Micro-Doppler Classification of Humans and Animals Using FMCW Radar

Presented by:
Amisha Manga

Prepared for:
Dr. Stephen Paine, Dr. John-Philip Taylor
Dept. of Electrical Engineering
University of Cape Town

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Master of Science degree in Electrical Engineering with Specialization in Radar.

January 2025

Key words:
Classification, Micro-Doppler, Radar, Detection, FMCW

I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own. This thesis/dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.

Signed by candidate

. . . .

Amisha Manga

Micro-Doppler Classification of Humans and Animals Using FMCW Radar

Amisha Manga

January 2025

Abstract

This research attempts to address the issue of animal poaching by exploring human and animal target classification by making use of micro-Doppler data generated by a Frequency-Modulated Continuous Wave (FMCW) radar. Raw Analogue to Digital Converter (ADC) data was collected of human and animal species including dogs, horses and cows. Signal processing techniques such as creating range-Doppler and Constant False Alarm Rate (CFAR) maps for detection and using Short-Time Fourier Transform (STFT) for spectrogram generation were applied. Principal Component Analysis (PCA) was employed as the method for data reduction.

The dataset was subsequently classified and evaluated across various target class configurations, comparing both the full dataset and its PCA-reduced versions, using Convolutional Neural Networks (CNN), k-Nearest Neighbors (kNN), Random Forest (RF) and Support Vector Machine (SVM) models. Following this, a two-stage classification process was implemented to further refine the analysis.

In the first stage, the 4 above-mentioned classifiers were used to distinguish between humans and animals. In the second stage, these classifiers differentiated among the specific animal species. The study experimented with all 16 permutations of classifier combinations, such as SVM-CNN and kNN-RF. Notably, the SVM-SVM combination achieved the highest accuracy at 97.66%, closely matching the 97.5% accuracy outcome obtained in the multi-class classification of humans versus dogs, horses and cows. The PCA-reduced set yielded results closely comparable to those from the full dataset evaluation, confirming its effectiveness.

This study highlights the challenges of data collection in natural settings and the need for publicly accessible micro-Doppler data of animal targets to further this research area. Recommendations for future work include developing tracking algorithms tailored to various animal movement patterns. The findings indicate that low cost radar surveillance systems with micro-Doppler classification technology hold promising application in animal conservation-oriented efforts.

Acknowledgements

I would like to express my sincerest gratitude and thanks;
To Mr. Pieter-Jan Wolfaardt for providing the opportunity to do this research;
To Reutech Radar Systems for providing the radar used in this research and generously allowing me to use their resources;
To STA Coal Mining Company (Pty) Ltd and the Council for Scientific and Industrial Research for their contributions towards my tuition;
To Mr. Daniel Naudé for graciously allowing the Ankole cattle at Yonder Hill Farm to be a part of the dataset used in this research;
To Mr. Rudolph Johannes Willemse Jnr and Mrs. Linsy Willemse for allowing their dog, Loki, to be a part of the dataset used in this research;
To Mr. Derrick Carstens for allowing the setup of the radar and data capture activities on multiple occasions at Pinto Stables;
To Mr. Thierry Feret and Ms. Kayla Feret for the endless amount of assistance provided to me with the horses at Pinto Stables;
To Mr. Eddie Oosthuizen and Mrs. Theresa Oosthuizen for allowing their dog, Link, to be a part of the dataset used in this research;
To my parents, family, friends and colleagues for the unconditional support and motivation;
To my partner, Mr. Privaan Ramchunder, for assisting me with all the data capture activities involved in this research;
And biggest thanks to my supervisors Dr. John-Philip Taylor and Dr. Stephen Paine for the endless guidance, patience, technical and emotional support throughout my study.

Contents

Acknowledgements **i**

List of Figures and Listings **xi**

List of Tables **xiv**

Nomenclature **xv**

1 Acronyms xv

2 Terminology xviii

1 Introduction **1**

1.1 Background 1

1.2 Problem Description 2

1.3 Objectives 2

1.4 Project Scope 3

1.5 Challenges and Limitations 4

1.6 Research Contribution 4

1.7 Document Outline 5

2 Literature Review **7**

2.1 Background 7

2.1.1 The Micro-Doppler Effect in Radar 7

2.1.2 Short-Time Fourier Transform 8

2.1.3 Spectrogram 9

2.2 Classification Techniques 10
2.2.1 Feature Extraction 10
2.2.2 Support Vector Machines 10
2.2.3 k-Nearest Neighbors 10
2.2.4 Random Forests 11
2.2.5 Hidden Markov Model – Gaussian Mixture Model 11
2.2.6 Neural Networks 12
2.3 Existing Work and Previous Studies 13
2.3.1 Autonomous Driving and Vehicle Safety 14
2.3.2 Human Gait Classification 14
2.3.3 Elderly Fall Detection 15
2.3.4 Wildlife Monitoring and Conservation Efforts 16
2.3.5 Animal Lameness Detection 17
2.3.6 UAV Classification 19
2.4 Research Gap 20
2.4.1 Classification of Different Animal Species 20
2.4.2 Comprehensive Datasets 20
2.4.3 Cost-Effective Radar Systems 20

3 Data Collection and Pre-Processing 21

3.1 LCR System 21
3.2 Pre-Processing Methodology 24
3.2.1 Data Logging and Acquisition 24
3.2.2 Remove DC Offset 25
3.2.3 Range FFT 25
3.2.4 Doppler FFT 25
3.2.5 Range-Doppler Map 25
3.2.6 Constant False Alarm Rate 26

3.3 Data Acquisition- Human Database 29
3.3.1 RRS Rooftop 29
3.3.2 Wintervogel Airstrip 30
3.3.3 RRS Parking Lot 30
3.4 Data Acquisition- Animal Database 31
3.4.1 Dogs 31
3.4.2 Horses 32
3.4.3 Cows 33

4 Data Preparation **35**
4.1 Data Verification 35
4.2 Target Extraction in Range 35
4.3 Spectrogram Generation 37
4.4 Spectrogram Database 38

5 Classification Algorithm Design **41**
5.1 Data Augmentation 41
5.2 Data Split 43
5.3 Principal Component Analysis 43
5.4 Data Preparation for Classification Models 43
5.5 Parameter Optimization 44
5.6 Support Vector Machine 44
5.7 k-Nearest Neighbors 45
5.8 Random Forests 45
5.9 Convolutional Neural Networks 46
5.9.1 Data Preparation 46
5.9.2 Model Design 47
5.9.3 Compiling and Training 48
5.10 Model Evaluation 50
5.10.1 Accuracy Assessment 50
5.10.2 Classification Report 50

6 Experimentation **53**

6.1 Micro-Doppler Spectrogram Data . . . 53

6.1.1 Spectrogram Parameters . . . 53

6.1.2 Analysis of Micro-Doppler Patterns . . . 54

6.1.3 Data Augmentation Technique and Visualisation . . . 56

6.2 Dataset Composition . . . 58

6.3 Radar Hardware Degradation . . . 58

6.4 Principal Component Analysis . . . 59

6.5 Classifier Parameter Optimisation . . . 63

6.5.1 Analysis of Parameters . . . 63

7 Results **65**

7.1 Classification Results . . . 65

7.1.1 Original Dataset Performance . . . 66

7.1.2 PCA-Reduced Dataset Performance . . . 67

7.1.3 Analysis of Results from Original and PCA-Reduced Datasets . . . 67

7.2 Multi-Stage Classification Process . . . 68

7.3 Analysis of Results . . . 70

7.3.1 SVM-SVM Classification . . . 70

7.3.2 Two-Stage Classification Analysis . . . 73

8 Conclusion **75**

8.1 Research Summary . . . 75

8.2 Discussion of Results . . . 76

8.3 Challenges . . . 77

8.4 Further Work . . . 77

8.5 Concluding Remarks . . . 78

References **92**

A FMCW Radar and Signal Processing Techniques **93**
A.1 LCR System 93
A.2 FMCW Radar 95
A.2.1 Range Measurement 96
A.2.2 Velocity Measurement 97
A.2.3 Range-Doppler Map 100
A.3 CFAR Target Detection 101
A.3.1 CA-CFAR 102
A.4 Spectrogram Generation 104

B Classification Techniques **107**
B.1 Principal Component Analysis 107
B.2 Support Vector Machines 108
B.3 k-Nearest Neighbors 109
B.4 Random Forests 110
B.5 Hidden Markov Model 111
B.6 Gaussian Mixture Models 113
B.7 Neural Networks 114
B.7.1 Convolutional Neural Networks 114

C Photographs of Data Capture Sessions **117**

D Spectrogram Images of Human and Animal Targets **127**

List of Figures and Listings

1 Introduction **1**

2 Literature Review **7**

3 Data Collection and Pre-Processing **21**

Fig. 3.1: Radar Deployment at Wintervogel Airstrip (left) and at RRS (right) . . 22

Fig. 3.2: LCR Radar Overview 23

Fig. 3.3: Data Structure of a Single Burst 25

Fig. 3.4: Range-Doppler and CA-CFAR Maps Illustrating a Target 28

4 Data Preparation **35**

Fig. 4.1: Person Target at Approximately 35 m Range (Range Bin 19) 36

Fig. 4.2: Creating Spectrogram Database: Animal (Herd of Horses) Target 39

5 Classification Algorithm Design **41**

Fig. 5.1: CNN Model Architecture 49

6 Experimentation **53**

Fig. 6.1: Micro-Doppler Patterns of Various Targets Walking . . . 54

Fig. 6.2: Comparative Micro-Doppler Patterns: Human Hand-Waving Versus Horse Tail-Swinging . . . 56

Fig. 6.3: Spectrograms of Horse Walking Towards Radar Illustrating Data Augmentation Technique . . . 57

Fig. 6.4: Spectrograms of Horse Walking . . . 59

Fig. 6.5: Cumulative Explained Variance Showing the Number of Principal Components Capturing 99% of the Variance . . . 61

Fig. 6.6: Comparison of Original and Reconstructed Spectrograms of a Person Walking Using PCA . . . 62

7 Results **65**

Fig. 7.1: SVM First Stage . . . 71

Fig. 7.2: SVM Second Stage . . . 72

8 Conclusion **75**

A FMCW Radar and Signal Processing Techniques **93**

Fig. A.1: Transmitted and Received Chirp Frequencies as a Function of Time . . . 95

Fig. A.2: Velocity Measurement for Multiple Objects Moving at the Same Range . . . 99

Fig. A.3: Range FFT . . . 100

Fig. A.4: Range-Doppler FFT Map . . . 100

Fig. A.5: CFAR Detection with Range Threshold and Target Signal . . . 102

Fig. A.6: CA-CFAR Detection for Range-Doppler Maps . . . 103

B Classification Techniques **107**

Fig. B.1: Support Vector Machine 109

Fig. B.2: k-Nearest Neighbor Example 110

Fig. B.3: Random Forests 111

Fig. B.4: Markov Chain Example 112

Fig. B.5: Neural Network Architecture 114

Fig. B.6: Basic CNN Architecture 116

C Photographs of Data Capture Sessions **117**

Fig. C.1: RRS Rooftop Radar Setup 118

Fig. C.2: RRS Parking Radar Setup 119

Fig. C.3: Snapshots from Person Dataset Captured at RRS 120

Fig. C.4: Radar Setup at Paardevlei Old Industrial 121

Fig. C.5: Golden Retriever - Link; Link Walking Toward Radar; Link Running Away from Radar 122

Fig. C.6: Border Collie x Staffordshire Bull Terrier - Loki; Loki Walking Away from Radar; Loki Walking Toward the Radar 123

Fig. C.7: Pinto Stables: Radar Recording Data of Herd of Horses 124

Fig. C.8: Pinto Stables: Radar Recording Data of Stallion - Seun 125

Fig. C.9: Yonder Hill Radar Setup 126

D Spectrogram Images of Human and Animal Targets **127**

Fig. D.1: Spectrograms of Different Human Movements Part 1 128

Fig. D.2: Spectrograms of Different Human Movements Part 2 129

Fig. D.3: Spectrograms of Different Dog Movements 130

Fig. D.4: Spectrograms of Different Horse Movements 131

Fig. D.5: Spectrograms of Different Cow Movements 132

List of Tables

1 Introduction **1**

2 Literature Review **7**

3 Data Collection and Pre-Processing **21**

TABLE 3.1 LCR System Parameters 24

TABLE 3.2 CFAR Parameters 27

TABLE 3.3 Distribution of Individuals by Location 29

TABLE 3.4 Speed of Motion Corresponding to Different Actions 29

4 Data Preparation **35**

5 Classification Algorithm Design **41**

6 Experimentation **53**

TABLE 6.1 STFT Parameters 53

TABLE 6.2 Dataset Composition by Target Type 58

TABLE 6.3 Composition of the Datasets 58

TABLE 6.4 Parameter Optimisation: Results from GridSearchCV 63

7 Results **65**

TABLE 7.1 Classification Accuracy Results 66

TABLE 7.2 Two-Stage Classification Accuracy Results 69

TABLE 7.3 SVM-SVM Classification Report 70

8 Conclusion **75**

A FMCW Radar and Signal Processing Techniques **93**

TABLE A.1 Parameters of LCR system 94

B Classification Techniques **107**

C Photographs of Data Capture Sessions **117**

D Spectrogram Images of Human and Animal Targets **127**

Nomenclature

1 Acronyms

AC .... Alternating Current
ADC .... Analogue to Digital Converter
ANN .... Artificial Neural Network
API .... Application Programmable Interface
AR .... Autoregressive

