginalized due to the absence of the building blocks [5]. Generally speaking, if the input of a neural network is complex numbers, this neural network is called complex-valued neural networks (CVNNs). Because the information of general neural networks is real numbers, their basic calculation rules in the network are also based on real numbers. Many operations cannot be carried out directly anymore when the input is complex, so the computation rules need to be modified to adapt to the complex.

4.4.2. CVNNS

Recent years, more researchers have noticed the importance of complex numbers and some have built complex neural networks [22] [23]. We only discuss approaches of CVNNs based on CNN models to classify human motion. There are three main approaches of CVNN, multi-channel approach, DCN, SurReal for complex-valued input.

MULTI-CHANNEL ARCHITECTURE

Inspired by separate RGB images into three channels, the multi-channel approach splits either the magnitude and phase or the real and imaginary parts of data into two channels. In other words, complex input is regarded as two-channel image (absolute value phase or real imaginary). This solution is quite simple but loss the relation of two parts, since both channels are neither independent. For example, in the first CONV layer, the output of a filter is the linear summation of the two channels. The simple summation is not reasonable and does not respect the intrinsic geometry structure of the space of complex numbers.

DEEP COMPLEX NETWORKS

DCN is based on the theoretical developments introduced by Paper [5] in the context of image classification. The main idea is to build complex building blocks which support complex-convolution, complex batch normalization, and so on. Based on CNN, deep complex networks consist of complex convolution layers to compute the output of neurons that are connected to local regions in the input, complex-valued activation layers, complex batch normalization, and complex-valued pooling layers. Using these blocks, we can implement the CVNNs in a manner similar to the real-valued CNN models. We will introduce the specific operation process of each kind of layer in detail later. By the way, this approach is proposed by Paper [5], so the name of this approach, DCN, is the abbreviation of this paper's title. "Deep" does not mean that only DNNs can use this approach, it also can be appied to a shallow real-valued network.

SURREAL

This method is providing a novel convolution operator using weighted Fréchet mean on a Riemannian manifold and a novel fully-connected layer operator using the distance to the wFM [25]. The architecture is several wFM convolutional layers are stacked as bottom layers then followed by the real-valued neural network. The input must be absolute value phase format.

4.4.3. DEEP COMPLEX NETWORKS

The concept of DCN comes from paper [5]. DCN is not a special CNN model with a new architecture, but updates each layer based on the real-valued CNN architecture, such as upgrading the convolution layer to the corresponding complex-convolution layer to adapt to complex operations. Paper [5] provides the mathematical foundation of a set of building blocks for deep complex-valued CNN, so it is named DCN. In fact, not only deep neural networks can use this technique, but shallow neural networks are also very suitable. Besides, key atomic components and building blocks for DCN can be applied to both CNN and LSTM. Complex number is $z = a + ib$, and N features is divisible by 2, so N/2 represent the real components and N/2 represent imaginary components. The input form must be real imaginary fashion given certain constraints and interactions [26]. Complex building blocks are presented below.

COMPLEX CONVOLUTION

A complex filter matrix $W = A + iB$ is convolved by a complex vector $h = x + iy$, so that $W * h = (A * xB * y) + i(B * x + A * y)$. Figure 4.8 is the illustration of the complex convolution.

COMPLEX DIFFERENTIABILITY

Complex differentiability means 'holomorphic'. The requirement of holomorphic functions is satisfying Cauchy-Riemann equation, $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}$ where $f(z) = u(x, y) + iv(x, y)$. The activation functions are usually complex differentiable that can be used in complex valued neural networks. Paper [5] and [26] also mention that it is unnecessarily restrictive to limit oneself only to holomorphic activation functions, because those functions that are differentiable with respect to the real part and the imaginary part of each parameter are also compatible with backpropagation.

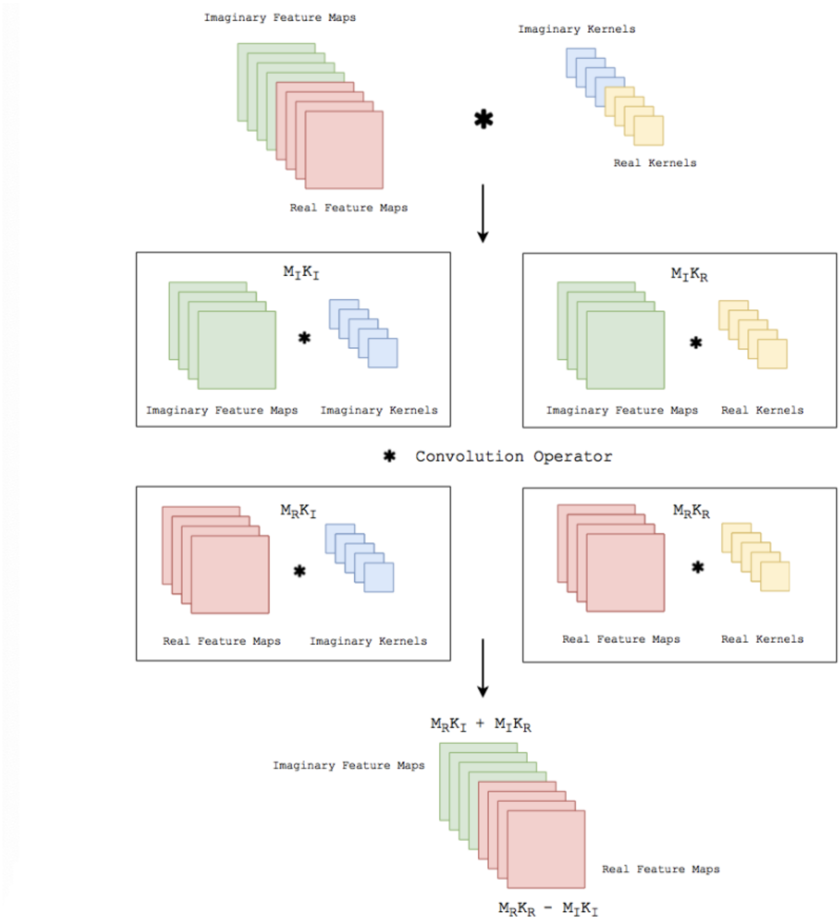


Figure 4.8: An illustration of the complex convolution operator in [5]

COMPLEX-VALUE ACTIVATION

Complex-value activation functions is usually ReLU-based. There are three kinds of complex-value ReLU functions: modReLU(z), Complex ReLU (or CReLU) and zReLU. The first one is not holomorphic while the last two are holomorphic. Three functions are defined as follows:

$$\mathrm{modReLU}(z) = \mathrm{ReLU}\, U(|z|+b)e^{i\theta_z} = \begin{cases} (|z|+b)e^{i\theta_z} & \text{if } |z|+b \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

$$\mathrm{CReLU}(z) = \mathrm{ReLU}(\Re(z)) + i\,\mathrm{ReLU}(\Im(z)) \tag{4.5}$$

$$z\mathrm{ReLU}(z) = \begin{cases} z & \text{if } \theta_z \in [0, \pi/2] \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where θ_z is the phase of z. b is a real number and is the hyperparameter of modReLU(z).

COMPLEX POOLING LAYER

For feature cube $z = x + iy$, x and y are operated separately. The example below is for max pooling. Average pooling has the same operational mode.

$$\text{ComplexMaxPool}(z) = \text{MaxPool}(\Re(z)) + i\,\text{MaxPool}(\Im(z)) \tag{4.7}$$

COMPLEX BATCH NORMALIZATION

Considering input has real and imaginary components, it can be seen as whitening 2D vectors $z = (x, y)^T$. This technique was firstly used in multi-channel signal processing area. The expression is shown below.

4

$$\tilde{\boldsymbol{z}} = (\boldsymbol{V})^{-\frac{1}{2}}(\boldsymbol{z} - \mathbb{E}[\boldsymbol{z}]) \tag{4.8}$$

where the covariance matrix V is

$$\boldsymbol{V} = \begin{pmatrix} V_{rr} & V_{ri} \\ V_{ir} & V_{ii} \end{pmatrix} = \begin{pmatrix} \text{Cov}(\Re\{\boldsymbol{z}\}, \Re\{\boldsymbol{z}\}) & \text{Cov}(\Re\{\boldsymbol{z}\}, \Im\{\boldsymbol{z}\}) \\ \text{Cov}(\Im\{\boldsymbol{z}\}, \Re\{\boldsymbol{z}\}) & \text{Cov}(\Im\{\boldsymbol{z}\}, \Im\{\boldsymbol{z}\}) \end{pmatrix} \tag{4.9}$$

$(\boldsymbol{V})^{-\frac{1}{2}}$ is calculated as follows, where Λ is a diagonal matrix of eigenvalues and U is a matrix whose columns are the corresponding right eigenvectors. The eigenvector in matrix U is normalized (the Euclidean norm of the vector equals to 1) so that $U^H U = I, U^H = U^{-1}$. $\Lambda^{\frac{1}{2}}$ is the diagonal matrix of the square root of eigenvalues. This calculation holds if V is a positive semidefinite matrix.

$$\begin{aligned} &V = U\Lambda U^H = U\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}U = \left(U\Lambda^{\frac{1}{2}}U^H\right)\left(U\Lambda^{\frac{1}{2}}U^H\right) = V^{\frac{1}{2}}V^{\frac{1}{2}} \\ &V^{\frac{1}{2}} = U\Lambda^{\frac{1}{2}}U^H \\ &V^{-\frac{1}{2}} = \left(V^{\frac{1}{2}}\right)^{-1} \end{aligned} \tag{4.10}$$

After batch normalization, z has the standard complex distribution with mean 0 and covariance $I = [1, 0; 0, 1]$. The normalization step allows the imaginary and real parts of a unit decorrelated. This has the advantage of avoiding co-adaptation between the two components which reduces the risk of overfitting [47] [48]. This approach can quickly complete batch normalization. However, since the covariance matrix V is not always positive semidefinite, this method is not universal. When there is no analytical solution, another method approaches the target value by gradient descent method. As we can see from the function, there are two parameters γ and β.

$$\text{BN}(\boldsymbol{z}) = \gamma\boldsymbol{z} + \beta \tag{4.11}$$

Shift parameter β is the complex parameter with two learnable components (real mean and imaginary mean). Scale parameter γ is a 2*2 matrice with only three degrees of freedom, so there are only three learnable components. In much the same way, the covariance matrix V of $BN(z)$ normalizes the input variance to 1. During training and testing, we use the optimizer to run averages to maintain estimates of standardized statistics for complex batches.