BW........... Bandwidth

CA-CFAR .... Cell-Averaging CFAR
CFAR .... Constant False Alarm Rate
CH .... Channel
CNN .... Convolutional Neural Network
CPU.......... Central Processing Unit
CSIR .... Council for Scientific and Industrial Research
CUT.......... Cell Under Test
CW........... Continuous Wave

DAC.......... Digital to Analogue Converter
DC .... Direct Current
DFT.......... Discrete Fourier Transform
DSP .... Digital Signal Processor (or processing)
DT .... Decision Tree
DTFT .... Discrete-Time Fourier Transform

EM . . . . Expectation-Maximization

FIFO . . . . First-In First-Out

FIR. . . . . Finite Impulse Response

FFT . . . . Fast Fourier Transform

FMCW . . . . Frequency-Modulated Continuous Wave

FPGA . . . . Field Programmable Gate Array

G . . . . Giga

GB . . . . Gigabyte

GMM. . . . . Gaussian Mixture Model

HMM . . . . Hidden Markov Model

Hz . . . . Hertz

ICA . . . . Independent Component Analysis

IF . . . . Intermediate Frequency

IP . . . . Internet Protocol

kNN . . . . k-Nearest Neighbors

LCR. . . . . Low Cost Radar

LDA . . . . Linear Discriminant Analysis

LoRa . . . . Long Range

LPC . . . . Linear Predictive Coding

M . . . . Mega

MATLAB. . . . . Matrix Laboratory

MLE . . . . Maximum Likelihood Estimation

mmWave. . . . . . Millimetre Wave

MOCAP . . . . . . Motion Capture

MUSIC . . . . Multiple Signal Classification

NB . . . . Naive Bayes

NN . . . . Neural Network

No. . . . . Number

PCA . . . . Principal Component Analysis

PC . . . . Personal Computer
PDF . . . . Probability Density Function
PD . . . . Probability of Detection
PFA . . . . Probability of False Alarm
PoE . . . . Power over Ethernet
PRF . . . . Pulse Repetition Frequency
PRI . . . . Pulse Repetition Interval

RADAR . . . . . . Radio-Assisted Direction and Ranging
RAM . . . . Random-Access Memory
RBF . . . . Radial Basis Function
RCS . . . . Radar Cross Section
ReLU . . . . Rectified Linear Unit
ResNet . . . . Residual Network
RF . . . . Random Forests
RGB . . . . Red, Green, and Blue
RRS . . . . Reutech Radar Systems
Rx . . . . Receive

SFC . . . . Sensitivity Frequency Control
SPI . . . . Serial Peripheral Interface
STFT . . . . Short-Time Fourier Transform
SVC . . . . Support Vector Classifier
SVD . . . . Singular Value Decomposition
SVM . . . . Support Vector Machine

Tx . . . . Transmit

UAV . . . . Unmanned Aerial Vehicle
UK . . . . United Kingdom
UWB . . . . Ultra-Wideband

ZMQ . . . . ZeroMQ

2 Terminology

Classification	Refers to the process of identifying and categorising radar return signals into different types of targets using signal characteristics and pattern recognition for decision making [1].
Epoch	An epoch in machine learning refers to one complete pass of the training data though the learning algorithm [2].
FMCW Radar	Frequency-Modulated Continuous Wave (FMCW) radar is a type of radar system that continuously transmits a signal with a varying frequency [3].
Low Cost Radar	The Low Cost Radar (LCR) is the FMCW radar system developed by Reutech Radar Systems (RRS) that is employed in this research.
Micro-Doppler	Micro-Doppler refers to the small variations in the radar return caused by the motion of individual components within a target [4].
Poaching	Poaching refers to the illegal hunting, capturing, or killing of protected wildlife species [5].
Tensor	A tensor is a multi-dimensional array that generalizes scalars, vectors and matrices to higher dimensions, serving as a fundamental data structure to represent complex data and operations [6].
X-band	The X-band refers to the designation of a specific range of frequencies located within the microwave region of the electromagnetic spectrum covering frequencies from 8 to 12 GHz [7].
ZeroMQ	ZeroMQ is an open-source high-performance asynchronous messaging library [8].

Chapter 1

Introduction

1.1 Background

The African continent, renowned for its biodiversity, faces significant threats from illegal poaching, impacting iconic species such as lions, elephants and rhinoceri. The data reported by conservation organizations [9, 10] paint a distressing picture, with drastic declines in populations and frequent poaching incidents. The African Wildlife Foundation [11] reports a decline of 43% in the lion population over the past 21 years and that approximately 35,000 African elephants are killed by poachers annually [11], while Save the Rhino [12] reveals a reality where one rhino is poached every 16 hours.

South Africa's Kruger National Park and adjacent private reserves represent the epicentres of these challenges but also focal points for ongoing conservation efforts and anti-poaching strategies. In response to escalating poaching rates, the reserves have implemented security measures such as polygraph testing, Long Range (LoRa) tracking [13, 14], thermal and video surveillance technologies [15] as well as de-horning programs [10]. These recent efforts have led to a reduction in poaching incidents within the reserves [10, 16], underlining the effectiveness of these security measures. However, challenges persist as poachers adapt their strategies, shifting to other reserves and hotspots [16], highlighting the necessity for continuous innovation in anti-poaching techniques.

Radar technology offers a robust approach for monitoring wide geographical areas continuously. These systems are highly effective at detecting movement in real time, making them ideal for wildlife surveillance and anti-poaching initiatives by providing early detection and alerts for potential illegal activities.

1.2 Problem Description

The protection of wildlife and livestock against illegal poaching is a critical concern, necessitating practical and cost-effective technological solutions to support ongoing conservation efforts.

To address these challenges, Reutech Radar Systems (RRS) in collaboration with multiple entities [17] including the United Kingdom (UK) People's Postcode Lottery, the South African Department of Environmental Affairs, SANParks, Peace Parks Foundation and the Council for Scientific and Industrial Research (CSIR) have developed the RSR-904, colloquially known as "Ngada", which translates to "Meerkat" in the Nguni languages [18, 19]. The radar is a ground surveillance system that uses Frequency-Modulated Continuous Wave (FMCW) technology in the X-band spectrum. This radar delivers reliable detection performance under adverse weather and low-light conditions [19], making it a valuable complement to traditional methods such as optical surveillance [15]. The surveillance capabilities aid in identifying and tracking unauthorised individuals or vehicles that breach restricted areas [18].

One of the advantages of using radar technology in wildlife protection is its ability to generate unique micro-Doppler signatures of different species. By leveraging these movement patterns for target classification, radar technology can be used to distinguish between humans and animals, offering valuable support for anti-poaching efforts [20] and conservation-oriented initiatives.

This research seeks to advance the development of this technology by investigating various classification techniques on micro-Doppler data generated using an alternative, more cost-effective radar system (approximate cost of 6000 USD).

1.3 Objectives

This research explores the use of a low cost radar system in order to perform classification on micro-Doppler spectrogram data and encompasses the following aspects:

- **Real-World Data Collection:** Data of human and animal targets are obtained from a low cost, FMCW radar system in order to enhance the practical applicability of research findings, moving beyond simulated environments.
- **Spectrogram Generation and Analysis:** Perform signal processing on collected data to produce micro-Doppler spectrograms, identifying unique characteristics and

features that distinguish the movement patterns and profiles of human and animal targets.

- **Application of Classification Techniques:** Research and implement classification methodologies on the micro-Doppler spectrogram data in order to assess the radar's ability to accurately classify into three categories:
  - **Binary Classification**: Distinguishing between humans and animals.
  - **Multi-Class Human and Specific Animal Classification**: Differentiating humans from specific animals such as dogs, horses and cows.
  - **Multi-Class Animal Classification**: Distinguishing between animal species.

1.4 Project Scope

The scope of this research project is defined below:

- **Livestock Animals:** Data capture of animals will be limited to livestock animals as securing entry to wildlife reserves housing animals like elephants and lions is a challenge. The radar employed in this study has a maximum range of 1 km, therefore deploying the radar within 1 km of any wildlife animal is a formidable task.
- **Data Collection:** The project involves collecting radar data of both human and animal targets.
- **Signal Processing:** Signal processing techniques are performed on the captured radar data in order to detect targets and generate spectrogram images representing the micro-Doppler movement patterns.
- **Exclusion of a Tracker:** The project does not include the implementation of a tracking component in the signal processing chain. The focus is primarily on the classification of targets based on their movement patterns rather than the tracking of their movements.
- **Implementation of Different Classifiers:** Different classification algorithms are designed, implemented and assessed based on its ability to differentiate between targets using micro-Doppler spectrogram data.
- **Animal vs Human Classification:** The focus is placed primarily on assessing the classification algorithm's ability to distinguish between human and animal targets.

1.5 Challenges and Limitations

Challenges and limitations faced in the duration of this research effort are listed below:

- **Limited Length of Continuous Movement from Animal Subjects:** Certain animals, such as cows, have shorter periods of movement when compared to other species thereby posing a slight challenge in capturing representative movement data of the individual species.
- **Consistent Environment for Data Capture Activities:** Conducting all data capture activities in the same environment is challenging, as relocating the livestock animals is not feasible.
- **Environmental Factors:** Interference caused by environmental factors such as weather, terrain and vegetation can affect the strength of the signal returned by the target.
- **Radar Availability:** Time available to use the radar for data capture activities is limited.
- **Radar Hardware Degradation:** The degradation of the radar hardware during the course of the project has hampered the objective of comprehensive data collection.
- **Computation Power:** Limited computational capacity (16 GB of RAM and 13th Generation Intel(R) Core i7-13800H processor: 20 CPUs, 2.5 GHz) posed a hurdle in efficiently experimenting with classification models. The classification process required running numerous iterations with large datasets and complex models, which were extremely time-intensive and ultimately constrained the scope of the research.

1.6 Research Contribution

The primary focus of this research is to develop and assess the capability of various classification algorithms to accurately differentiate between humans and animal species based on their micro-Doppler signatures generated by using a low cost, FMCW X-Band radar. This research, exploiting real data, aims to provide valuable insights into the feasibility of using micro-Doppler radar technology as additional support for anti-poaching operations and wildlife protection efforts.

1.7 Document Outline

Chapter 2 provides an in-depth review of the relevant literature of current radar applications with focus on target classification. This review serves as the foundation for understanding the current state of the art and identifying any research gaps.

Chapters 3 and 4 detail the methodology for capturing radar data to create a database of targets in the form of micro-Doppler spectrograms. It covers the radar system used, the data collection and acquisition process, the signal processing techniques for generating the spectrograms and the various experimental setups involved.

The design and development of the various algorithms selected for target classification is presented in Chapter 5. It also covers data augmentation and dimensionality reduction techniques. Chapter 6 provides an analysis of the spectrogram patterns of various targets. It discusses the dataset composition used for classification, linking it to the initial objectives outlined in Section 1.3. Additionally, it presents the results from the classifier model parameter optimization stage.

Chapter 7 presents the results and analysis of the various classification techniques applied to the data captured in this research. Finally, Chapter 8 concludes the research by summarising the key findings and contributions of the research. Potential areas for future work are proposed.

Chapter 2

Literature Review

RADAR is an acronym comprised of the following words: **RA**dio **D**etecting **A**nd **R**anging [21]. It can be broadly defined as an electronic system that uses reflected electromagnetic energy to detect and locate the presence of objects in its vicinity [21].

The use of radar in order to detect and classify living beings such as humans and animals has gained increased attention and research due to its potential applicability in various scenarios. This chapter initiates with a brief explanation of the micro-Doppler effect followed by an exploration of the various classification methodologies applied in the micro-Doppler domain to achieve target classification. A review of previous studies that have leveraged micro-Doppler radar properties for target classification is presented. Additionally, this review aims to identify and detail any gaps within the existing research, highlighting areas for future investigations.

2.1 Background

2.1.1 The Micro-Doppler Effect in Radar

Chen's book, *The Micro-Doppler Effect in Radar* [22], provides a comprehensive and insightful explanation of the micro-Doppler effect in radar. The following overview summarises the key concepts discussed in his book.

The radar transmits a signal towards an object and receives the corresponding reflected signal [3]. By analysing the time delay of the received signal, the radar can determine the range of the object [3]. If the object is in motion, the frequency of the received signal experiences a shift from the transmitted signal [3, 22]. This phenomenon is commonly termed as *the Doppler Effect*: "By examining the Doppler frequency shift, the radar can measure the radial velocity of the moving object" [22].

"If the object or any structural component of said object undergoes oscillatory motion alongside its bulk motion, the oscillation will induce an additional frequency modulation on the returned signal" [22]. This modulation creates "sidebands around the Doppler shifted frequency caused by the object's bulk motion" [22], resulting in what is known as *the Micro-Doppler Effect* [4, 22].

The micro-Doppler effect manifests as a distinctive signature and encapsulates the object's unique motion characteristics. Through the analysis of the joint time and Doppler frequency domain, it is possible to uniquely depict this frequency modulation, thereby obtaining valuable insights into the kinematic properties that give an object its identity [22].

2.1.2 Short-Time Fourier Transform

In order to observe how an object's signal spectral content varies with time, the Short-Time Fourier Transform (STFT) is typically employed. Chen [22] highlights its ease of implementation, computational efficiency and well-established methodology.

The STFT, also known as the windowed Fourier transform or sliding Fourier transform, divides the time-domain input signal into shorter, overlapping blocks [23]. Window functions [24] are selected and applied to each block in order to taper the signal, thereby reducing side lobes. A Fast Fourier Transform (FFT) is then performed on each windowed block of the signal.

As a result, the STFT generates a time-frequency representation of the signal, reflecting its spectral content at distinct time intervals. By shifting the window, the STFT captures the evolving spectral content of the signal across various periods, representing changes in both time and frequency. The spectrum produced by the STFT represents the Doppler frequency shift and the sideband modulation, known as the micro-Doppler signature [22].

In practice, the STFT is often assessed at discrete frequency points along the frequency axis. The process of computing the Discrete Fourier Transform (DFT) based STFT is described as follows [23]: The time domain signal, $x[m]$ is multiplied by a window function $w[n-m]$ and thereafter the Fourier transform is computed on the product.

$$X[n,k] = \sum_{m=n-(N_\omega-1)}^{n} x[m]w[n-m]e^{-j\omega_k m} \quad \text{,where} \quad \omega_k = \frac{2\pi k}{N} \tag{2.1}$$

In equation 2.1, N specifies the number of discrete frequency channels, and N_ω indicates the length of the finite-duration window function $w[n]$. $X[n,k]$ depends on discrete time n and discrete frequency k indices. The position of the analysis window along the time axis, termed the analysis frame, is denoted by n. The frequency index, also referred to as a frequency bin, is denoted by k [23].

The choice of window function and length in the STFT directly impacts the balance between time and frequency resolution. A narrow (short) window has a short time duration, offering fine temporal resolution at the expense of low frequency resolution [23, 24]. Conversely, a wide (long) window has a long time duration, providing sharper spectral resolution but poor time resolution [24].

While the STFT produces a fairly good representation of the micro-Doppler signal, it involves balancing trade-offs between time and frequency resolution. Advanced techniques such as Multiple Signal Classification (MUSIC) [25], Wavelet Transforms [26], Empirical Mode Decomposition [26], Wigner-Ville distribution [22, 27], and the S-Transform [28] aim to overcome the challenges faced with balancing time and frequency resolution.

2.1.3 Spectrogram

The STFT is used to produce a spectrogram, which depicts the spectral density of a time-varying signal. It is a two-dimensional spectro-temporal representation [22] revealing the dynamic change in frequency components within a signal across time.

The spectrogram is computed by using the squared magnitude of the $X[n,k]$, which is the result of the STFT. The discrete-time representation of the spectrogram presented below [22]:

$$\text{Spectrogram}[n,k] = |\mathrm{X}[n,k]|^2 \tag{2.2}$$

2.2 Classification Techniques

There are several notable classification methods that have been successful for micro-Doppler classification, as reported in literature. A few of these techniques are selected for classification purposes in this research and will be discussed in this section.

2.2.1 Feature Extraction

For spectrogram classification, there are several dimensionality reduction techniques, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Independent Component Analysis (ICA), each offering unique advantages and applications as detailed in [29].

However, for the purposes of this research, PCA has been selected as the method for extracting features from the spectrograms due to its simplicity, effectiveness and computational efficiency. Additionally, PCA has demonstrated considerable success in similar applications, as evidenced by its positive outcomes in related studies [29–33].

PCA is a linear dimensionality reduction technique used to reduce the number of components in a feature vector while still preserving the feature's discriminative ability. This is done by converting the original variables into a new set of uncorrelated variables, known as the principal components [34]. For more information with regard to how the PCA algorithm works, refer to Appendix B.1.

2.2.2 Support Vector Machines

Support Vector Machines (SVMs) are highly regarded for micro-Doppler classification due to their efficacy in managing high-dimensional data and separating complex feature spaces, as demonstrated in numerous studies [35–42]. It utilises the kernel trick method to address non-linear relationships, which helps the SVM to differentiate subtle variations in the signatures. For a more detailed understanding of the SVM's operation, refer to Appendix B.2.

2.2.3 k-Nearest Neighbors

The k-Nearest Neighbors (kNN) algorithm is one of the most fundamental techniques used for classification tasks and finds extensive application in micro-Doppler classification as evidenced by various studies [36, 37, 43–46], due to its simplicity and ease of implementation.

The kNN algorithm is advantageous for micro-Doppler classification due to its non-parametric nature, which ensures flexibility across diverse feature spaces without assuming a specific data distribution. This, combined with its effectiveness in small datasets, minimal hyper parameter tuning (as k, the number of neighbors, is the only parameter), and its ability to identify similar patterns based on feature space proximity, makes kNN a good choice for classifying micro-Doppler signatures. Refer to Appendix B.3 for a deeper explanation of the kNN algorithm.

2.2.4 Random Forests

The Random Forest (RF) classification algorithm is well-suited for micro-Doppler spectrogram classification, as highlighted in literature [47–49]. The random forest model fits a number of decision tree classifiers on several sub-samples of the dataset and uses an averaging technique to improve the predictive accuracy performance and limit over-fitting [50].

Its capability to capture complex, non-linear relationships, without the need for data transformation, makes it a good choice for analysing the intricacies of micro-Doppler signatures. Furthermore, Random Forest performs well with imbalanced datasets by employing strategies to mitigate bias towards the majority class [50]. Refer to Appendix B.4 for more details.

2.2.5 Hidden Markov Model – Gaussian Mixture Model

Prior to deep learning models, the combination of the Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM) were fundamental components of speech recognition systems [51]. These models play a role in interpreting time-varying signals, such as those found in micro-Doppler spectrograms. These signals can be considered as sequences of states (e.g., different modes of motion) with observations that can vary within each state [52, 53] due to factors such as speed variation and signal interference.

The HMM component of the HMM-GMM model is used to model the sequential aspect of the spectrogram, as it can effectively represent the transitions between different states over time [51]. This ability to model the temporal dynamics make HMMs a good choice for micro-Doppler classification tasks, as outlined in the literature [54–56].

GMM based classification methods are largely applied to micro-Doppler classification tasks as showcased in literature [4, 55, 57–60] as they have the ability to approximate any density with a finite mixture of Gaussians [61] thereby justifying their application in classification tasks. The GMM component of the HMM-GMM model is used to model the observation distributions within each state [51]. Movement patterns from a single state (e.g., walking at a constant speed) can exhibit considerable variation due to factors such as individual gait, observation angles, environmental clutter and so on. GMMs can capture this variability by representing the observation likelihood as a mixture of multiple Gaussian distributions [62,63], each corresponding to different facets of the same motion state [51].

Together, the HMM-GMM framework offers a nuanced approach to micro-Doppler classification. While HMMs addresses the temporal sequence of states, GMMs provide a detailed and adaptable model of the observations within these states [57]. This combination is effective for managing the stochastic processes underlying various movement patterns [51].

A decision is made against the implementation of the model as using GMMs as emission probabilities in the HMM, significantly raises the computational burden. This is due to the necessity to compute and adjust a large set of parameters (means, variances and mixture coefficients) for each GMM component across every HMM state. This complexity grows with the number of states and mixture components. Additionally, a shift in technology towards more advanced deep learning models has resulted in decreased support and development for HMM-GMM models in major frameworks, favouring Neural Network (NN) architectures instead.

Refer to Appendices B.5 and B.6 for further information on these models.

2.2.6 Neural Networks

Convolutional Neural Networks (CNN's) have the ability to extract and learn feature representations directly from raw image data [64]. The CNN model automates feature extraction, learning intricate patterns and characteristics which is particularly advantageous for classifying micro-Doppler images [64]. By applying convolutional layers, which act as learnable filters, CNN's are able to identify spatial hierarchies of features from simple edges in the initial layers to complex motion patterns in deeper layers [65]. CNN's have demonstrated successful application [39, 66–70] in image based classification using micro-Doppler patterns. Refer to Appendix B.7 for further details on the CNN.

2.3 Existing Work and Previous Studies

Micro-Doppler technology finds diverse applications across multiple industries, showcasing its versatility. Below are several notable use cases that illustrate its wide ranging potential:

- **Aerospace and Defence:** Micro-Doppler radar technology can be used in aerospace and defence applications for target identification, missile guidance, Unmanned Aerial Vehicle (UAV) detection and surveillance purposes. Classifying drone types based on their micro-Doppler signature is useful for monitoring unauthorised drone activity [31, 71].

- **Surveillance and Security:** Radars can be deployed for perimeter surveillance in restricted areas, aiding in the identification and classification (using micro-Doppler technology) of potential threats or intruders [18].

- **Wind Farms:** Micro-Doppler radar systems can be used to detect and classify the presence of flying animals (such as birds and bats) near turbines based on their wing-flapping patterns in order to provide early warning, thereby reducing the risk of collision [71–73].

- **Wildlife and Farmed Animals Monitoring:** Classifying animal species and tracking their movements, based on their micro-Doppler patterns, without physical tags or harming them is useful in the conservation and study of animal behaviours [36, 55, 74].

- **Health and Elderly Care:** Radar equipped with micro-Doppler technology can be used in monitoring and classifying human motions such as falls, which is essential for patient and elderly care in medical and home environments [67].

- **Gesture Control:** Micro-Doppler radar can be integrated into technology to detect and classify specific hand or arm gestures [71]. Devices equipped with gesture detection could possibly be used in applications to translate sign language into written text or spoken words in real time.

- **Road Safety and Traffic Management:** Micro-Doppler radar technology can assist in advanced driver assistance systems by classifying traffic signalling motions [68], pedestrians, cyclists, and other vehicles in order to improve situational awareness and reduce accidents.

The following subsections explore specific application areas of micro-Doppler technology, each of which contributes valuable insights and serve as inspiration for this research.

2.3.1 Autonomous Driving and Vehicle Safety

Micro-Doppler radar applications can enhance automotive safety by detecting and classifying objects around vehicles. It enables collision avoidance systems, blind-spot detection, pedestrian detection and gesture-based vehicle control.

Project SALUS, currently being conducted by researchers at the Universities of Applied Sciences of Ulm and Heilbronn in collaboration with industry partners [75], aims to develop a novel real time intelligent road radar system to detect and classify wild animals and other potential hazards on the roads. This will be achieved by combining micro-Doppler radar with sensors like optical and infrared cameras, together with NNs allied in machine learning systems. The project objective is to differentiate between various objects such as pedestrians, cars, bicycles, motorbikes, deer, foxes and wild boars. It aims to predict the behaviour of these objects and send warnings to drivers and road users to prevent accidents. The project also envisions deploying solar-powered roadside installations to detect hazards and communicate warnings to approaching vehicles for users without smart vehicle displays [76].

The goal of the research presented by Biswas *et al.* in [68] was to explore the use of an FMCW Millimetre Wave radar for the classification of traffic signalling motions performed by a human traffic control officer with the aim of utilising this technology in automotive contexts. The researchers gathered a total of 840 samples, encompassing 12 distinct traffic control gestures performed by 14 different participants. They employed a CNN architecture to classify these gestures achieving accuracies of approximately 93.45% and 91.67% when utilising Red, Green, and Blue (RGB) and gray-scale images, respectively. According to other performance metrics such as precision, recall and F1-score, the CNN using RGB images also achieved better performance.

2.3.2 Human Gait Classification

Micro-Doppler classification as demonstrated in previous studies [42, 77] proves particularly valuable for human gait classification due to its inherent capability to capture and discern the nuanced variations in micro-Doppler signatures resulting from diverse walking or running patterns. This unique ability enables the radar system to extract crucial information for identifying specific gait characteristics.

In a study done by Alzogaiby in [78], new functionality is added to an existing surveillance X-Band FMCW radar system to be able to automatically classify human activity based on their gait. The study presents a design that incorporates a detection mechanism, feature extraction, and a feed-forward multilayer NN classifier. Micro-Doppler signatures of the targets are extracted and a feature extraction mechanism is developed. Six features are then derived based on the periodicity of the target signature, including average velocity, micro-Doppler bandwidth, stride length and gait-amplitude ratios [78]. The NN classifier, trained only on extracted features, achieves a high accuracy of 96% from a 406 sample size set in classifying seven different classes of human activities.