EXAMPLE OF A COMPLEX-VALUED CNN

A simple complex-valued CNN is chosen as an example to show the process pipline of CVNNs. For the normal initial real numbers as input, performing the operations is presented within a single real-valued block [4]:

$$BN \rightarrow ReLU \rightarrow Conv \rightarrow BN \rightarrow ReLU \rightarrow Conv \tag{4.12}$$

The process flow of a complex-valued CNN is as follows:

$$ComplexBN \rightarrow CReLU \rightarrow ComplexConv \rightarrow ComplexBN \rightarrow CReLU \rightarrow ComplexConv \tag{4.13}$$

4.5. CONCLUSION

4

To classify human motions based on radar data, this chapter provided related methods and strategies. Because the radar data is different from optical images, its processing method is different, significantly step 4 and step 5 in Figure 3.8, which are also the core of the whole pipeline. In short, radar-based classification is decomposed into three main subtasks: raw radar data generation by experiments, data pre-processing, and machine learning classification. Figure 4.9 shows the full process of radar-based human motion classification with machine learning.

This chapter focused on the methods we implemented in the thesis and their differences from the regular pipeline (the red frames in Figure 4.9). Firstly, we do use not only Doppler-time but also explore the feasibility of range-Doppler and range-time. Secondly, we retain the complex data form in patterns, and take the complex as the input of neural networks. At present, the mainstream neural network frameworks, such as TensorFlow and PyTorch, is real input in default. It is important to stress that in this thesis the effort are not to just use a predefined network defined for images, but construct a complex-valued smaller network designed specifically for radar. This required the PyTorch framework and therefore more time to rewrite all the pieces of code.

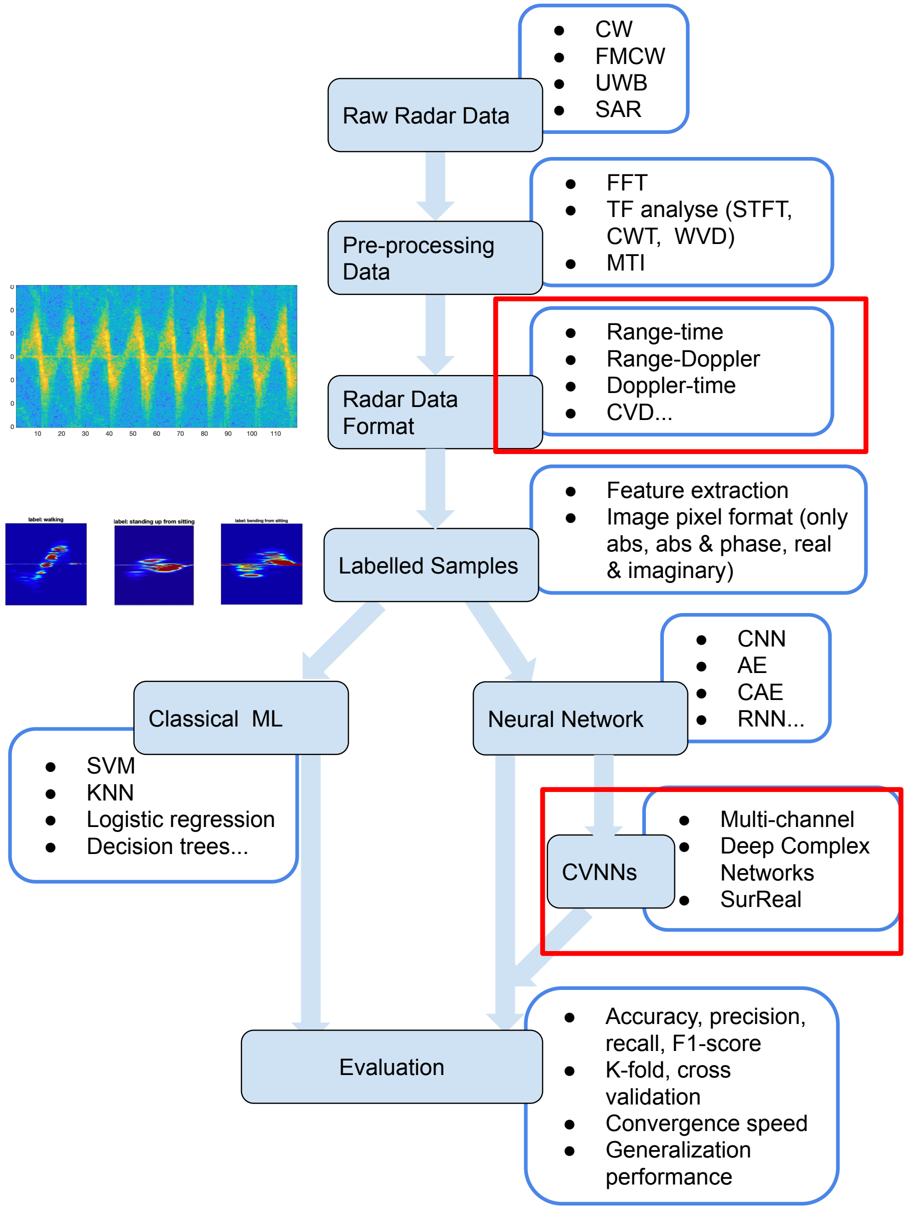


Figure 4.9: The full process of radar-based human motion classification

5

Input Data Preparation

This chapter discusses the implementation of data collection, data exploration, and data pre-processing. We will introduce the task in detail, the experimental condition and the processing techniques for radar data, and show the results of different processing methods.

5.1. Collecting Data

5.1.1. Overview

An extensive set of data and labels are needed before training the ML algorithm. Collecting radar data is time-consuming and expensive compared to camera data. Even though simulated data is feasible, the best option is still the captured realistic radar data. A few publicly available datasets online can be downloaded directly to train the ML model, for example, the CIFAR-10 dataset which includes 60000 32x32 color images in 10 classes (6000 images per class) [49]. With these labeled and cleansed datasets, the collecting and cleaning data steps can be skipped, which is very convenient. However, unlike optical images and speech, there are few public labeled radar datasets. For example, whereas the ILSVRC ImageNet database includes 1.5 million images, most work on the classification of radar micro-Doppler signatures involves just 1000–2000 measured data samples [7]. And there are much less online radar data about human movement, so developing a standard set of experiments about human motions and gaining raw radar data are necessary.

5.1.2. Experimental Setup

An experiment was implemented to obtain datasets to evaluate various ML algorithms. The radar data was acquired from the MS3 group on the 22nd floor, EEMCS building, for a current project on human movements for assisted living. The five distributed radar systems collected radar data at the MS3 lab. Here Figure 5.1 shows the radar system from node 1 in the left, node 3 in the middle, node 5 in the right. The circumference diameter of the measuring space is about 4.38m.

The five monostatic radars are pulse UWB radar with carrier frequency 4.3GHz and

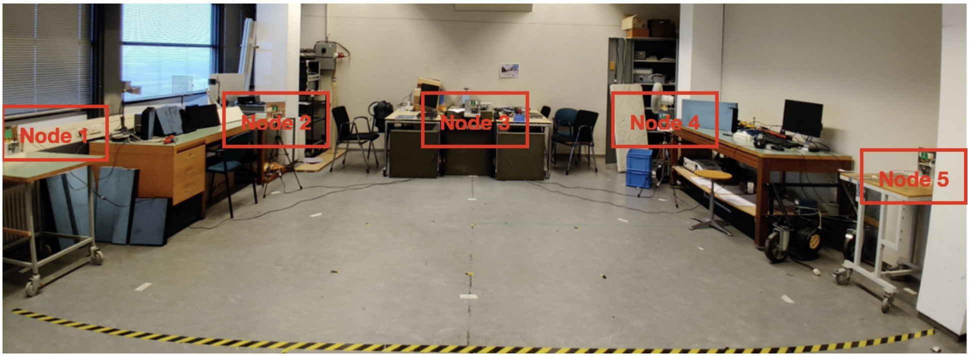


Figure 5.1: Five distributed radars at the MS3 lab

bandwidth 2.2GHz. The distance or range that the radar can detect is from 1m to 5.38m. Fifteen volunteers joined in the experiment, and each volunteer took 29 measurements consisting of 9 activity classes, so a total of 435 measurements was collected. Nine activities for training are shown below, labelled from 1 to 9 separately. One measurement is a continuous motion consisting of one or multiple activities and lasted about 2 minutes long sequence of data. Pulse repetition interval (PRI) is 8.2ms, so one measurement counted 14634 samples.

- 1: walking
- 2: stationary
- 3: sitting down
- 4: standing up from sitting
- 5: bending from sitting
- 6: bending from standing
- 7: falling from walking
- 8: standing up from ground
- 9: falling from standing

5.1.3. DATASET GENERATION

The radar system generated five 2D radar matrices (L*M) for each measurement with L in length and M in width, since there are five radars in the experiment, representing the five receiver positions for target. Length L is the range axis, also called the fast-time dimension. Width M is the pulse number axis, also called the slow-time dimension.

In the experiment, there are M = 14630 pulses with each pulse interval 0.0082s, so the duration of one measurement is 14630 * 0.0082s = 119.966s. Each pulse stands for L = 480 range bins, presenting the radar-target distance between 1m and 5.38m. As shown in Figure 5.2, the data are five 2D matrices of M pulses, each containing L range bins, so there are five range-time matrices in total. Since we use the pulse UWB radar, the raw data is range-time representation. Not all raw radar data are the same format, for example, in FMCW a first FFT is needed to extract range information from the received radar waveforms. Also note that from a data processing perspective, these data are complex numbers. The absolute values of these complex numbers are just used for image display.

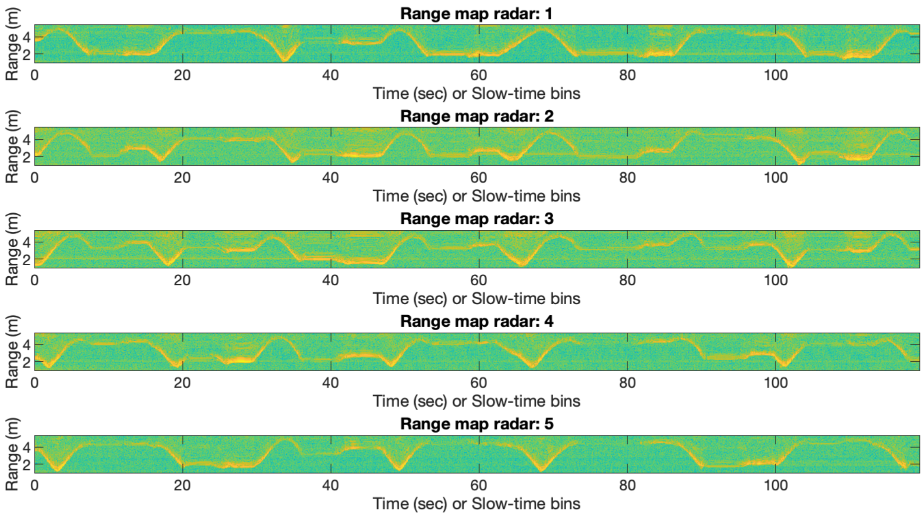


Figure 5.2: Range-time matrix of five distributed radars

During the measurement, the activity labels of participants over time were also recorded. The recording method is when the participant starts with activity 1, button 1 on the remote controller is pressed. At the time the participant alters to activity 2, button 2 is pressed. The remote controller is connected to the computer, which is synchronously recording the radar data and labels. The activities corresponding to each pulse is labeled, so the horizontal axis is the slow time dimension, and the vertical coordinate is the label. Figure 5.3 is the corresponding labels of activities in Figure 5.2, from which we can see that one measurement is a continuous motion considering one or multiple activities. In this measurement, the participant first walked, fell from walking, stood up from the ground, kept stationary, and repeated these activities. Note that label 0 presents random movements and does not need to consider.