In research conducted by Singh *et al.* [35], a Continuous Wave (CW) radar operating at 6.5 GHz was employed to collect data on five different human activities, including walking, pushing an object, swinging arms, walking without swinging arms, and walking with swinging one arm only. A total of 150 samples were recorded for analysis. Once the spectrograms were constructed, by using a STFT, spacial features were extracted. These include entropy (a statistical measure of randomness that characterises image texture), average intensity, variance (a measure of spread), skewness (a measure of distribution asymmetry based on the third moment), and kurtosis (based on the fourth moment of the distribution) [35]. These features, both individually and in combination, were used for training and prediction using a SVM classifier. During the validation phase, the classifier achieved an accuracy rate of 96.7%, while during testing, an accuracy greater than 93.33% was observed [35].

2.3.3 Elderly Fall Detection

Numerous valuable research efforts [32, 33, 38, 40, 44, 67, 70, 79, 80] have been conducted regarding the use of micro-Doppler radar technology in the application of fall detection systems in elderly homes for in-home health monitoring. The use of radar is a promising candidate as it is non-intrusive and appropriate for monitoring use in homes and hospitals [79].

Saho *et al.* [67] compare deep learning and gait parameter-based methods to classify elderly non-fallers (group of adults with no fall history), multiple fallers (group of adults with a history of multiple falls), middle-aged adults (50s) and elderly adults (70s) based on their gait patterns using an open motion capture gait dataset. The gait classifications were performed using a SVM with a feature vector containing 12 extracted gait parameters using techniques described in [40] along with a Gaussian kernel function selected. The deep learning method used spectrogram images as input to a Residual Network (ResNet) architecture for the CNN, as it has been proven effective for radar motion classification in previous studies. The gait parameter-based method achieved classification rates of 79% for faller/non-faller classification and 82% for 50s/70s classification [67], outperforming the deep learning approach with the corresponding accuracies of 73% and 76% respectively.

A comprehensive review is presented in [38] by Hanifi and Karsligil on other radar based fall detection studies. The main classification methods employed across these studies are the SVM, kNN and CNN, along with various feature extraction methods such as STFT spectrogram, cepstral heat-maps of spectrograms, wavelet transforms and Mel-frequency cepstral coefficients. This research concluded that feature extraction based classification methods provide high performance accuracy. An evaluation of the efficiency of each subset of features with 5 different classification techniques revealed that the SVM, LDA and NB (Naive Bayes) produced better classification results than the kNN and Decision Tree (DT).

In their study, Amran *et al.* [70] have successfully developed a fall detection system by utilising a 24 GHz radar sensor. The radar signals were sampled for a duration of 2.5 seconds, collecting a total of 270 data samples, which are then transformed into spectrograms. These spectrogram images are subsequently used as input for a CNN classifier. Impressively, the classifier achieves an average accuracy rate of 96.30%.

Research conducted by Seifert *et al.* [33] in provides valuable contribution to fall detection systems by investigating cane-assisted walks and their effect on the characteristics of radar micro-Doppler signatures. A cadence-velocity diagram along with PCA is used to extract gait features that enable the identification of walking aids and recognition of walking patterns.

Liu *et al.* [80] present a novel approach of utilising the fractional Fourier transform to enhance the micro-Doppler signature, which improves gait detection in low signal-to-noise ratio scenarios.

2.3.4 Wildlife Monitoring and Conservation Efforts

Micro-Doppler classification finds compelling applications, as indicative by literature [81], in aiding the tracking of animal movement, conducting behaviour analysis and facilitating species identification. Researchers can gain insights into animal behaviours, migratory patterns and population dynamics, thereby fostering a deeper understanding of ecosystems and supporting effective conservation efforts.

Van Eeden *et al.* [55] conducted a study that explored the use of motion capture (MOCAP) data to simulate radar data however it did not accurately represent real data and subsequently emphasized the significant effort involved in aligning the two. The use of an S-Band radar was examined as it offers foliage penetration however it compromises Doppler resolution concluding the X-Band radar might be better suited for ground target classification.

The main contribution of [55] is a Gaussian Mixture Model Hidden Markov Model (GMM-HMM) based classification approach that is presented and tested in an operational environment, with results demonstrating classification rates ranging from 75% to 90%, depending on the time on target. The GMM-HMM model seems to have better classification performance in comparison to Bilik's method [57], which is a GMM model trained with mel-cepstrum coefficients. Further analysis revealed the classification power primarily stemmed from the GMM model, rather than the HMM transition model. Notably, the research achieved a significant milestone by successfully distinguishing between different species of animals, i.e. a giraffe and zebra, which had not been previously explored.

Aldabashi *et al.* conducted a study [47] on the use of a 5.8 GHz CW radar for honeybee monitoring and behaviour classification. The radar classification system, reliant on audio-frequency feature extraction, was able to identify different near-hive behaviour such as leaving, entering and hovering. An accuracy of 93.37% was achieved when using the SVM algorithm. In addition to monitoring free flying bees, micro-Doppler signatures related to the bee wing/limb motion was detected by the radar, enabling identification of different insect species, which makes for exciting future work.

Lee *et al.* [39] proposed two algorithms for classifying human and dog movement based on micro-Doppler signals using an Ultra-Wideband (UWB) radar. The first algorithm is a two-stage SVM classifier that uses a Radial Basis Function (RBF) kernel and 16th-order Linear Predictive Coding (LPC) coefficients as feature vectors, achieving an average classification probability of 95.54%. The second algorithm is a CNN-based classifier that uses an input image of an Autoregressive (AR) spectrogram generated within a specific time period and achieved 100% classification accuracy which far outperformed the SVM-based classifier.

The CSIR have recently developed a radar system that achieves long range classification specifically tailored for environments like the Kruger National Park [74]. Animal measurements of elephant, rhino, impala, zebra, giraffe, buffalo and wildebeest, as well as humans, are captured and included in the dataset. By using a ResNet CNN architecture, the algorithm achieves 93.5% accuracy in classifying humans and labels 6% of cases as uncertain.

2.3.5 Animal Lameness Detection

A group of researchers affiliated with the University of Glasgow, UK [36, 37, 43] have been conducting investigations on the detection of lameness in farm animals by analysing radar micro-Doppler signatures. The research group employed a C-Band FMCW radar for data capture in all of their studies, which are described below.

The research conducted by Shrestha *et al.* [36] focused on exploring the detection of lameness in various farm animals, including dairy cows, sheep and horses. Four main features were extracted, namely the mean and standard deviation of the Doppler centroid and bandwidth. The Doppler centroid represents the centre of mass of the micro-Doppler signature, providing an estimate of the average Doppler component or bulk velocity of the animal body, while the bandwidth quantifies the spread of the signature around the centroid, reflecting leg motion patterns. The classification was performed using SVM and kNN classifiers, both yielding similar, promising results. Specifically, lameness in dairy cows (53 samples) was identified with an accuracy of 80%, emphasising the importance of the anterior and posterior angles due to the lameness pronunciation in the hind limbs. For sheep, a sample size of 42 yielded nearly 100% accuracy, which is attributed mainly to their mean velocity. In the case of horses, with a sample size of 162 and a three-class problem of severe, mild, and absence of lameness, an accuracy of 92% was achieved with the difference of using a cubic (instead of quadratic) kernel for the SVM.

Additionally, another study by Shrestha *et al.* [37], presenting preliminary findings on investigations in the equestrian context, demonstrated 90% accuracy in classifying horses for lameness detection by using a dataset comprising 165 samples. The study employed two classifiers: kNN with five neighbors and SVM with Gaussian radial basis functions, both of which yielded similar performance outcomes. The selected features differed from previous studies and included the mean of the Doppler centroid, the entropy of the image and the third order moment of the histogram [37], which captures pixel intensity values. Image entropy quantifies the level of randomness present in the data comprising an image, thereby defining its texture. Additionally, it signifies the distribution and concentration of energy within the image's texture. The third-order moment of the statistical histogram represents both the skewness of the data and the symmetry of the histogram's curve envelope [37], along with the variations in greyscale within the image. These features contribute valuable information for assessing and understanding the quality of the movement of a healthy horse.

In a subsequent study conducted by Fioranelli *et al.* [43], particular attention was given to cattle (51 samples) and sheep (75 samples). The feature extraction approach utilised features related to the statistical moments of the micro-Doppler signature's centroid and bandwidth, as well as its Singular Value Decomposition (SVD). To identify the most informative features, a feature ranking and selection process, based on the T-test criterion, was implemented. Through extensive experimentation, different combinations of these features were tested for accuracy, and the optimal feature set was chosen for classification. Two classifiers were used this study: the kNN with 3 neighbors and NB classifier. The results indicated that the NB classifier provided better accuracy compared to the kNN classifier, achieving approximately 82% accuracy for dairy cows and 93% accuracy for sheep. Further experimentation revealed that adjusting the duration of the STFT window to match the animal's walking kinematics had an impact on the classification performance.

2.3.6 UAV Classification

UAVs have become increasingly prevalent in various industries, ranging from military and surveillance operations to commercial and recreational use. With the rapid proliferation of UAVs, there arises a critical need to accurately detect, track, and classify these airborne objects for purposes of security, safety and efficient airspace management. This is where micro-Doppler classification can play a pivotal role and has emerged as a promising technique, as reported in various studies [31, 41, 82–84]. Micro-Doppler analysis can reveal fine grained details such as spectrum width, blade flash, blade rotation rate, blade length, number of blades and the blade tip velocity, which make it particular useful when differentiating drones from other small-sized airborne targets that exhibit distinct flight dynamics [71]. Other targets that typically fly at the same speeds and low altitudes include birds, which can trigger false alarms on radars meant for detecting UAV's. Studying the locomotion of birds and the micro-Doppler signatures can therefore be useful in discriminating against different flying objects.

Atkinson *et al.* [69] used 2 L-Band radars to obtain a fairly large dataset (greater than 10000 images) of micro-Doppler signatures from various drones, controlled birds and opportune birds, for purposes of classification. A CNN classifier using Alexnet was used to perform the binary classification between birds and drones. The classifier works by utilising the spacial features from the spectrograms, and provided good accuracy with the percentage of correct classifications at 90.1% for the test set.

In research conducted by Zhao *et al.* [85], a vision is shared regarding the use of drones for parcel delivery in the future. If allowed into legislature, the use of micro-Doppler radar to classify commercial drones carrying payloads becomes of great interest and application for air traffic monitoring in the future.

2.4 Research Gap

Micro-Doppler radar target recognition and classification is a field of study that has a wide array of applications. This section discusses some of the key gaps identified in the existing literature and research in this area.

2.4.1 Classification of Different Animal Species

Valuable research contributions have been made toward human gait classification [35,42,77,78] as well as fall detection [32, 33, 38, 40, 44, 67, 70, 79, 80] and the analysis thereof. There is a gap in literature with regard to animal species classification, which is mainly due to the challenges around obtaining such data. Many of the existing studies relating to animal classification [36, 37, 43, 55, 74] have been carried out in controlled environments, or with a limited variation of animal species. Additional research is required to investigate the feasibility of using radar for classification of multiple different animal species.

2.4.2 Comprehensive Datasets

To advance the application of deep learning classification algorithms in the micro-Doppler domain, there is a need for more accessible, comprehensive datasets that include both human and animal targets.

2.4.3 Cost-Effective Radar Systems

Many of the studies currently available have used state of the art radar systems [55,74] that are expensive and not easily accessible. Therefore, there is a need to research and evaluate the performance of classification algorithms using more cost-effective systems. Low-cost radar systems inherently face limitations in their ability to transmit, receive and process signals due to cost-driven compromises in the hardware and software resulting in weaker signal reflections and higher susceptibility to noise, directly leading to low signal-to-noise scenarios. For example, Liu *et al.* [80] developed methods using a fractional Fourier transform to improve gait detection in low signal-to-noise ratio scenarios, which could be further explored for use in applications involving low-cost systems. Furthering this research will enhance the usage and practicality of micro-Doppler radar technology.

Chapter 3

Data Collection and Pre-Processing

This section provides a high level overview of the radar system employed in this research for data collection, along with the signal processing performed for target detection. Thereafter, the logging methodology and data capture activities performed in order to compose the dataset for this research is described. MATLAB [86] is the chosen tool in order to acquire, log and process the data.

3.1 LCR System

The radar system selected for this study is the Low Cost Radar (LCR), engineered by RRS for short-range human detection and monitoring. It is a pole-mountable radar with dimensions 250 mm by 250 mm by 150 mm, and is powered via 24 V Power over Ethernet (PoE). The LCR's design prioritises portability and simple deployment, rendering it particularly advantageous over a radar such as the RSR-904 (refer to Figure 3.1) for the purposes of data collection.

(a) LCR and RSR-904 Radars

(b) LCR Employed for this Research

Fig. 3.1. Radar Deployment at Wintervogel Airstrip (left) and at RRS (right)

A high level overview of the radar system is provided in Figure 3.2. The radio frequency subsystem consists of a transmit front end and two receive front ends, each comprising a 10-element series-fed patch array antenna. The signal generated from the master oscillator is divided using a splitter. The divided signal feeds the transmit antenna and the two receiver mixers (mixer 1 and mixer 2) simultaneously in order to generate the baseband signals. The outputs from these mixers are subjected to filtering (Sensitivity Frequency Control (SFC) and anti-aliasing) and amplification (gain) which are digitally sampled and then processed in the FPGA (Field Programmable Gate Array).