5.2. DATA PRE-PROCESSING

The 2D radar data images generated by radar cannot be directly used as the machine learning input because it contains multiple labels. We need to cut it off to get fragments of the same label from a 2-minute continuous activity sequence. In addition, there are several radar data domain representations. We need to characterize these movements in

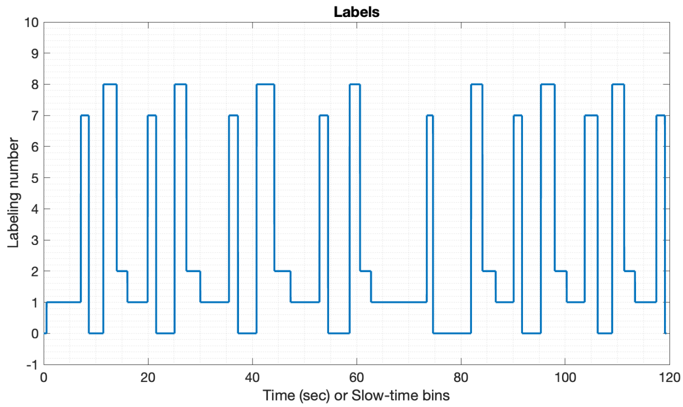


Figure 5.3: Labels of range map over time

5

different domains, not just a range-time matrix.

5.2.1. SEGMENTS EXTRACTION

Different human activities will exhibit different patterns in these frames. Continuous pulses with the same label are cut to obtain segments. Some of motions are fast (about 1s) and some last longer. The length of most segments is between 120 and 480 pulses (around 0.984s to 3.936s, PRI = 8.2ms). Those short segments with less than 90 pulses can be discarded since they are unlikely to represent the activity. There are two other questions existing below.

- The size of the input of the neural network must be the same, but because the duration of each activity is different, the size of the segments is not the same.
- The labels of some pulses are incorrect. The label is artificially marked, but each movement is short. If the button is not pressed precisely simultaneously, there may be about 0.5 seconds delay or advance.

Since the interval of movements is short, the error effect caused by wrong labels is great. There are three solutions that have been explored to solve this problem of segmentation of continuous human motion data.

- **Solution 1: Fixed cutting + Fixed padding.** The length of the segment cut is fixed at N (the labels of N consecutive pulses are the same and are not equal to 0). Paddings are added before and after the segment to solve the problem of label errors. We choose N = 120 and front padding = 80, and back padding = 120, therefore the final segment length = 120+80+120 = 320. Because many motions last more than 120 pulses, it is more reasonable to take more pulses for back padding. An example diagram 5.6 is shown where activity1 is short (120 pulses) and activity2 is long (200 pulses).
- **Solution 2: Dynamic cutting + Dynamic padding.** The n pulses in one movement are cut off entirely, and n is dynamic. The padding size is determined by n. If the

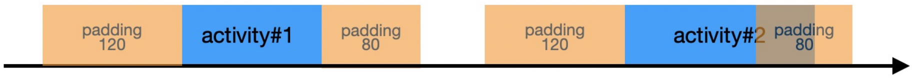


Figure 5.4: Schematic diagram of Solution 1

motion is fast, the padding size is large; If the motion is slow, the padding is small. The length of the final segment is fixed at N. The pulses cut off are in the middle of the segment when filling, so front padding = back padding = (N-n)/2. We choose N = 480 during experiments. When n > N, it means that the motion lasts a long time, and it can be cut directly without padding. Such long segments account for about 30% of all segments.

Figure 5.5: Schematic diagram of Solution 2

- **Solution 3: Dynamic cutting + Fixed padding + Rescale.** All pulses in one activity are cut off entirely, and fixed paddings are added before and after the segment (Front padding = back padding = 80). The sizes of segments after adding paddings are different, so we need to rescale all segments to one size 480*480.

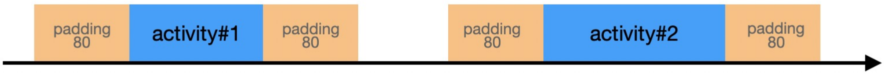
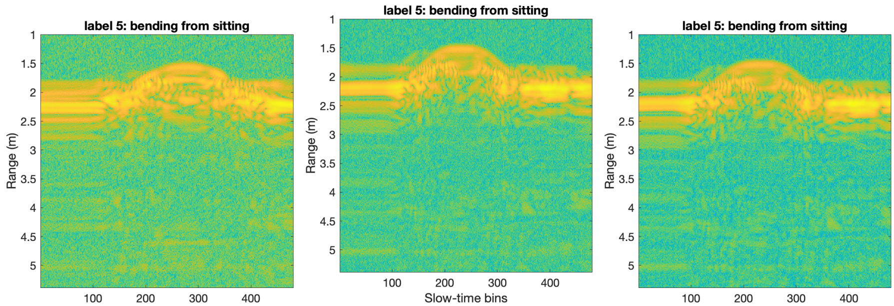


Figure 5.6: Schematic diagram of Solution 3

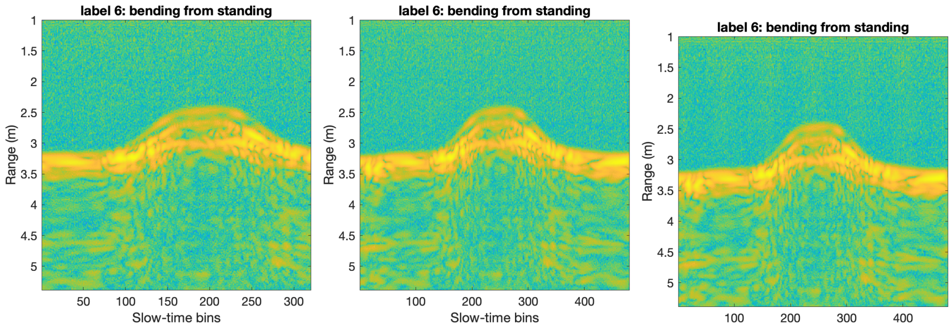
These processing methods mainly consisting of segments lengthening, patching, and scaling. These are some segments that can be compared. As can be seen from Figure 5.7 and 5.8, the motion feature in Solution 1 is not in the center of the image. As for Solution 3, because the image will be deformed due to scaling, it loses the meaning of the slow-time axis.

Only the raw data from Radar 3 is processed to generate segments by Matlab. From Figure 5.9, the datasets of the three solutions are very unbalanced, which is due to the experimental design. During the measurement, participants spent more time walking (label=1), staying stationary (label=2), bending from sitting (label=5), and bending from standing (label=6). To balance the data, if the number of segments in the same label is more than 1000 in Solution 2&3 (2000 in Solution 1), the redundant part is removed. The results are displayed in Table 5.1. Solution 1 generates the most samples. The main reason is that one activity with a long duration is divided into multiple segments, but this breaks continuity of motions. Solution 3 makes images distorted. It is reasonable to put essential features in the center of the image and retain the characteristics of the motion



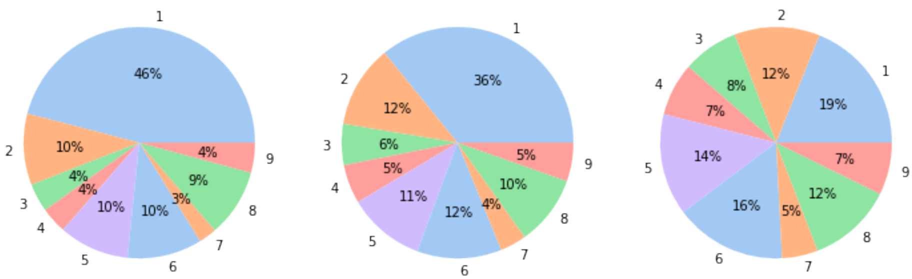
(a) Solution 1 (b) Solution 2 (c) Solution 3

Figure 5.7: Range-time segments of Label 5: bending from sitting



(a) Solution 1 (b) Solution 2 (c) Solution 3

Figure 5.8: Range-time segments of Label 6: bending from standing



(a) Solution 1 (b) Solution 2 (c) Solution 3

Figure 5.9: The distribution of the class samples labelled from 1 to 9 as extracted from range-time segments when applying the three proposed solutions to segmenting the data.

itself (no distortion), so Solution 2 is more optimal. The following processing steps are all based on Solution 2.

Table 5.1: Segments extracted, remained, and discarded by three solutions

	Solution 1	**Solution 2**	**Solution 3**
Segments extracted	30330	10944	7996
Segments remained	14344	7215	7103
Segments discarded	15986	3729	893

5.2.2. RADAR DATA FORMATS

Radar data with a single transmitter and receiver have three domains, range/distance, Doppler, and time. The range is how far the target is from the radar, Doppler is how fast the target is, and time is how the target is variable. The spectrogram is mainly used as the input data of neural networks. There are also different formats of the same data as input to neural networks.

RANGE-TIME AND RANGE-DOPPLER

The initial stage of radar signal processing is the temporal sequence of digitized received raw radar data. The form of raw radar data is a 2d range-time matrix where the column is each radar pulse (range bins, the distance of a possible target), and the row is the temporal sequence of these pulses. Examples, Figure 5.8(b) and 5.9(b), are the typical 2D fast-time vs slow-time data structure on which radar signal processing is applied. This range–time-intensity matrix is seen as an image whose intensity of pixels is a complex number, called a complex-valued image. Therefore segments extracted by Solution 2 can be the input with the size of 480*480 directly.

To extract speed information of the target, we calculate the fast Fourier transform across all the rows of the previous matrix and obtain a new matrix, called the range-Doppler matrix with the size still 480*480. The fast-time (column) axis is associated with range, and the slow-time axis (row) is now converted to Doppler or velocities.