In the FPGA, the signals undergo further refinement via Finite Impulse Response (FIR) filtering and decimation. Subsequently, the signal is packetised and queued in a First-In, First-Out (FIFO) buffer, and streamed via a Serial Peripheral Interface (SPI) to the Raspberry Pi. The signal data is then transmitted by using the ZeroMQ (ZMQ) messaging protocol. Logging and further detailed signal processing is performed in order to extract information about targets detected by the radar.

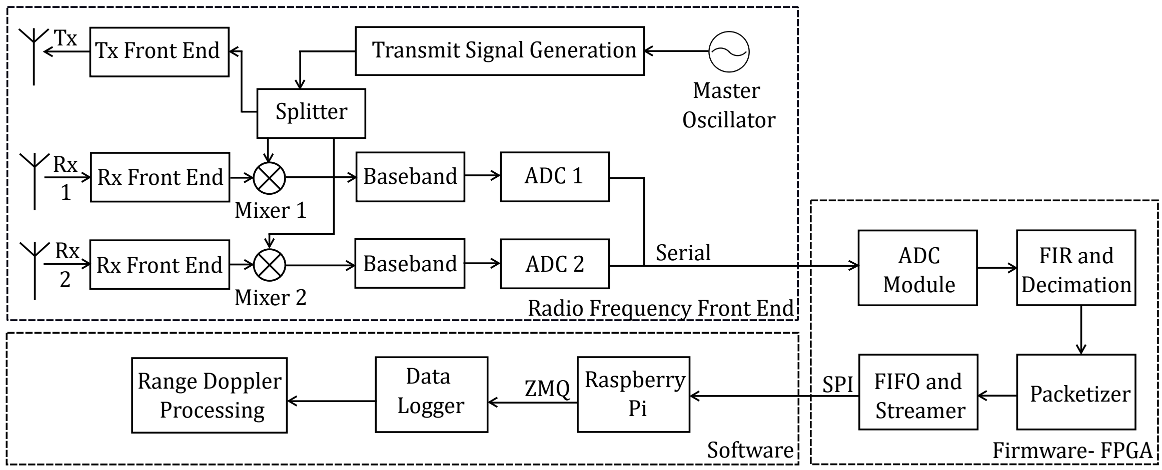


Fig. 3.2. LCR Radar Overview

Table 3.1 presents the main LCR system parameters. Refer to Appendix A.1 for further information about the LCR.

TABLE 3.1
LCR System Parameters

Type	FMCW Staring
Frequency	X-band
Maximum Detection Range	1 km
Range Resolution	1.875 m
Maximum Measurable Target Velocity	4.93 m/s
Doppler Resolution	36.56 mm/s
Horizontal Half Power Beam Width	70° Azimuth
Vertical Half Power Beam Width	20° Elevation

3.2 Pre-Processing Methodology

This section describes the process used to transform the raw Analogue to Digital Converter (ADC) data into a range-Doppler map, which is then used to detect targets. Refer to Appendix A.2 for background theory on the signal processing chain. The positional information from detected targets are subsequently used to generate the spectrograms, as described in Chapter 4.

The raw ADC data is logged into a `.log` file continuously, as soon as a burst of data is received, by using MATLAB. While logging the data, a timestamp is appended to each burst.

The log file is labelled automatically with the date, time and location of radar deployment and is created after subscribing to the ZMQ stream.

3.2.1 Data Logging and Acquisition

A single burst of data (4096 bytes) contains a header, and the real data component from channel 1 (2016 bytes, 1008 samples), followed by the real data component from channel 2 (2016 bytes, 1008 samples). Channel 1 refers to receive antenna 1 and channel 2 refers to receive antenna 2, respectively. Refer to Figure 3.3.

The raw ADC radar data is published continuously via ZMQ, therefore the data can be processed in real time simply by subscribing to the ZMQ stream. Thereafter the data is reshaped and reorganised into the format required for subsequent signal processing.

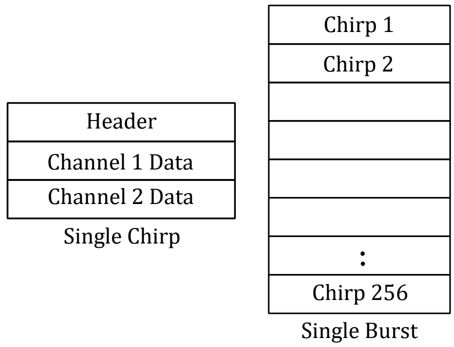


Fig. 3.3. Data Structure of a Single Burst

3.2.2 Remove DC Offset

In the signal processing framework, the first step is to remove the Direct Current (DC) offset from the data. The mean value of the data is essentially the DC offset value. Therefore, subtracting this mean value from the signal itself removes the DC offset.

3.2.3 Range FFT

A Hann [87] window is applied to the data in order to reduce spectral leakage. Thereafter, the windowed data is subjected to a one-dimensional FFT along the range axis (i.e. each of the rows contained in the burst matrix). The Range FFT is separately performed on channel 1 and channel 2 data, and the resultant matrices are added.

3.2.4 Doppler FFT

A similar process is performed for the Doppler FFT, where a window using the Hann [87] function is applied to the data. Thereafter a one-dimensional FFT along the Doppler axis (i.e. each of the columns in the burst matrix) is executed.

3.2.5 Range-Doppler Map

Once the FFT's are computed in both the range and Doppler dimensions, the magnitude is calculated to visually represent the intensity of the radar returns. The zero frequency component is shifted to the centre of the spectrum in order to facilitate the interpretation of negative and positive frequencies which represent different directions of motion.

3.2.6 Constant False Alarm Rate

The Constant False Alarm Rate (CFAR) algorithm chosen is the Cell-Averaging Constant False Alarm Rate (CA-CFAR) due to its simplicity, ease of implementation and effectiveness in homogenous environments [88]. It is implemented based on Jen's algorithm in [89], for real time data capture and validation. The algorithm aids the range-Doppler map in confirming the correct setup of the radar and facilitates in the visualization of targets.

The CA-CFAR algorithm dynamically adjusts the detection threshold based on local conditions to maintain a constant false alarm rate [89]. A CA-CFAR function is designed to take the desired probability of false alarm (P_{fa}), the total number of training cells (N), the number of guard cells and the input signal as parameters.

The selection of cells is influenced primarily by the radar's range resolution of 1.8 m and a Doppler resolution of 36.56 mm/s and are listed in Table 3.2. The P_{fa} parameter is empirically determined, with values ranging between 10^{-6} and 10^{-5} providing a practical false alarm rate. This range has been shown to be effective in achieving a balance between detection and false alarm control in various studies [88–91].

The CA-CFAR function [89] returns the calculated detection threshold. It is implemented using the following steps:

- **Threshold Multiplier Calculation:** A threshold multiplier is computed using N and P_{fa}.
- **Signal Preparation:** The length of the input signal is determined and indices for a sliding window, comprising training cells, guard cells and a cell under test (CUT) are generated to traverse the entire signal.
- **Training Cells Selection:** From the sliding window, indices corresponding to guard cells and the CUT are excluded, isolating the training cells used to estimate the noise level.
- **Noise Level Estimation:** The power values of the training cells are averaged for each window position to estimate the local noise level.
- **Threshold Calculation:** The noise estimate is scaled by the previously calculated threshold multiplier to derive the detection threshold for each position of the CFAR window across the signal.

Integration in Main Processing Chain:

- **Detection Loop for Range:** A for-loop iterates over each row of the range-Doppler map, applying the CA-CFAR algorithm to compute detection thresholds for the range dimension.
- **Detection Loop for Doppler:** Similar to the range processing, a for-loop iterates over each column of the range-Doppler map, utilising the CA-CFAR algorithm to determine detection thresholds for the Doppler dimension.

The output thresholds from the CFAR detection loops are used to create binary detection matrices by comparing these thresholds with the range-Doppler map (see Figure 3.4(a)). These binary matrices, representing detection outcomes in range (refer to Figure 3.4(b)) and Doppler (refer to Figure 3.4(c)) dimensions, are then used to filter the range-Doppler map, isolating detected targets.

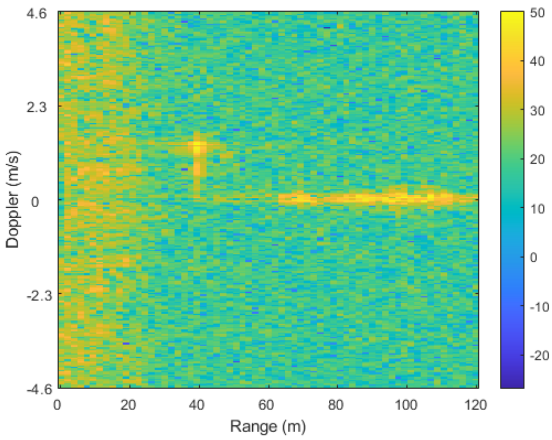
The final detection matrix is obtained by element-wise multiplication of the binary detection matrices for range and Doppler dimensions, providing a resultant composite detection matrix representing the potential target location, as illustrated in Figure 3.4(d).

Refer to Appendix A.3 for more information on the CA-CFAR algorithm.

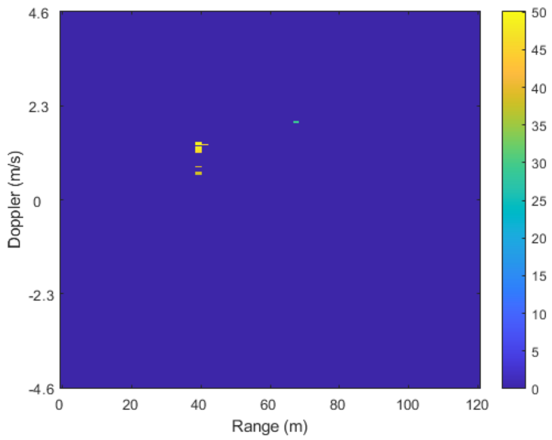
TABLE 3.2

CFAR Parameters

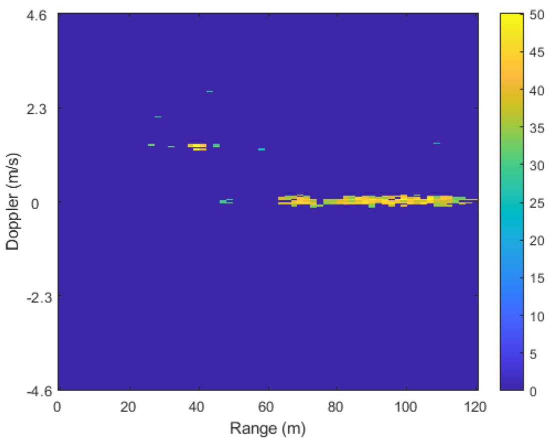
Desired P_{fa}	10^{-5}
Training cells for range CFAR	4
Training cells for Doppler CFAR	8
Guard cells for range CFAR	6
Guard cells for Doppler CFAR	10



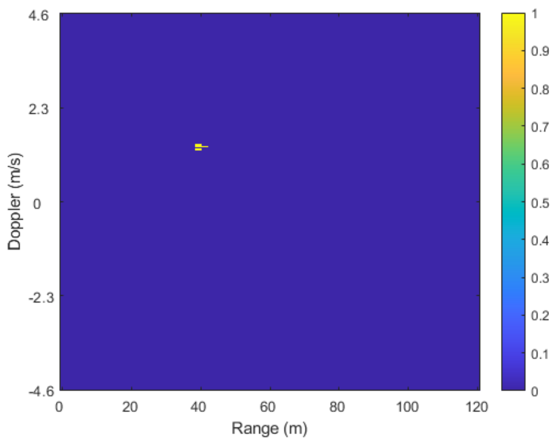
(a) Range-Doppler



(b) Range CA-CFAR



(c) Doppler CA-CFAR



(d) Range and Doppler CA-CFAR

Fig. 3.4. Range-Doppler and CA-CFAR Maps Illustrating a Target

3.3 Data Acquisition- Human Database

This section outlines the data capture activities undertaken to compile the dataset used in this research.

The LCR is mounted on a pole with 1.2 m of height, supported by a tripod with approximately 0.5 m of height, see Figure 3.1(b). An IP camera is mounted to the pole below the radar unit in order to facilitate synchronisation of radar data to the video footage, where possible. These videos ensure that the recorded radar data can be correlated with the visual content during playback at a later time.

There are 11 different individuals recorded in the human database. A breakdown is provided in Table 3.3.

TABLE 3.3

DISTRIBUTION OF INDIVIDUALS BY LOCATION

Location	**Count**
RRS Rooftop	3
WinterVogel Airstrip	5
RRS Parking Lot	2
Paardevlei Old Industrial Area	5

The approximate speeds of various types of motion are categorised in Table 3.4.