STFT FOR DOPPLER-TIME

FFT can be applied to characterize the target's velocity through its Doppler effect. However, the overall Doppler is caused by the macro-movement of the person and has nothing about how the body and its parts are moving over time. Selecting a row of raw data collected from every measurement takes STFT to obtain Doppler–time pattern. In order to find the strongest range bin for each measurement (480 rows), we firstly sum all pixels in one row and then find the row with the largest intensity, indicating that the target is mainly active at this distance.

When implementing STFT, we choose the Hann window. Its function expression and shape are as below.

$$w(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \tag{5.1}$$

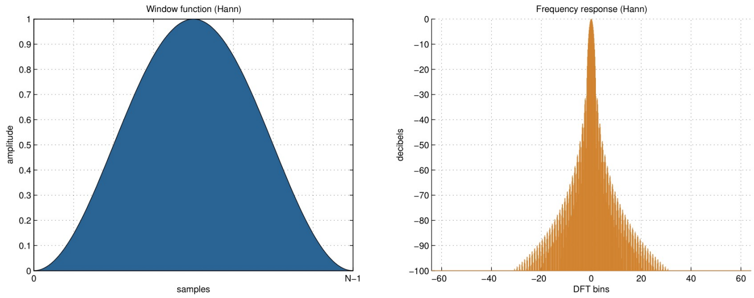


Figure 5.10: Hann window function and frequency response [6]

Depending on the duration and motion task, the length of the window is fixed at 244. Overlap between two consecutive windows has a typical default value (50%). However, we took 240 since we want to catch as many micro-motions as possible. The number of discrete Fourier transform (DFT) points is 240, referred to as frequency solution. Hence, the size of the Doppler-time matrix after extracting frames (Solution 2) is 240*120. It is worth mentioning that the Hann window [0 N-1] has the largest value at the middle point and small values at both ends. The obtained frequency distribution is closest to the frequency distribution of the time point (N-1)/2. Since the Hann window function in MATLAB will default to the time point at the left end, there will be about 1 second prior. Many motions only last 1 second to 4 seconds, thus the error has a significant impact and needs to be corrected. After STFT, the short-time axis should shift 122 pulses to the right and generate segments according to Solution 2.

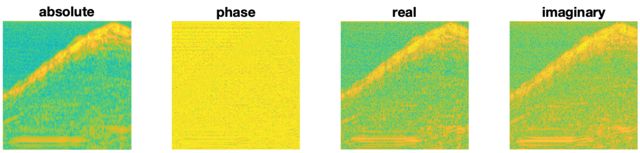
RANGE-SPECTRUM-TIME AND PSEUDO-DOPPLER-TIME

Doppler-time only uses one range of data, but we can also use all range data. The range axis is fast-time samples from individual radar pulse. We can calculate FFT across all the columns to obtain the spectrum of range. This operation can also be understood as the feature obtained by the spatial Fourier transform. FFT on each column obtains a new matrix called the range-spectrum-time matrix, which reflects the change of range frequency over time. Then the row of the most strong frequency bin is selected so that the range information is gathered for the next STFT step. The exact position of the strongest range bin does not matter. After STFT, the pattern is called pseudo-Doppler-time matrix.

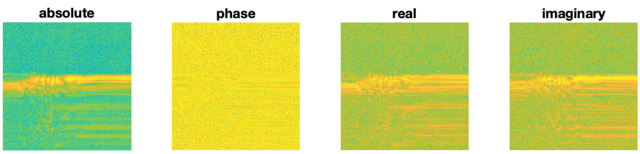
5.3. RESULTS AND DISCUSSION

Further processing after generating the patterns typically consists of using machine learning to teach an algorithm how to classify patterns related to different activities automatically. Patterns represent different meanings and have different image sizes. All data formats are rescaled into the same matrix size, 240x120. The first advantage is to reduce the image resolution, the time and computational cost of neural network training. The second advantage is that the unified image size is convenient for determining the hyperparameters of the network model and comparing different patterns.

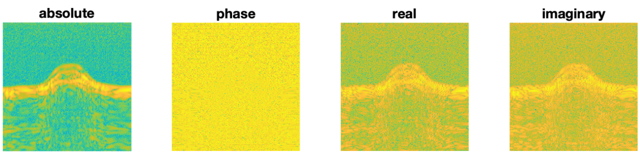
The whole preprocessing process is carried out on MATLAB. These are some example samples after preprocessing, including range-time, range-Doppler, Doppler-time, range-spectrum-time, and pseudo-Doppler-time below. The value of these data representations is complex. People have fed these representations into the neural networks in the radar domain, but only absolute values and no phase. In the thesis project, the phase will be fed too.



(a) Label 1: walking



(b) Label 4: standing up from sitting



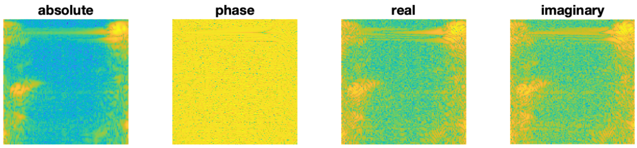
(c) Label 6: bending from standing

Figure 5.11: Range-time frames with Label 1, 4 and 6

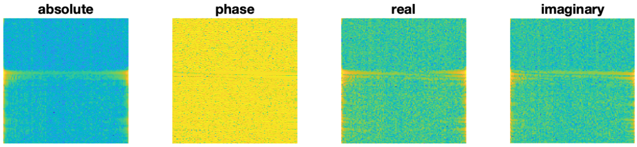
There are samples of three activities in the range-time format in Figure 5.11. Subfigure 5.11(a) illustrates walking, showing that the person moved toward the radar and turned back, and the range-time image is zigzag shape. Standing up from sitting indicates that the distance between the target and the radar is basically unchanged, while the bending from standing has a fluctuation. The real and imaginary parts are similar to the pattern and texture of amplitude.

For range-Doppler in Figure 5.12, there are both positive and negative Doppler value, since the target moving toward the radar (positive Doppler) and away from the radar (negative Doppler). It also indicates the volunteer's moving position. The phases of the three activities are also different textures which is meaningful.

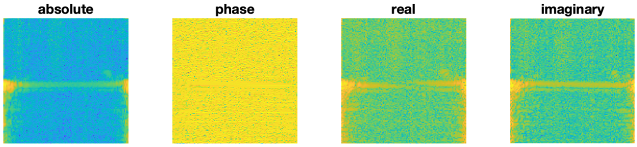
Doppler-time by STFT has more intense signature (yellow colour) due to the movement of the torso and main body, referred as micro-Doppler signature in Figure 5.13. The absolute value of Doppler-time is spectrogram. Spectrum image analysis is that different intensities in the frequency domain represent the small movements of human limbs. Each frequency intensity of TF distribution changes over time. It seems that there is no difference between different activities phases.



(a) Label 1: walking



(b) Label 4: standing up from sitting

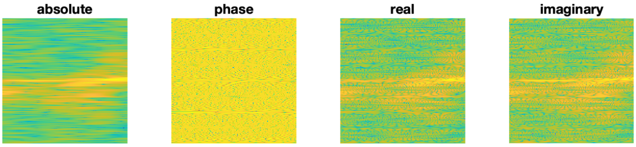


(c) Label 6: bending from standing

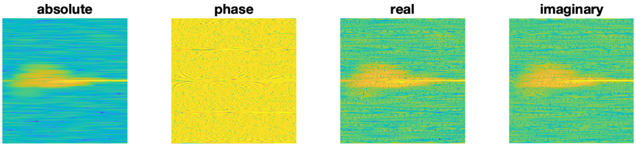
Figure 5.12: Range-Doppler frames with Label 1, 4 and 6

5

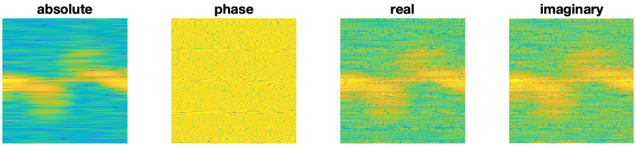
Figure 5.14 of range-spectrum-time reveals the spatial frequency intensity after Fourier transform, indicating that the spatial frequency of each activity is concentrated in about the 110th row in [1 480]. The intensity of frequency still varies with time.



(a) Label 1: walking

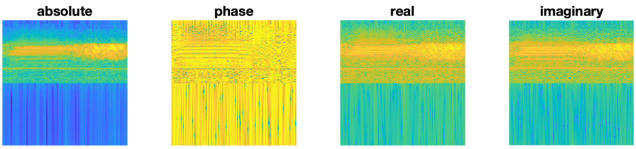


(b) Label 4: standing up from sitting

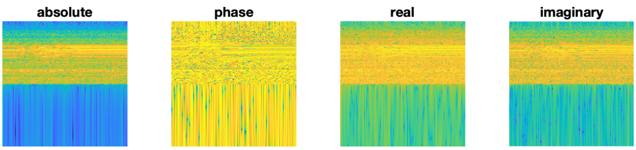


(c) Label 6: bending from standing

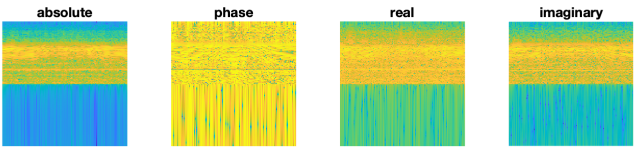
Figure 5.13: Doppler-time frames with Label 1, 4 and 6



(a) Label 1: walking



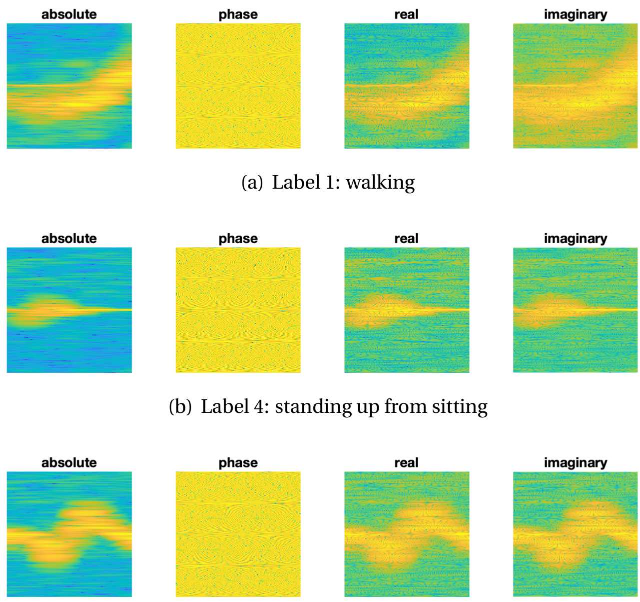
(b) Label 4: standing up from sitting



(c) Label 6: bending from standing

Figure 5.14: Range-spectrum-time frames with Label 1, 4 and 6

After the 110th row is taken as a 1D vector, the intensity of this characteristic spatial frequency changes with time, and then TF analysis is carried out. Pseudo-Doppler-time images in Figure 5.15 is similar to Doppler-time in that it performs TF analysis by range information. However, this operation loses the physical meaning of Doppler-time. As a distinction, it is called pseudo-Doppler-time. Different activities' phases are almost the same, and the phase seems not to help classify.