TABLE 3.4

SPEED OF MOTION CORRESPONDING TO DIFFERENT ACTIONS

Walking	1 m/s
Jogging	2 to 3 m/s
Running	4 to 6 m/s

All forms of motion are characterised as piece-wise and continuous, with movement patterns predominantly in a straight line both toward and away from the radar.

3.3.1 RRS Rooftop

During the radar's development, the radar was situated on the rooftop at RRS (Technopark, Stellenbosch) because this location offered optimal visibility of targets, due to the open field area in front of the building and the adjacent road.

Figure C.1, in Appendix C, depicts the line of sight of the radar as well as a satellite layer screenshot of the area, captured from Google Maps.

Data recordings are taken of individuals walking between the boundary fence, at 30 m from the radar, to approximately 120 m away.

There are 3 person targets contributing to the database that contain individuals walking alone and pairs of individuals walking side-by-side, all amongst vegetation clutter such as trees and bushes.

3.3.2 Wintervogel Airstrip

At the Airstrip (Wintervogel Farm, Malmesbury), refer to Figure 3.1(a), data recordings are taken of person targets between the radar and a tree landmark with range 160 m.

There are 5 person targets, and the database contains the following scenarios:

- Individuals walking, jogging and running alone.
- Group of 4 individuals walking close together.
- Pair of individuals walking side-by-side.
- Individuals walking with a paintball gun.
- Individuals walking with swinging arm motions.
- Individuals walking without hand motions by placing hands firmly to the side.

3.3.3 RRS Parking Lot

One individual is the focus of data capture activities in the RRS Parking Lot. This individual engages in various movements, starting from the radar and travelling approximately 70 m to the boundary fence and back, as illustrated in Figures C.2 and C.3, in Appendix C.

The contributions to the database contain the following scenarios:

- Walking.
- Running.
- Walking with arms swinging.
- Walking with both hands in pockets.
- Walking with one hand in a pocket.
- Standing and waving one hand, and then both hands.
- Walking while holding a hammer.
- Standing in the same position while ducking up and down.

3.4 Data Acquisition- Animal Database

In this research, dogs, horses and cows are selected as the primary subjects. These animals belong to distinct size categories and exhibit unique movement patterns. They therefore serve as proxies within their respective size groups for possibly understanding a wider range of animal species with comparable body size and similar gait characteristics.

Dogs, are selected to represent smaller-sized animals presenting an opportunity to explore a variety of gait patterns similar to that of animals such as foxes, jackals and small deer species.

Horses are chosen to embody the category of large animals. Animals, such as zebras and antelopes, share comparable body structures, making the domesticated horses suitable subjects for studying the movement dynamics in this category.

Cows, represent a group of medium to large sized animals that are heavier and live a more sedentary lifestyle when compared with dogs and horses. This category includes animals like buffalo and bison, which share similar build size and herd dynamics.

3.4.1 Dogs

The animal database contains movements from two different canine subjects. Dog 1 is a trained golden retriever, while Dog 2 is a mixed breed of Border Collie and Staffordshire Bull

Terrier. Radar data from these dogs are collected at the Paardevlei Old Industrial Area, Somerset West. Refer to Figures C.4, C.5, and C.6 in Appendix C.

In each experimental scenario, one individual would position themselves at the end of the path (approximately 80 m), while another would stand in front of the radar equipment. Upon calling the dog, the dog would walk or run in the space between the two individuals, allowing the radar to exclusively capture the motion of the dog.

The owners of the dogs would walk and/or run from the radar equipment to the end of the path and back when required. This provided an opportunity to expand the database by including data from 5 individuals.

3.4.2 Horses

Radar data from the horse targets are collected at Pinto Stables, Joostenberg, Kraaifontein.

The stables are visited on multiple different occasions in order to gather radar data from the horses.

The first visit to the stables was to assess whether the radar is capable of successfully recording the data of a horse target. On the day, there were horse riding lessons taking place in a sand-filled enclosure, 80 m long and 60 m wide. The radar gets deployed 10 m in front of the enclosure and data is recorded while the students participate in their horse riding lessons.

This data capture activity is sufficient to conclude the following:

1. The radar can successfully identify the horse as a target.
2. The different speeds, i.e. walking, trotting and galloping, at which the horses move are also correctly captured and reflected in the range-Doppler maps.

The data from this experiment is not incorporated in the database used in the research as the dataset contains human targets that cannot be separated in the subsequent signal processing steps.

The second encounter at the stables involved monitoring and observing a herd of horses within a grass-filled enclosure. The radar is positioned at a distance of 30 m from the enclosures perimeter, as depicted in Figure C.7, in Appendix C. Data recorded from the horses primarily include grazing activities, accompanied by slow movement. Notably, the movement patterns are characterised by intermittent head and tail motions resulting in discrete segments of activity.

The third visit to the stables presented a great data recording opportunity. The stallion, named Seun, is placed and observed in a separate grazing enclosure. This isolation provided an ideal environment to record data of Seun's movement patterns, minimising interference from any other targets. The radar is placed approximately 10 m away from the enclosure boundary, as shown in Figure C.8, in Appendix C.

3.4.3 Cows

Radar data of the cows is collected at Yonder Hill, Raithby, Somerset West.

Yonder Hill farm features grazing lands laid out on an inclined plane, affording the radar with strategic placement at the most elevated vantage point to maximize the radar's ability to detect and record cattle across various range bins. The radar is positioned on slightly elevated terrain, granting the radar an additional meter of height.

The radar captures a range of movements from a diverse cohort of cattle, including dairy cows and ankole breeds. It is observed that the cattle frequently move their tails when grazing and exhibit collective movement patterns, often migrating in groups to new grazing areas. While most of the data is captured from groups of cattle, an occasional opportunity to record a single cow walking independently did fortunately arise. Refer to Figure C.9 in Appendix C for a depiction of the experimental setup.

Chapter 4

Data Preparation

Following the data collection, this chapter addresses the preparation of data for spectrogram generation and analysis, which is critical for the classification stage. It covers details on data verification, target extraction, manual tracking procedures, spectrogram creation and compilation of the spectrogram database.

4.1 Data Verification

The data is extracted from the log file in the same way that it is extracted from the ZMQ stream (see Section 3.2), with the difference being that the log file contains a timestamp that is appended to each burst.

The video footage obtained during the data collection process is analysed. Each video frame includes an embedded timestamp which is then cross-referenced with the corresponding timestamps recorded in the log file. This cross-referencing is important for verifying the congruence between the observed target's position and movements within the video and their representations on the range-Doppler and detection maps, ensuring the accuracy of the data captured.

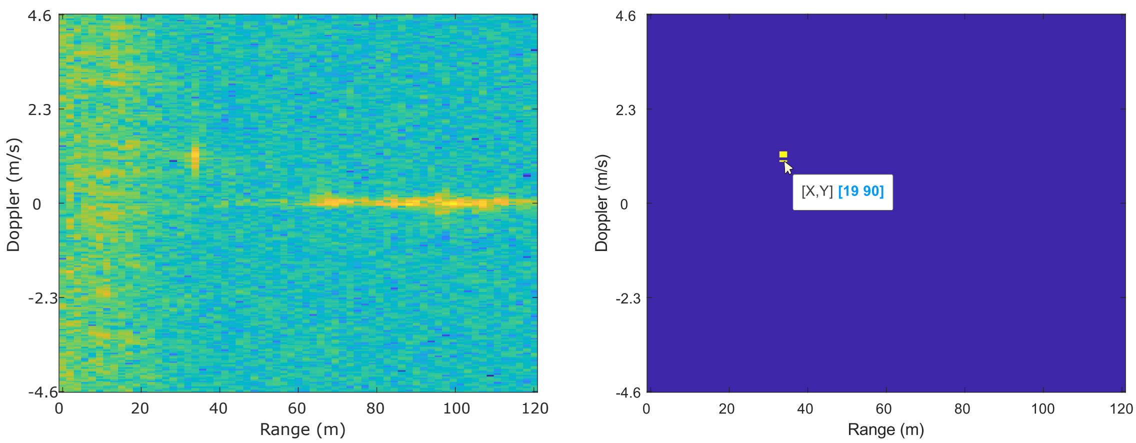
4.2 Target Extraction in Range

Once a target is detected (see Section 3.2), the subsequent step in the processing sequence involves tracking the target. Tracking methodologies are diverse and complex, constituting a domain that falls beyond the scope of this research. Considering the data is primarily collected in controlled settings, the tracking of a target is conducted manually. This involves the identification and recording of the target's position as it shifts from one burst to another.

The manual target location procedure is outlined as follows:

- The target's position is identified on the range-Doppler or CA-CFAR detection map and is captured through a physical mouse click. Refer to Figure 4.1.
- At that specific time, the range bin housing the target is extracted from the coordinate, appended and saved into a vector.
- The identified range bins for each target across different log files are stored in a MATLAB function file.
- This file maps the log file names to their corresponding target range bins through a switch-case structure.
- Each case in the structure corresponds to a unique log file name, with an associated vector of range bins that represent the manually tracked target's location over time.

For instance, when a target is detected, as shown in Figure 4.1, a mouse click on the detection map populates the file with the X coordinate representing the range of the target.



(a) Range-Doppler Map (b) CA-CFAR Detection Map

Fig. 4.1. Person Target at Approximately 35 m Range (Range Bin 19)

```
switch log_file_name
    case PersonWalkingTowardRadar  % Person walking in range bins 6 through 10
        Range_bin = [6,6,6,6,7,7,7,8,8,9,9,10,10];
    case PersonRunningAwayfromRadar % Person running in range bins 15 through 30
        Range_bin = [15,17,19,21,23,25,27,29,30];
end
```

4.3 Spectrogram Generation

In order to do the spectrogram analysis, the process begins by reading the raw ADC data continuously, in bursts, from the log file. This data is then reshaped and partitioned according to the relevant channels. Each channel's data undergoes a range FFT, after which the resultant frequency domain data from channels 1 and 2 are combined. Thereafter, the signals corresponding to specific range bins indicative of target presence is extracted (as described in Section 4.2). These signals are then concatenated over multiple bursts to build the input signal for the STFT analysis.

The STFT is used to convert the time-domain signal into its time varying frequency domain representation [22].

The STFT computation is carried out by dividing the input signal into overlapping segments, based on the length of the input signal, the window length, and the overlap between windows. Each segment is multiplied by a windowing function and is normalised by its sum. This windowed and normalised segment undergoes a Fourier transform, with the number of DFT points specified. The `fftshift` operation is applied to centre the zero frequency component in the middle of the spectrum. The resulting spectra for each segment are concatenated column-wise to form the spectrogram matrix. The spectrogram matrix represents the time-varying frequency content of the signal, with each column corresponding to a time frame and each row to a frequency bin.

The magnitude of the STFT matrix is then computed to emphasise the energy present. Thereafter, the matrix is logarithmically scaled and normalised by using the formula $20\log_{10}(\texttt{magnitudeSTFT}/\texttt{maxMagnitudeSTFT})$, where `maxMagnitudeSTFT` is a normalization factor that aligns with the reference level of the spectrogram data (Refer to Appendix A.4). The transformation of amplitude values into a decibel scale is a standard practice in signal processing, as it aids in highlighting the relative differences in signal strength across the spectrogram. In general, this is depicted by a colour bar; however in this research, it will specifically be shown on the right side of all spectrogram images, as illustrated in Figure 4.2.

The STFT can be constructed through the adjustment of specific parameters, as listed below [92]. These modifications allow for varying levels of continuity and resolution in the spectrogram representation.

- **Burst Count Interval:** The interval over which the signal is concatenated, correlating directly to the number of bursts.
- **Window Length:** Determines the segment of the signal to be analysed in each step.
- **Overlap:** Number of samples between consecutive windows.
- **Number of DFT points:** Defines the frequency resolution of the Fourier transform.
- **Window Function:** The windowing function to apply to each segment.
- **Dynamic Range Scale:** Set in order to compress the wide range of the spectrogram's magnitudes into a more manageable scale, facilitating better visual analysis.

Refer to Section 6.1.1 for further detail on the spectrogram parameters and Appendix A.4 for the source code.

4.4 Spectrogram Database

The spectrogram matrix consists of `nRows` rows (Doppler Frequency) and `nCols` columns (Time). The spectrogram is flattened into a one-dimensional vector that is `nRows` $\times$ `nCols` components long for data handling simplicity.

The spectrograms are generated and classified manually with aid of video footage. Based on the button selection of human, animal or noise (See Figure 4.2) a `.mat` file is populated in a structure (collection of variables), with X as the spectrogram data flattened, and Y as the class label. If human, the class label is the number 1 and, if animal, the class label is 2. The `.mat` files are categorised based on the date and venue of the set.