(a) Label 1: walking

(b) Label 4: standing up from sitting

(c) Label 6: bending from standing

Figure 5.15: Pseudo-Doppler-time frames with Label 1, 4 and 6

5.4. CONCLUSION

Overall, this chapter focused on implementation, generating five datasets from different formats. At first, the experimental setup was presented in detail, such as the radar type, radar frequency, and motion classes. Then the segments were captured from raw measurement data by three solutions. At last, five radar data pre-processing approaches were implemented to generate dataset samples, including FFT and STFT. Samples generated from MATLAB are saved as NPY files to classify human motions on Python for subsequent processing.

6
Proposed Classification Algorithm

Five datasets were obtained after pre-processing the experimental data, based on various data representations (Doppler-time, range-Doppler, range-time, range-spectrum-time, and pseudo-Doppler-time). These datasets are used in conjunction with the developed complex-valued network architectures using metrics such as accuracy and generalization performance. It is shown that CVNNs do improve the CNN performance on certain datasets.

6.1. CNN Models

6.1.1. Plain ConvNet Models

A basic CNN architecture is demonstrated in the flow chart 6.1, consisting of CONV layer, batch normalization, ReLU layer, pooling layer, flattening, fully-connected layer, and sigmoid function. This small network is explicitly designed for radar called "shallow ConvNet" in this thesis. The first four layers (CONV, BN, ReLU, Pool) together are regarded as one building block.

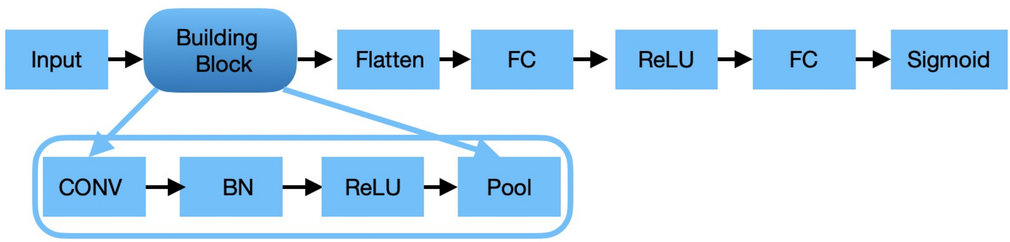


Figure 6.1: Shallow ConvNet: Only one CONV layer in the network

More building blocks stacked before flattening makes the network complex with more parameters.Generally speaking, the small number of layers indicates that the network is "shallow" while many layers mean "deep". It is hard to strictly define "shallow" or "deep" strictly. We choose five building blocks called "deep ConvNet" and illustrated in Figure 6.2.

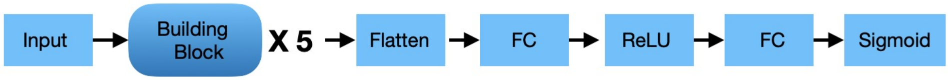


Figure 6.2: Deep ConvNet: 5 building blocks stacked for deep network

Input is a 2d matrix, and the hyperparameters of layers in blocks are in Table 6.1, where the out channels are the number of filters. Shallow ConvNet corresponds to Block 1. The output layer is the sigmoid function.

Table 6.1: Hyperparameter settings in ConvNet

	Block 1	**Block 2**	**Block 3**	**Block 4**	**Block 5**
CONV: out channels	64	64	128	128	128
CONV: kernel size	(4,4)	(3,3)	(3,3)	(2,2)	(2,2)
CONV: padding	1	1	1	1	1
CONV: stride	(2,2)	(1,1)	(1,1)	(1,1)	(1,1)
MaxPool: kernel size	(2,2)	(2,2)	(2,2)	(2,2)	(2,2
MaxPool: stride	(2,2)	(2,2)	(2,2)	(2,1)	(1,1)

6

6.1.2. RESNET MODEL

Shallow and deep ConvNets are the plain networks that may drop off in the effectiveness of additional layers due to the vanishing gradient problem. ResNet has a more complicated structure to solve the degradation problem caused by activation functions. ResNet has the residual units to "identity shortcut connections". More specifically, there is a short-circuit connection every two CONV layers, which forms residual learning. In addition, an important design principle of ResNet is that when the feature map size is reduced by half, the number of feature maps is doubled, which maintains the complexity of the network layer.

There are predefined ResNet models under PyTorch framework. However, they expect the input images normalized in the same way, i.e., mini-batches of 3-channel RGB images of shape (3 x H x W), where H and W are expected to be at least 224. Our radar data is complex-valued, the image size is 240 * 120, and the number of channels is not 3. In short, it is different from general optical images, so it does not meet the requirements. We built the ResNet model suitable for the radar data, mainly referring to the structure and parameters of **ResNet-18** [4]. The res block is shown in Figure 6.3. One res block includes two CONV layers whose hyper-parameters are totally same. The flow chart in Figure inllustrates all layers of ResNet. The basic block is as the same as the building block of the plain network, consisting CONV layer, BN layer, ReLU and MaxPool layer.

Then the 8 res blocks are stacked, Res11, Res12, Res21, Res22, Res31, Res32, Res41 and Res42.

Figure 6.3: One res block: CONV+BN+ReLU+CONV+BN

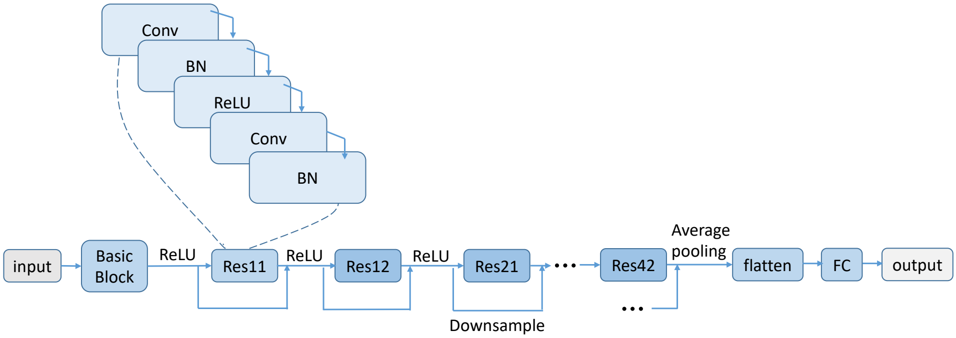


Figure 6.4: ResNet architecture: network stacked by one basic block and 8 res blocks

Two res blocks are combined, and every two res blocks may perform downsampling with a stride of 2 (Res21, Res31 and Res41). When the dimensions are inconsistent (the corresponding dimension is doubled), they cannot be added directly. There are two strategies for downsampling: one is the pooling layer with stride 2, and the other is the projection shortcut, that is, the convolution layer with filter size 1*1 and stride 2. The projection shortcut will increase parameters and computation. We chose the latter one, so there are three more CONV layers in the architecture. Besides, the final pooling layer applied is a 2D adaptive average pooling. All the hyper-parameters of the model are displayed in Table 6.2 below. The values in Res block is the parameters of inside CONV layer. Conv2, 3 and 4 are for downsampling.

6.2. THREE MAIN APPROACHES OF CVNNS

6.2.1. OVERVIEW

In this section, the complex-valued networks are built based on regular neural network models. The architecture of models almost remains except the input is complex numbers. Therefore slight modification exists. There are six models to compare in the thesis: shallow ConvNet, complex-valued shallow ConvNet, deep ConvNet, complex-valued deep ConvNet, ResNet, and complex-valued ResNet. At present, mainstream machine and deep learning frameworks do not support complex-valued input due to the absence of the corresponding complex-valued building blocks. In this section, three approaches of CVNNs to classify human motions are achieved.

Table 6.2: Hyperparameter setting in ResNet

Blocks	Out channels	Kernel size	Stride	Padding
Conv1	64	(7,7)	(2,2)	3
MaxPool	-	(3,3)	(2,2)	1
Res11/Res12	64	(3,3)	(1,1)	1
Conv2	128	(1,1)	(2,2)	-
Res21/Res22	128	(3,3)	(2,2) / (1,1)	1
Conv3	256	(1,1)	(2,2)	-
Res31/Res32	256	(3,3)	(2,2) / (1,1)	1
Conv4	512	(1,1)	(2,2)	-
Res41/Res42	512	(3,3)	(2,2) / (1,1)	1

6.2.2. MULTI-CHANNEL ARCHITECTURE

This main idea is the complex-valued data represented by two-channel real-valued data. Inspired by 3-channel RGB optical image, the complex-valued radar date is regarded as 2-channel image (the magnitude and phase, or the real and imaginary parts). Before that the absolute value of radar image is default as the input, so the input is 1-channel image. No matter how many channels the input holds, it is real-valued representation. Therefore, it can be trained directly with the previous ordinary CNN models. There is nothing to modify the model itself.

6

6.2.3. DEEP COMPLEX NETWORKS

This approach is to exploit the complex building blocks, such as complex convolution and complex batch normalization. Paper [5] laid the theoretical foundation for complex blocks and tried to implement them on the TensorFlow framework. They applied this technique to the image and audio dataset. We trained the radar dataset. More importantly, due to the version issue of TensorFlow, the codes from Paper [5] can not run any longer. The main reason is that some essential embedded functions have been sifted out with TensorFlow upgrade, and there are no replaceable functions. There is also a compatibility problem between TensorFlow and Keras. Therefore, we reproduced the core part of this paper under PyTorch framework. We built blocks and layers suitable for complex numbers, including complex convolution, max pooling, adaptive average pooling, CReLU, and complex batch normalization. In this way, the regular model replaces layers to the corresponding complex layers before the flattening. There are several types of complex ReLU. Experiments on various tasks proved the CReLU function to be vastly the most effective 8768161, so we all take CReLU.. It is worth mentioning that the format of the input complex must be real plus imaginary parts. Data can not be in absolute values and phases form because the convolution layer performs complex arithmetic based on the theory of real and imaginary.

The flatten step follows the multi-channel approach, flattening the real and imaginary parts as different features into a 1D vector, followed by a fully-connected layer that projects the data into the real domain.

6.2.4. SURREAL

The SurReal approach is adding several wFM layers and tReLU layers as bottom layer of the baseline CNN models to process complex-valued data. It is theoretically proven equivariance and invariance properties [25]. It trains the network on magnitude/phase tensors like the other networks, so the input must be in magnitude/phase form.