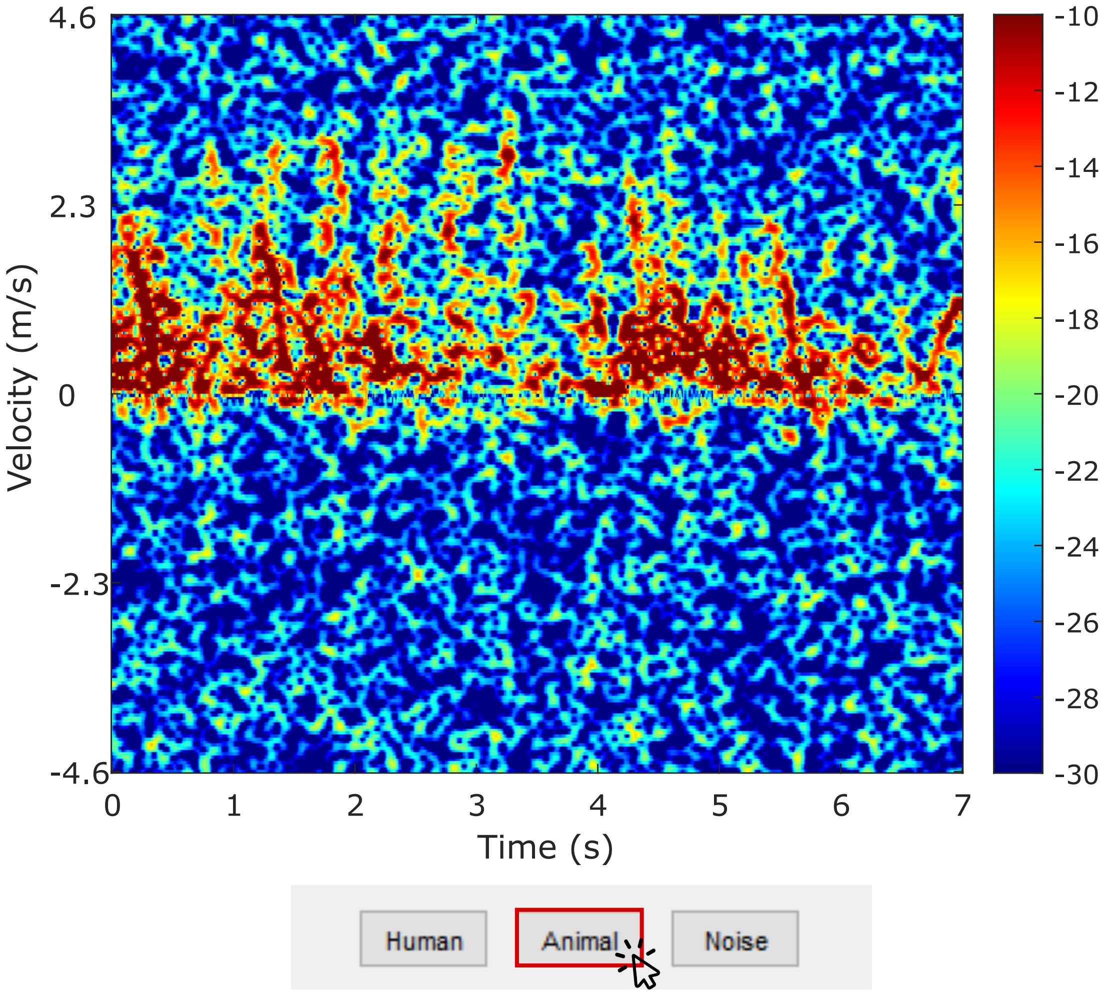


Fig. 4.2. Creating Spectrogram Database: Animal (Herd of Horses) Target

Chapter 5

Classification Algorithm Design

The initial processing and generation of micro-Doppler spectrograms is conducted in MATLAB, however, limitations are encountered, in terms of scalability and flexibility, with the datasets in the classification stage. Consequently, this limitation guided the decision towards adopting Python for the machine learning pipelines, with particular emphasis on leveraging the Scikit-learn [93], Keras [94] and TensorFlow [95] frameworks.

The classification models selected for this research are the SVM, kNN, RF and CNN. This chapter outlines the steps involved in preparing data for the classification models, designing and developing algorithms, optimizing parameters and evaluating model performance.

5.1 Data Augmentation

The full data acquisition process was hampered due to hardware degradation of the LCR. In order to complete the research, data augmentation methods were used in order to artificially increase the volume of the dataset. Data augmentation techniques are typically employed when the quality or quantity of the initial training set is insufficient to train a model [96].

The process of data augmentation involves generating new data points from the existing dataset [97] through various techniques, such as modifying the signal attributes and applying noise.

In this research, data augmentation is achieved by using the following methods:

- **Window Functions Applied in STFT Segmentation:** The STFT process is refined by employing a variety of window functions to segment the signal.
  These include:
  - Hann [87]
  - Chebyshev [98]
  - Blackman [99]
  - Hamming [100]
  - Kaiser [101]
- **Spectrogram Generation using Channel-Specific Data:** Data is processed from separate and combined channel data, specifically:
  - Channel 1 only
  - Channel 2 only
  - Combined data from Channel 1 and Channel 2
- **Larger Targets - Range Bin Expansion:** The process of determining the appropriate range bin for a target involves selecting the bin that exhibits the strongest signal (as described in Section 4.2). For larger targets, such as horses and cows, the strongest signal may span multiple bins (typically 2-3 bins). In such cases, the bins immediately preceding and following the strongest bin are also considered. These additional bins are incorporated into the analysis by adding their data after applying the STFT to generate the spectrogram. This method alters the spectrogram appearance when compared to a spectrogram generated from a single bin. This approach is specifically suitable for scenarios involving large targets where the presence of only a single target is assured, such as the data recording conducted with the stallion (refer to Section 3.4.2).

The selection of window functions introduces unique characteristics, particularly with respect to sidelobe behaviour and frequency resolution. For certain datasets, the raw ADC data from each channel is processed individually generating spectrograms for each channel separately as well as for both channels combined (by adding their signals).

The integration of the above-mentioned methodologies effectively increases the initial dataset size to enable more robust model training and evaluation.

5.2 Data Split

The `train_test_split` function, from scikit-learn's model selection module [102], is used to partition the dataset into random training and testing subsets: 80% is allocated for training the model while 20% is reserved for testing.

5.3 Principal Component Analysis

PCA is achieved through the `PCA` model in the scikit-learn's decomposition module [103], which employs SVD for linear dimensionality reduction.

Before applying PCA, the PCA model is fitted to the spectrogram data. Subsequently, the data is transformed using the fitted PCA model, reducing its dimensionality. Thereafter, the cumulative variance explained by the components is calculated and the smallest number of components required to explain 99% of the total variance is identified, as shown below:

```
pca = PCA()
pca.fit(data)
reduced_data = pca.transform(data)
cumulative_variance = np.cumsum(pca.explained_variance_ratio_)
num_components = np.where(cumulative_variance >= 0.99)[0][0] + 1
```

5.4 Data Preparation for Classification Models

The spectrogram data, stored as a one-dimensional vector, is directly used to train the SVM, kNN and RF classifiers. For the CNN, however, this data is reshaped back into its original image format (refer to Section 5.9.1).

After performing PCA on the datasets, both the PCA-reduced dataset and the full original dataset are divided using the 80/20 split as previously mentioned. For the CNN, the PCA-processed data is restructured to mimic its original image format, though this restructuring is not straightforward or easily visualised. The original spectrogram aspect ratio of `nRows` $\times$ `nCols` is maintained as closely as possible.

5.5 Parameter Optimization

Throughout the research and development of the various classification models, the selection of optimal parameters emerged as an important factor influencing both model performance and classification accuracy.

In order to find these parameters, `GridSearchCV` from the scikit-learn library is used. It is designed to methodically search through a specified parameter grid of potential combinations by cross-validating to determine which set results in the best performance [104].

5.6 Support Vector Machine

The Support Vector Classifier, `SVC` [105], is an implementation of the SVM by the scikit-learn library and is used to develop the SVM classification.

The parameters considered for optimisation in the SVM model are as follows [105]:

- **Kernel:** The choice of kernel, which can be 'rbf', 'linear', 'sigmoid' or 'polynomial' and affects the decision boundary's shape.
- **The regularisation parameter, C:** C is common to all SVM kernels and regulates the trade-off between achieving a larger margin with more training misclassification's and a narrower margin with fewer errors by setting the penalty for incorrectly classified training data [106].
- **Gamma:** The gamma parameter is used with non-linear kernels and it influences the distribution of points after mapping into a new dimension.

The SVM parameter grid is defined to search across the following parameter set:

```
param_grid = {'C':[0.1,1,10],
        'gamma': [1,0.1,0.01],
        'kernel': ['rbf', 'linear', 'sigmoid', 'poly']}
grid = GridSearchCV(SVC(), param_grid, cv= 3, scoring= 'accuracy', verbose= 3)
```

When armed with the parameters in order to maximise the accuracy outcome, the training dataset can be fitted to the SVM model as shown in the following code listing:

```
model = SVC(C=0.1,kernel='linear')
model.fit(train_data,train_label)
```

5.7 k-Nearest Neighbors

In order to perform the kNN classification, the `kNeighborsClassifier` algorithm [107], provided by the scikit-learn library, is used.

In the context of the kNN algorithm, k refers to the number of nearest neighbors that the algorithm considers when making a prediction. The k parameter fundamentally determines the kNN's behaviour, directly influencing the balance between bias and variance in the model [107]. Adjusting k can significantly impact the model performance, more so than other parameters such as distance and weighting.

The kNN algorithm's implementation is initialised with default parameters, including the Euclidean as the distance metric and uniform weights, which are selected based on their effectiveness across a variety of common scenarios.

The k parameter is optimised using the `GridSearchCV`, exploring a range from 1 to 30:

```
param_grid = dict(n_neighbors=list(range(1, 31)))
grid = GridSearchCV(knn, param_grid, cv= 10, scoring= 'accuracy', verbose= 3)
```

The `kNeighborsClassifier` can be subsequently initialised with the k value result and the model can be trained using the training dataset.

5.8 Random Forests

The `RandomForestClassifier` [50] class available in the scikit-learn library is employed to implement the RF algorithm.

Several key parameters for the RF Classifier can affect the model's performance. These parameters are listed below [50, 108]:

- **Number of Estimators:** This parameter refers to the number of trees in the forest [50]. A larger number of estimators generally improves the model's performance, however, this may also increase the computational cost and time, thereby plateauing the effect.
- **Maximum Depth:** The maximum number of levels of each tree [108]. This contributes to the complexity of the decisions being made by the model [50, 108].
- **Minimum Samples Split:** This parameter determines the minimum number of samples required to split an internal node [50].

- **Minimum Samples Leaf:** Setting the minimum number of samples required to be at a leaf node affects the granularity of the classification decisions [50]. This ensures that each leaf has a sufficient number of samples, thereby stabilising the model's predictions.

- **Maximum Features:** This parameter controls the number of features to consider at every split [108] when looking for the best split at each node [50]. By limiting the number of features, it introduces randomness into the model, which can lead to better model performance.

The RF parameter grid [108] of combinations to consider is listed below:

```
param_grid = {'n_estimators': [100, 200, 300],
        'max_depth': [10, 20, 30, 40],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4],
        'max_features': ['sqrt', 'log2']}
grid_search = GridSearchCV(rf, param_grid, cv= 3, scoring= 'accuracy', verbose= 3)
```

Once the optimal parameters have been calculated, the RF classifier should be initialised and can be trained with the training dataset.

5.9 Convolutional Neural Networks

TensorFlow [95] is an open-source machine learning and deep learning framework known for its powerful computational capabilities, developed by the Google Brain team [109]. Keras is an open-source high-level neural networks Application Programmable Interface (API) written in Python [94]. Keras is tightly integrated into TensorFlow and is the preferred default high-level API [109], due to its simplicity in model architecture design. The integration of Keras into TensorFlow provides a simple workflow for designing, training and evaluating CNNs and is therefore the chosen framework for the CNN development in this research.

5.9.1 Data Preparation

The flattened data arrays are restructured back into their original spectrogram image format, as CNNs are inherently designed to process image data [64]. Thereafter, the training and test data sets are wrapped into TensorFlow's dataset objects and grouped into smaller "batches" that the model processes sequentially during training [110]. This approach assists with the memory management in the training stage.

5.9.2 Model Design

The CNN model is defined by TensorFlow's simplest model, i.e. the Sequential model. It is a linear stack of layers with one input tensor and one output tensor [111].

The model consists of two convolutional layers. The first layer applies 32 filters, while the second layer applies 64 filters. Both layers use a 3x3 kernel size [112] and incorporate the Rectified Linear Unit (ReLU) activation function. These layers are responsible for extracting high level features from the input images through learnable filters [113], with ReLU adding non-linearity to the model [114].

A 3x3 kernel size is chosen because it effectively captures local features in an image when applying convolutions with a few pixels at a time [115]. This small kernel size allows for weight sharing across different regions of the image, reducing the number of parameters and thus lowering the computational cost compared to larger filters or fully connected layers [115].