6.3. TRAINING PARAMETERS

During the experiment, the whole dataset was splitted into trainning data (80%) and testing data (20%). No matter which kind of radar format, the number of samples is around 7000. The primary performance metric is accuracy. Cross entropy loss is represented as a cost value or loss function to minimize. Adam optimizer is adopted, the batch size is 32, and the epochs are from 40 to 60, depending on models. The learning rate is dynamic from 1e-3 to 1e-6 with the specific scheme also according to models. Learning rate and epoches are crucial and these hyper-parameters need to be tuned every session.

For ML, k-fold validation is a smart strategy so that the difference between the accuracy of training and testing is little. However, there are so many situations (at least 12 situations for each dataset) to handle. We have only utilized simple train/test mode and one performance metric so far.

6.4. RESULTS

6.4.1. CVNNS ON DIFFERENT DATA DOMAIN REPRESENTATIONS

After training and testing, each proposed approach of CVNNs and real-valued CNNs is evaluated on five radar-based data. "Abs only" means that the input is absolute of complex numbers thus the models are corresponding real-valued CNN models for comparison. The multi-channel approach has two formats. The final train and test accuracy may differ every training session since some models are unstable and sensitive to data splitting (dataset was split randomly every session). Accuracy shown on tables are the highest accuracy among several training sessions. The performance is analysed to determine which pre-processing task is more suitable for CVNNs.

PSEUDO-DOPPLER-TIME

The first situation is pseudo-Doppler-time, a type of micro-Doppler representation generated by STFT. The results of pseudo-Doppler-time dataset are shown in Table 6.3.

Table 6.3: Accuracy (train; test) on pseudo-Doppler-time dataset

Models	Shallow ConvNet	Deep ConvNet	ResNet
Abs only (real-valued network)	85.8%; **76.0%**	97.5%; 86.2%	100.0%; 87.0%
Multi-channel: Abs and phase	81.5%; 74.6%	95.7%; 84.7%	100.0%; 86.5%
Multi-channel: Real and imaginary	84.2%; 74.2%	95.4%; 84.6%	100.0%; 85.6%
DCN (Deep complex network)	45.5%; 44.5%	96.4%; **86.6%**	100.0%; **87.5%**
SurReal	13.8%; 13.8%	13.8%; 13.8%	93.1%: 79.2%

From Table 6.3, for shallow CNN model, the traditional real-valued one is the best, two multi-channel CVNNs are the second, the results of DCN and SurReal are far less than the real-valued CNN. For deep ConvNet and ResNet models, the accuracy of DCN is slightly higher than the real-valued CNN (around 87%). The result of SurReal is a little worse than other CVNN approaches. Pseudo-Doppler-time is one of the most sophisticated pre-processing techniques that excavate features to the greatest extent and, finally, has the relatively high accuracy, up to 87.5%. However, CVNNs do not improve the accuracy of the model apparently. The reason may be that the phase of the complex obtained by STFT has little meaning in itself. In other words, this phase feature cannot help target classification. As for the training speed of CVNN, multi-channel is the fastest while DCN is medium. SurReal is slow, and extremely slow when training plain CNN models. SurReal is only suitable for ResNet architecture, so we do not consider SurReal in the following experiments.

RANGE-SPECTRUM-TIME
Table 6.4 shows the accuracy of range-spectrum-time data. The multi-channel approach with input in absolute and phase format has the highest accuracy (53.2%) on the shallow ConvNet. For the deep models, including deep ConvNet and ResNet, the accuracies of DCN are both around 80%. In this case, the performance of the DCN and multi-channel approach is better than the real-valued CNNs obviously.

Table 6.4: Accuracy (train; test) on range-spectrum-time dataset

Models	**Shallow ConvNet**	**Deep ConvNet**	**ResNet**
Abs only (real-valued network)	13.9%; 13.9%	91.8%; 77.5%	100.0%; 76.3%
Multi-channel: Abs and phase	62.3%; **52.3%**	90.9%; 79.3%	100.0%; 76.5%
Multi-channel: Real and imaginary	13.9%; 13.9%	93.9%; 79.9%	100.0%; **81.3%**
DCN (Deep complex network)	13.9%; 13.9%	94.7%; **81.1%**	100.0%; 80.0%

RANGE-DOPPLER
When the input is range-Doppler, Table 6.6 exhibits the highest accuracy of range-Doppler 74.9% (ResNet + multi-channel with real and imaginary). The CVNNs do improve the accuracy on each CNN model (3.1%, 7.2%, 8.9% respectively). The accuracy of range-Doppler is obvious lower than that of pseudo-Doppler-time. The range-Doppler is better than range-spectrum-time only on the shallow neural network.

RANGE-TIME
Range-time format is the raw data, making each activity cut down and the label into samples, and training the models directly. The best test result is 92.6% accuracy shown in Figure 6.6. Range-time is suitable, at least for human motion classification. Besides, the CVNNs helps improve a little accuracy (1.5%) on ResNet network.

DOPPLER-TIME
Doppler-time images do not perform well with only around 60% accuracy, whose accuracy is the lowest among all radar data representations. The reason may be that only

Table 6.5: Accuracy (train; test) on range-Doppler dataset

Models	**Shallow ConvNet**	**Deep ConvNet**	**ResNet**
Abs only (real-valued network)	80.8%; 58.4%	89.6%; 64.7%	100.0%; 66.0%
Multi-channel: Abs and phase	84.0%; 59.2%	89.6%; 66.5%	100.0%; 66.9%
Multi-channel: Real and imaginary	84.3%; **61.5%**	96.1%; **71.9%**	100.0%; **74.9%**
DCN (Deep complex network)	30.7%; 28.0%	96.5%; **71.9%**	100.0%; 73.3%

Table 6.6: Accuracy (train; test) on ranger-time dataset

Models	**Shallow ConvNet**	**Deep ConvNet**	**ResNet**
Abs only (real-valued network)	90.2%;**78.4%**	98.0%; **90.4%**	100.0%; 91.1%
Multi-channel: Abs and phase	67.0%; 60.3%	97.4%; **90.4%**	100.0%; 91.5%
Multi-channel: Real and imaginary	80.3%; 67.4%	98.0%; 89.1%	100.0%;**92.6%**
DCN (Deep complex network)	37.7%; 36.7%	97.9%; 89.3%	100.0%; 91.5%

one range of data is used. The poor result of getting just one single range bin to take the spectrogram may come from the fact that the target is spread across many range bins. Operating in that way, only a small part of the body is captured. As for CVNNs, only multi-channel in magnitude and phase helps increase accuracy on the deep ConvNet.

6

Table 6.7: Accuracy (train; test) on Doppler-time dataset

Models	**Shallow ConvNet**	**Deep ConvNet**	**ResNet**
Abs only (real-valued network)	72.7%;**54.6%**	78.3%; 60.9%	100.0%; **60.7%**
Multi-channel: Abs and phase	58.6%; 45.6;%	77.6%; **62.0%**	100.0%; **60.7%**
Multi-channel: Real and imaginary	67.4%; 44.4%	75.3%; 58.9%	99.9%;60.6%
DCN (Deep complex network)	28.4%; 27.4%	69.0%; 57.0%	100.0%; 58.9%

DISCUSSION

Summary across all tables and some discussions are shown below.

- As for five radar data formats, range-time and pseudo-Doppler-time have the highest accuracy (92.6% and 87.5%, respectively), followed by range-spectrum-time and range-Doppler (81.3% and 72.3%, respectively). Doppler-time has the worst performance with only 62% accuracy.
- CVNNs improve the accuracy of range-spectrum-time and range-Doppler (4% to 7%) on deep ConvNet and ResNet). CVNNs are of little help for the other three formats since the improved accuracy (1%) is negligible.
- For the shallow CNN model, CVNNs have little advantage. For the deep CNN (deep plain model and ResNet), except SurReal, the effect of CVNNs is postive compared

to that of real-valued CNN.

- It is worth mentioning that the DCN approach is often much less accurate in the shallow model than the real-valued, but on deep models, the performance is satisfying.
- The accuracy of multi-channel (phase & abs) is close to real-valued CNN, especially on deep ConvNet and ResNet, which can be regarded as a special case of the latter.

6.4.2. LEAVE ONE PERSON OUT MODE

The pseudo-Doppler-time and range-time datasets are selected to validate the best network in leave one person out mode, because these two formats have better performance on the last subsection. The strategy is that the data of one person is randomly selected as the test data while others are training data. This validation strategy can better evaluate the activity classification performance of the model in the actual situation. It is easy to find in Table 6.8 and Table 6.9 that the test accuracy increase about 3%-7% higher than before. The reason lies in the imbalance of data. Previously, some samples have been deleted in order to make the data more balanced. The current test dataset is from a person without any deletion. Thus its accuracy becomes higher, even higher than the training set in some cases. The highest one is 98.8% on ResNet model by the range-time dataset.

6

Table 6.8: Accuracy (train; test) on pseudo-Doppler-time dataset in leave one person out mode

Accuracy	Shallow ConvNet	Deep ConvNet	ResNet
Abs only (real-valued network)	86.0%;83.3%	95.3%; **90.3%**	100.0%; **90.3%**
Multi-channel: Abs and phase	86.1%; **85.1%**	95.6%; 89.2%	100.0%; 89.6%
Multi-channel: Real and imaginary	91.2%; 84.3%	95.7%; 87.7%	100.0%;89.9%
DCN (Deep complex networks)	52.7%; 62.0%	96.1%; 89.5%	100.0%; 89.5%

Table 6.9: Accuracy (train; test) on range-time dataset in leave one person out mode

Accuracy	Shallow ConvNet	Deep ConvNet	ResNet
Abs only (real-valued network)	83.6%;87.3%	98.3%; 98.3%	100.0%; **98.8%**
Multi-channel: Abs and phase	85.8%; **89.8%**	98.6%; 98.4%	100.0%; 98.7%
Multi-channel: Real and imaginary	82.8%; 81.1%	98.4%; **98.5%**	100.0%;98.3%
DCN (Deep complex networks)	14.2%; 33.1%	91.1%; 93.9%	100.0%; 98.3%

6.4.3. CONFUSION MATRIX

Since ResNet architecture by multi-channel (real imaginary) on range-time dataset has the highest accuracy, its test confusion matrix is illustrated in Figure 6.5. We can see that

Class 1,2,5,6,8 account for most samples, representing unbalance. Class 7 accounts for only 81 samples with 10 wrong classified, whose recall and precision is relatively low. Class 1 holds highest recall. Anyway, this classifier works well.

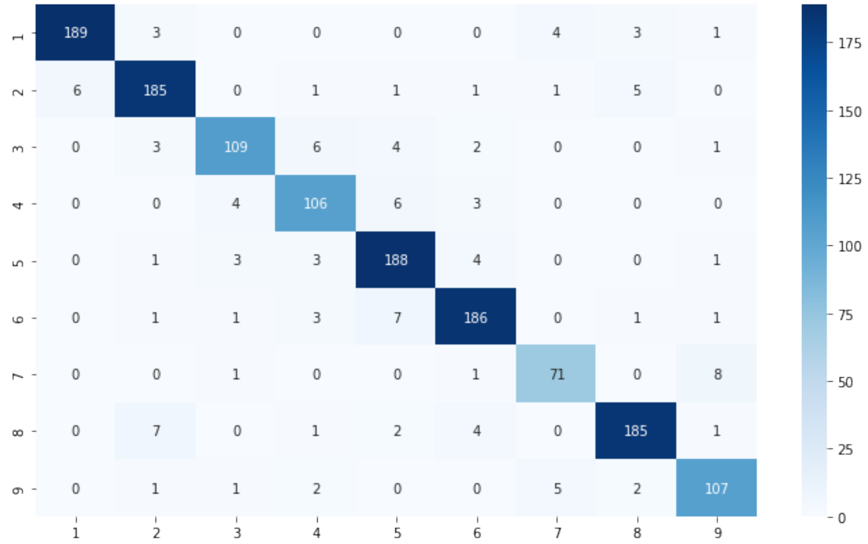


Figure 6.5: Confusion matrix of ResNet by multi-channel (real imaginary) on range-time dataset

6.4.4. COMPARISON OF MODEL COMPLEXITY

To explore the model complexity, the chosen dataset is the pseudo-Doppler-time and range-time, whose accuracy is high, and the basic model is the plain CNN. DCN technique is selected and the counterpart is real-valued plain CNN. When n = 1, the complexity of the model is low (shallow model) and DCN performs bad. The performance of real-valued CNN is better than DCN, which means that DCN makes no sense for very shallow network. From n = 2, the model begins to overfit, and the gap between DCN and real-valued CNN becomes narrow. The highest accuracy of the model is 86.6% on pseudo-Doppler-time dataset and 89.7% on range-time dataset with five building blocks CNN.

As can be observed from Table 6.10 and Table 6.11, the five-block network is complex enough for radar data. The neural network with five blocks is not real "deep" since deep CNN generally has hundreds of layers in image domain [4]. This tables prove that the relatively shallow neural network is more suitable for radar data.

6.4.5. GENERALIZATION PERFORMANCE

Generalization ability presents how the model performs in limited data. The model with high accuracy in small dataset means high generalization performance. Generalization performance is important metrics because radar data is rare and hard to collect. To evaluate generalization ability of CVNNs, the pseudo-Doppler-time and range-time datasets

Table 6.10: Accuracy (train; test) of the plain ConvNet on pseudo-Doppler-time dataset

n building blocks	Abs only(real-valued network)	DCN
n = 1	83.5%; **78.7%**	46.2%;46.4%
n = 2	95.8%; **81.5%**	87.2%; 76.5%
n = 3	95.5%; 82.5%	94.6%; **82.6%**
n = 4	95.5%; **85.5%**	96.6%; 84.3%
n = 5	96.5%; 86.2%	97.4%; **86.6%**
n = 6	96.3%; **84.6%**	97.1%; 83.9%
n = 7	95.3%; **83.7%**	94.9%; 82.7%

Table 6.11: Accuracy (train; test) of the plain ConvNet with different complexities on range-time dataset

n building blocks	Abs only(real-valued network)	DCN
n = 1	85.3%; **74.6%**	25.0%;23.4%
n = 2	97.9%; **86.8%**	97.1%; 81.8%
n = 3	97.6%; **88.1%**	97.9%; 86.3%
n = 4	97.4%; **88.6%**	98.1%; 88.1%
n = 5	98.0%; 89.6%	98.2%; **89.7%**
n = 6	97.4%%; 88.9%	98.1%; **89.1%**
n = 7	97.1%%; **88.6%**	97.8%; 87.9%

are chosen as input and the model is deep ConvNet. The values in Table 6.12 and 6.13 are test accuracy. CVNNs overpower real networks when presented with a large yet complicated dataset on pseudo-Doppler-time. As for range-time, the accuracy of the complex-valued CNN models is as almost the same as the CNN models under different training samples.

Table 6.12: Test accuracy on the pseudo-Doppler-time format; dataset size is changeable for evaluating generalization performance

Number of training samples	**250**	**500**	**1000**	**2000**	**4000**	**5772**
Abs only (real-valued network)	**60.7%**	68.9%	74.2%	**79.7%**;	81.8%	86.2%
Multi-channel: Abs and phase	58.7%	69.0%	**75.4%**	78.9%;	**83.8%**	84.7%
Multi-channel: Real and imaginary	58.4%	**69.4%**	75.3%	78.8%;	81.8%	84.6%
DCN	59.3%	64.8%	71.2%	72.9%;	82.3%	**86.6%**

6.5. CONCLUSION AND DISCUSSION

This chapter constructed three neural networks, and three CVNN approaches were implemented on five different format datasets. The highest classification accuracy determined

Table 6.13: Test accuracy on the range-time format; dataset size is changeable for evaluating generalization performance

Number of training samples	**250**	**500**	**1000**	**2000**	**4000**	**5772**
Abs only (real-valued network)	54.3%	**66.5%**	**79.3%**	**84.3%**;	**87.6%**	90.2%
Multi-channel: Abs and phase	56.0%	64.2%	79.0%	84.1%;	87.5%	**91.1%**
Multi-channel: Real and imaginary	49.7%	65.9%	73.6%	82.7%;	86.8%	89.8%
DCN	**59.5%**	63.4%	78.0%	82.7%;	87.7%	88.4%

by a series of experiments. There are still a few aspects to discuss for a broader analysis and understanding of the results presented in this chapter.

1. **Overfitting problem**: The results section shows that deep ConvNet and ResNet are overfitting, and shallow ConvNet is underfitting in some cases. The problem of ResNet is severe since the training accuracy is 100% all the time. Figure 6.6 represents the accuracy of three cases, real-valued ResNet (abs), DCN+ResNet and multi-channel (real&imaginary) on the range-time format. The gap between train and test is a sign of overfitting. It cannot generalize because the classifier is too complex. The optimal model is that both training and validation costs converge to an acceptable performance value. To solve the overfitting probelm, one way is the regularization method, and the other is to try a simpler model.

6

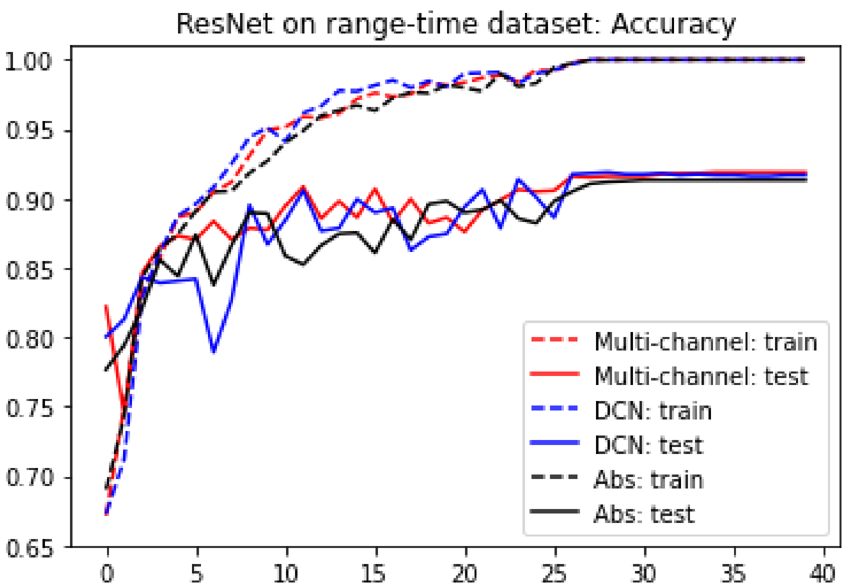


Figure 6.6: Accuracy curves of ResNet model on range-time dataset

2. **Limited data**: One assessment approach for overfitting is increasing the training set size. The gap between training and testing accuracy may be due to the small samples rather than overfitting. With more data available, the performance of the model improves. However, it is hard to get more radar data sometimes. One solution is transfer learning, where the model is pre-trained by an available dataset from other domains, such as the ImageNet dataset. Transfer learning is practical, making test accuracy increase, especially for CNN [50]. The more generalized the data is, the better the performance of transfer learning is. Discarded data in Chapter 5 can also be reused for transfer learning. Firstly, train the neural network by discarded segments, then reuse the lower layers of this network. After that, unfroze one or more of the top hidden layers. Finally, the learning rate is reduced, and all layers are unfrozen for training on the dataset to avoid wrecking their fine-tuned weights.

3. **Hyper-parameters tuning**: Supervised fine-tuning on measured data is very important. CVNNs may converge fast, and it arrives at local best rather than global best and is stuck in it. Hyper-parameters need to be adjusted to fit models. Some hyper-parameters, such as learning rate and epochs, directly decide the final result. One strategy is grid search. Fine hyper-parameters tuning cost lots of time.

4. **DCN training cost**: Intrinsicly, a complex network will have twice as many parameters as its real counterpart. It also will cost a four-times calculation. Training DCN is more expensive and slow. Besides, more parameters mean that DCN may also suffer from overfitting when a sufficient amount of diversified data is not met.

5. **Flatten layer in DCN**: Complex numbers cannot be calculated reasonably in the dense layer at present. It embeds complex-valued data within a Euclidean space that does not respect the intrinsic geometry of the space of complex numbers. So far, there is no layer to calculate the 2-norm (complex magnitude) of several channels in neural networks. 2-norm is a non-linear operation, and there is no ready-made layer to modify. Therefore, the complex flatten layer is an open question.

7

Conclusion and Future Work

7.1. Discussion and Conclusion

In this thesis, the human-motion classification based on several radar data formats has been explored. Many radar data processing techniques refer to image or speech signal processing according to the characteristics of radar data, but there are still outstanding questions to investigate as to whether using complex data can help with classification performances. Five radar data representations were exploited from pulse radar, including range-time, range-Doppler, Doppler-time, range-spectrum-time, and pseudo-Doppler-time. Neural networks can improve the accuracy of the human-motion classification and omits the step of feature extraction. Therefore, based on these five datasets and CNN models, three CVNN approaches were handled. Now, some sub-questions in Chapter 2 are answered below according to the results.

- Range-Doppler and range-spectrum-time formats are suitable for CVNNs with 4%-7% improvement on accuracy compared with their conventional real-valued network counterparts. As for the other three formats, CVNNs have almost no impact on accuracy.

- Generally speaking, relatively deep CNNs gained higher accuracy through CVNNs. Shallow CNNs keep constant, or provide even worse on performance.

- There are several CVNN techniques, such as DCN, SurReal. For the DCN approach, the deep CNN models benefited while the shallow CNN did not. The multi-channel approach with absolute value and phase did not improve compared to three real-valued CNNs. SurReal made sense only for ResNet structure.

- Besides classification accuracy, the model complexity and generalization performance were also discussed. Shallow neural networks are more suitable for radar data and CVNNs do not help improve generalization ability.

For the main research question, the proposed complex-valued neural network solution further improves performance in some conditions, such as the most favourable one being deep CNN on range-Doppler and range-spectrum-time formats.

7.2. FUTURE WORK

In future work, more approaches or trends are worth exploring for radar-based human-motion classification developments. Some potential works are shown below.

- **Simulated data**: Simulated data are not ideal because they either have additional artifacts or miss some details compared to the experimental data. However, they can easily generate and test some parametric changes before finalizing and optimizing the network architecture with experimental data.

- **Data augmentation**: Data augmentation is the generation of synthetic data to expand the training dataset for optical images. Such transformations often imply rotation, flipping, scaling, moving of patterns within images. This method is not so straightforward for radar data, but it is worth trying. Preserving the kinematic meaning within the data is necessary.

- **Data fusion**: Multi-radar mode and the combination of radar and other sensors can be explored. Information fusion can be used to overcome domain/sensor by merging different complementary sensors at different abstraction levels, compensating for the shortcomings of a single sensor [10]. The effect is usually better than that of a single radar. However, the system will also be more complex.

7

- **Radar data formats**: Considering the result of Doppler-time from one range bin is not ideal, it is worth trying the Doppler spectrogram generated by summing the STFT data among the proper range bins containing contributions from the target. Other radar domains for classification, such as CVD, can also be explored.

- **Complex weight initialization and complex flatten layer**: For micro-Doppler signatures, spectrograms often are converted to a logarithmic scale. However, the log-scale is ambiguously defined for complex values. Proper weight initialization can explicitly combat this issue. DCN approach from Paper [5] ignored fully-connected distance transform of complex numbers when flattening. The complex flatten layer is challenging and worth exploring.

- **Other CNN models**: There are some other advanced CNN architectures, such as Alexnet, VGG, GoogleNet, to investigate. A smaller ResNet with fewer layers also can be exploited.

- **RNN**: While the sequential classification of dynamic motion remains an open problem, recent work exploiting RNNs have made significant progress toward addressing this issue. In the future, RNN, LSTM, and GRU will have great development, especially in gesture recognition, because there is a certain coherence between the front and back actions. CVNNs on RNN can be implemented for radar-based tasks to analyze their performance.

- **Transfer learning**: Radar data is always rare and hard to obtain. Rather than generating new radar data, transfer learning is to pre-train the classifier with a large dataset available from different fields, such as image and speech. In other words, the weights of DNN can be initialized by using the knowledge obtained from various fields for radar classification. The discarded samples in Chapter 5 also can be reused for initial transfer learning.

REFERENCES

[1] M. Richards, J. Scheer, J. Scheer, and W. Holm, *Principles of Modern Radar: Basic Principles, Volume 1*, ser. Electromagnetics and Radar. Institution of Engineering and Technology, 2010. [Online]. Available: https://books.google.nl/books?id=nD7tGAAACAAJ

[2] B. Mostafa, N. El-Attar, S. Abd-Elhafeez, and W. Awad, "Machine and deep learning approaches in genome: Review article," *Alfarama Journal of Basic Applied Sciences*, 08 2020.

[3] "What is a convolutional neural network?3 things you need to know," [EB/OL], https://nl.mathworks.com/discovery/convolutional-neural-network-matlab.html.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[5] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," 2018.

[6] "Hann function," [EB/OL], https://en.wikipedia.org/wiki/Hann_function.

[7] S. Z. Gurbuz and M. G. Amin, "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 16–28, 2019.

[8] L. Zhaohua and G. Bochao, "Radar sensors in automatic driving cars," in *2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)*, 2020, pp. 239–242.

[9] F. Fioranelli, J. Le Kernec, and S. A. Shah, "Radar for health care: Recognizing human activities and monitoring vital signs," *IEEE Potentials*, vol. 38, no. 4, pp. 16–23, 2019.

[10] J. Le Kernec, F. Fioranelli, C. Ding, H. Zhao, L. Sun, H. Hong, J. Lorandel, and O. Romain, "Radar signal processing for sensing in assisted living: The challenges associated with real-time implementation of emerging algorithms," *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 29–41, 2019.

[11] S. A. Shah and F. Fioranelli, "Rf sensing technologies for assisted daily living in healthcare: A comprehensive review," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 11, pp. 26–44, 2019.

[12] H. Li, X. Liang, A. Shrestha, Y. Liu, H. Heidari, J. Le Kernec, and F. Fioranelli, "Hierarchical sensor fusion for micro-gesture recognition with pressure sensor array and radar," *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology*, vol. 4, no. 3, pp. 225–232, 2020.

[13] Z. Chen, G. Li, F. Fioranelli, and H. Griffiths, "Dynamic hand gesture classification based on multistatic radar micro-doppler signatures using convolutional neural network," in *2019 IEEE Radar Conference (RadarConf)*, 2019, pp. 1–5.

[14] S. Z. Gurbuz, A. C. Gurbuz, E. A. Malaia, D. J. Griffin, C. S. Crawford, M. M. Rahman, E. Kurtoglu, R. Aksu, T. Macks, and R. Mdrafi, "American sign language recognition using rf sensing," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3763–3775, 2021.

[15] S. Z. Gurbuz, A. C. Gurbuz, E. A. Malaia, D. J. Griffin, C. Crawford, M. M. Rahman, R. Aksu, E. Kurtoglu, R. Mdrafi, A. Anbuselvam, T. Macks, and E. Ozcelik, "A linguistic perspective on radar micro-doppler analysis of american sign language," in *2020 IEEE International Radar Conference (RADAR)*, 2020, pp. 232–237.

[16] R. G. Guendel, F. Fioranelli, and A. Yarovoy, "Phase-based classification for arm gesture and gross-motor activities using histogram of oriented gradients," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 7918–7927, 2021.

[17] B. Erol and M. Amin, "Fall motion detection using combined range and doppler features," 08 2016, pp. 2075–2080.

[18] X. Yang, P. Chen, M. Wang, S. Guo, C. Jia, and G. Cui, "Human motion serialization recognition with through-the-wall radar," *IEEE Access*, vol. 8, pp. 186 879–186 889, 2020.

[19] H. Li, G. Cui, S. Guo, L. Kong, and X. Yang, "Human target detection based on fcn for through-the-wall radar imaging," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 9, pp. 1565–1569, 2021.

[20] Z. Ni and B. Huang, "Open-set human identification based on gait radar micro-doppler signatures," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8226–8233, March 2021.

[21] Z. Chen, G. Li, F. Fioranelli, and H. Griffiths, "Personnel recognition and gait classification based on multistatic micro-doppler signatures using deep convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 669–673, 2018.

[22] T. Nitta, "On the critical points of the complex-valued neural network," in *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, vol. 3, 2002, pp. 1099–1103 vol.3.

[23] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 4, pp. 541–551, 2012.

[24] X. Wang, P. Chen, H. Xie, and G. Cui, "Through-wall human activity classification using complex-valued convolutional neural network," in *2021 IEEE Radar Conference (RadarConf21)*, 2021, pp. 1–4.

[25] R. Chakraborty, J. Wang, and S. X. Yu, "Surreal: Fréchet mean and distance transform for complex-valued deep learning," *CoRR*, vol. abs/1906.10048, 2019. [Online]. Available: http://arxiv.org/abs/1906.10048

[26] D. A. Brooks, O. Schwander, F. Barbaresco, J.-Y. Schneider, and M. Cord, "Complex-valued neural networks for fully-temporal micro-doppler classification," in *2019 20th International Radar Symposium (IRS)*, 2019, pp. 1–10.

[27] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric sar image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7177–7188, 2017.

[28] T. Scarnati and B. Lewis, "Complex-valued neural networks for synthetic aperture radar image classification," in *2021 IEEE Radar Conference (RadarConf21)*, 2021, pp. 1–6.

[29] A. Fuchs, J. Rock, M. Toth, P. Meissner, and F. Pernkopf, "Complex-valued convolutional neural networks for enhanced radar signal denoising and interference mitigation," in *2021 IEEE Radar Conference (RadarConf21)*, 2021, pp. 1–6.

[30] B. N. Dash and N. Khare, "Deep complex neural network applications in remote sensing: an introductory review," in *Radar Sensor Technology XXV*, K. I. Ranney and A. M. Raynal, Eds., vol. 11742, International Society for Optics and Photonics. SPIE, 2021, pp. 34 – 44. [Online]. Available: https://doi.org/10.1117/12.2589161

[31] H. Griffiths, P. Knott, and W. Koch, "Christian hülsmeyer: Invention and demonstration of radar, 1904," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 9, pp. 56–60, 2019.

[32] V. 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, no. 1, pp. 2–21, 2006.

[33] V. Chen and H. Ling. Artech, 2001. [Online]. Available: https://ieeexplore.ieee.org/document/9101078

[34] S. Gurbuz, B. Erol, B. Tekeli, and B. Cagliyan, "Operational assessment and adaptive selection of micro-doppler features," *IET Radar, Sonar Navigation*, 11 2015.

[35] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers Geosciences*, vol. 19, no. 3, pp. 303–342, 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/009830049390090R

[36] S. Suthaharan, *Support Vector Machine*. Boston, MA: Springer US, 2016, pp. 207–235. [Online]. Available: https://doi.org/10.1007/978-1-4899-7641-3_9

[37] O. Kramer, *K-Nearest Neighbors*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 13–23. [Online]. Available: https://doi.org/10.1007/978-3-642-38652-7_2

[38] W. S. Sarle, "Neural networks and statistical models," 1994.

[39] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608014002135

[40] W. Luo, J. Li, J. Yang, W. Xu, and J. Zhang, “Convolutional sparse autoencoders for image classification,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3289–3294, 2018.

[41] A. Sherstinsky, “Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network,” *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167278919305974

[42] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. O’Reilly Media, Inc., 2019.

[43] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015.

[44] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[45] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[46] A. F. Agarap, “Deep learning using rectified linear units (relu),” 2019.

[47] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra, “Reducing overfitting in deep networks by decorrelating representations,” 2016.

[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, Jan. 2014.

[49] A. Krizhevsky, “Learning multiple layers of features from tiny images,” pp. 32–33, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[50] Y. Li, K. He, D. Xu, and D. Luo, “A transfer learning method using speech data as the source domain for micro-doppler classification tasks,” *Knowledge-Based Systems*, vol. 209, p. 106449, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120305785