Following each convolutional layer, a max pooling filter with a 2x2 window is applied. This reduces the dimensions of the output from the previous layer [65, 112], thus reducing the computational load. A 2x2 pooling size is commonly used in max pooling layers, where the filter selects the pixel with the maximum value [116], thereby reducing the size of the input by a factor of 2.

A fully connected or flattening layer [112] is used to convert the two-dimensional feature maps into a one-dimensional feature vector, making it possible to connect the convolutional part of the model with dense layers [113].

A dense layer with 128 input units [117] and ReLU activation is followed by a dropout layer with a rate of 0.5 [118] to prevent overfitting. Dropout achieves this by randomly ignoring certain neurons during training, as they are "dropped out" [118]. This reduces complex co-adaptations among neurons, thereby enhancing the model's generalisation ability [118].

The final dense layer uses a softmax activation function [112] and outputs a categorical probability distribution over the target classes [112].

The CNN model architecture is illustrated in Figure 5.1 by making use of an open-source online tool called Netron [119].

5.9.3 Compiling and Training

The model is compiled with the Adam optimiser [120] and sparse categorical cross entropy [121] as the loss function (typically used for multi-class classification tasks) [122].

The model undergoes training over 10 epochs with the specified training and validation datasets.

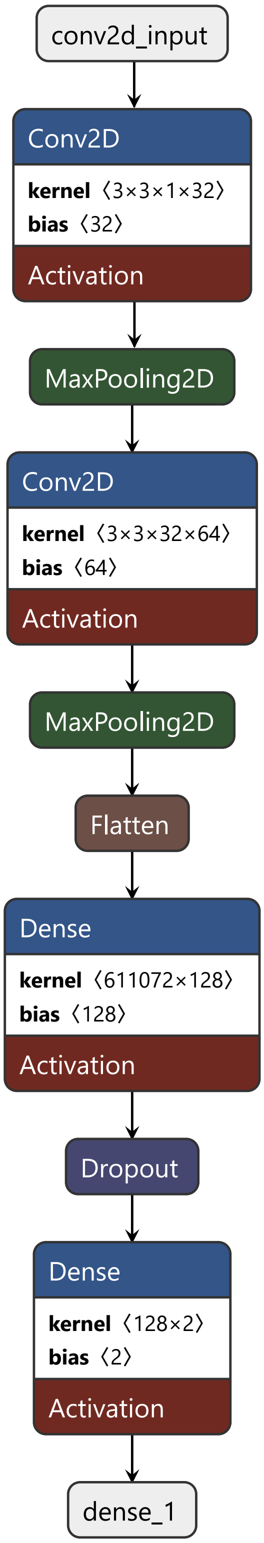


Fig. 5.1. CNN Model Architecture

5.10 Model Evaluation

5.10.1 Accuracy Assessment

Accuracy, defined as the ratio of correctly predicted observations to the total observations [123], is a primary metric for assessing model performance. To achieve this, the sklearn.metrics [124] `score` method is used for the SVM, kNN and RF models, whereas for the CNN model, the built-in `evaluate` function from the Keras API is used.

In addition, a confusion matrix is derived for each of the models using the sklearn.metrics [124] `confusion_matrix` function, once a prediction is made. Confusion matrices are used to represent the counts from the predicted and actual values.

5.10.2 Classification Report

The sklearn.metrics `classification_report` is used to provide a report containing metrics, such as the precision, recall, support and F1-score metrics for each class. These metrics [125] are elaborated on below:

- **Precision:** Defined as the ratio $\frac{tp}{tp+fp}$, where tp is the number of true positives and fp is the number of false positives [125]. Precision looks at the ratio of true positives to all positive predictions (both true and false positives) to assess how many of the positively predicted cases were actually positive.
- **Recall:** Defined as the ratio $\frac{tp}{tp+fn}$, where tp is the number of true positives and fn is the number of false negatives [125]. Recall examines the ratio of true positives to all actual positives (true positives and false negatives) to evaluate how well the model identifies all relevant instances.
- **F1-score:** Combines precision and recall into a single metric that balances the two, defined as the harmonic mean of precision and recall: F1 = 2 x $\frac{Precision x Recall}{Precision+Recall}$. The F1-score is useful in situations where one wants to manage the trade-offs between precision and recall effectively [125].
- **Support:** Support quantifies the actual number of instances for each class within the test dataset, providing insight into the distribution of classes in the dataset.

While accuracy is a straightforward metric for evaluating model performance, it can provide a biased result in imbalanced datasets where one class dominates. In such cases, metrics like precision, recall and the F1-score are valuable for capturing the model's effectiveness in identifying minority class instances.

Chapter 6

Experimentation

This chapter details the selection of STFT parameters for generating spectrograms. The micro-Doppler spectrogram patterns resulting from various targets are presented and analysed. The results of applying data augmentation techniques are provided. Additionally, the number of principal components selected and the parameters obtained from the `GridSearchCV` optimization for each classifier are discussed.

6.1 Micro-Doppler Spectrogram Data

6.1.1 Spectrogram Parameters

The parameters listed in Table 6.1 are carefully selected in order to balance the time-frequency resolution for the STFT computation within a 7 second time frame.

TABLE 6.1

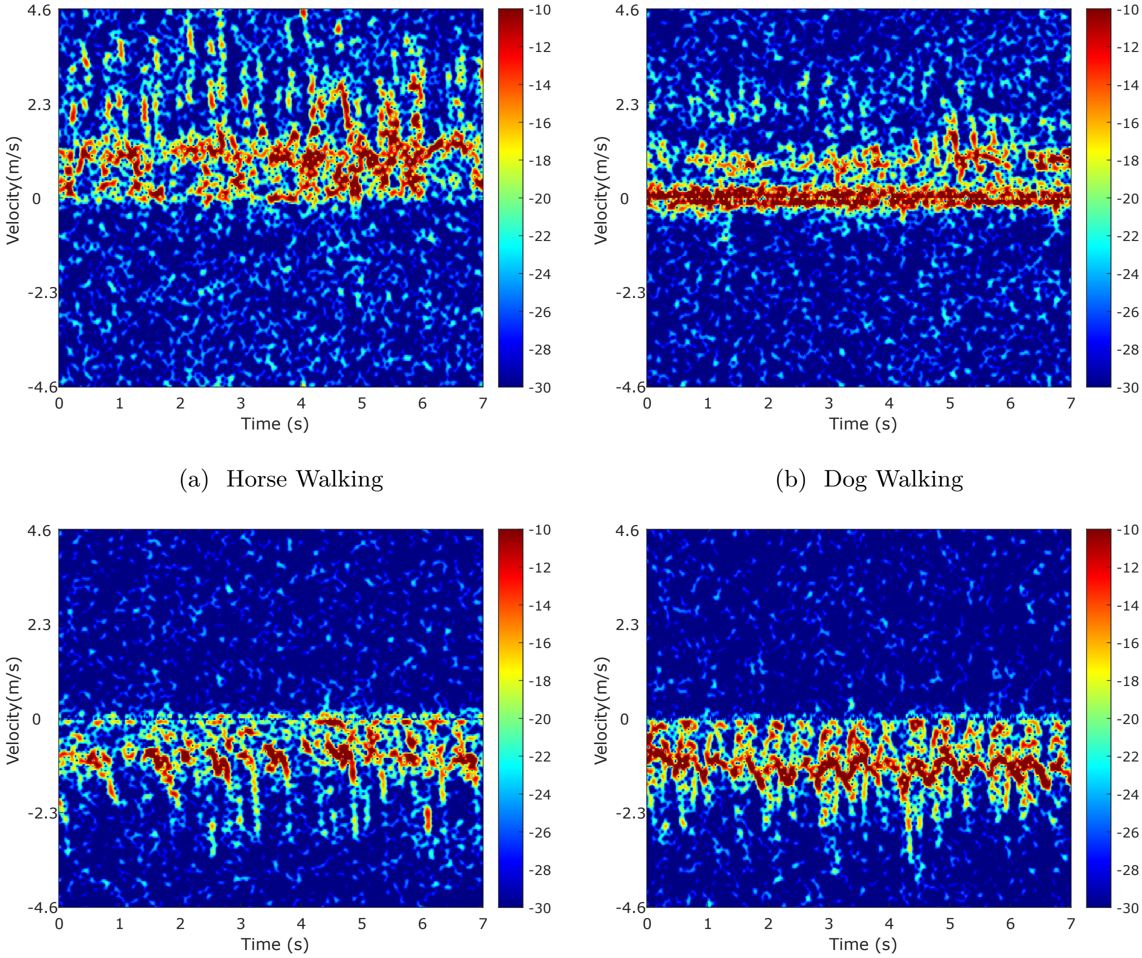
STFT Parameters

Window Size	128
Overlap	8
No. of DFT points	256

The window size is chosen to ensure sufficient frequency resolution for distinguishing Doppler shifts characteristic of the targets while ensuring the windowed signal remains approximately stationary within each analysis window [24]. The longer window size provides the necessary frequency detail without compromising on temporal resolution.

6.1.2 Analysis of Micro-Doppler Patterns

The spectrogram images of the various target subjects are illustrated in Figure 6.1. Additional representations and variations of these targets can be found in Appendix D.



(a) Horse Walking

(b) Dog Walking

(c) Cow Walking

(d) Person Walking

Fig. 6.1. Micro-Doppler Patterns of Various Targets Walking

- **Horse Walking:** The main body of the horse walking maintains a steady motion, evident in Figure 6.1(a) as a consistent trace with minor frequency fluctuations. Occasionally, the horse's head movement to the side may contribute to some of the micro-Doppler components around the main body motion. The tail movement, generates a distinct, transient feature between 2.3 m/s and 4.5 m/s, appearing faster and more erratic when compared to the body's movement.

- **Dog Walking:** The micro-Doppler signature of a dog walking (see Figure 6.1(b)) reveals a pattern characterised by smaller, more frequent oscillations when compared to the larger animals. This is likely due to the quick, short strides which correspond to the dogs movement. No distinctive tail movement is observed in the micro-Doppler patterns of the dogs walking due to their relatively small body size and even smaller tails.

- **Cow Walking:** Cows typically move at a slower and more deliberate pace. In the spectrogram shown in Figure 6.1(c), broader non-periodic spectral components are observed near 1 m/s, with occasional sharp peaks at higher velocities which likely could be attributed to tail movements.

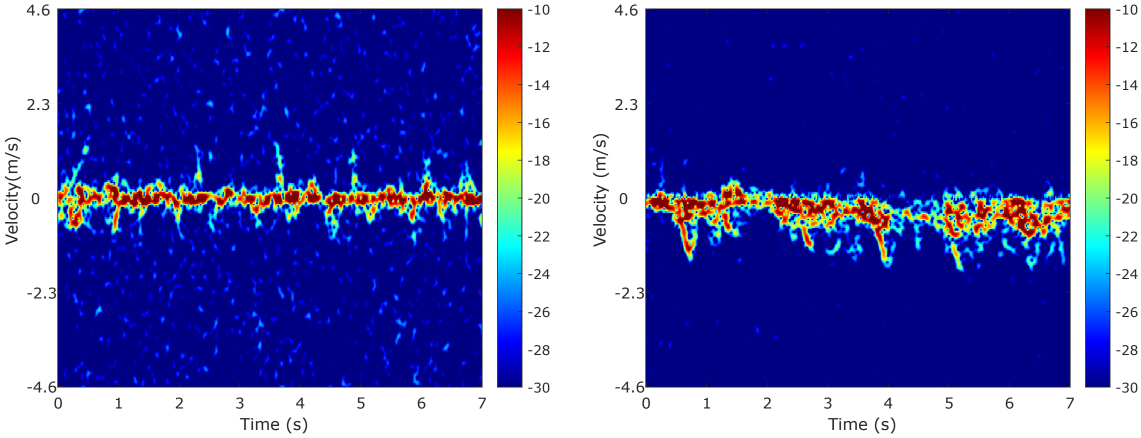
- **Person Walking:** In the micro-Doppler signature generated by a person walking (refer to Figure 6.1(d)), the repetitive patterns of the alternate movements of the legs and arms are clearly evident, and the main body mass shows a sinusoidal pattern around 1 m/s.

Both horses and cows have notable tail movements while walking, which are evident in the patterns of their micro-Doppler spectrograms. These movements are among the most periodic features observed.

The motion of quadrupeds is fundamentally different from bipedal motion, particularly in terms of the predictability and pattern of leg movements. In quadrupeds, the legs touching the ground follow a less predictable sequence when compared to the gait of humans. This is evident in the micro-Doppler signatures, where the spectral lines that are associated with leg movements display variations that are too complex to analyse.

Both a person waving and an animal's tail swinging feature repetitive, oscillatory movements, which might suggest similarity in their patterns. Figure 6.2 displays the spectrograms of a person standing and waving alongside those of several horses grazing. Despite initial expectations, the micro-Doppler patterns generated are notably distinct.

The grazing horses exhibit very low velocity fluctuations. The movement observed in the micro-Doppler signatures, in addition to the tail swings, is likely attributed to the horses' head movements as they lower and raise their heads to eat, as well as the slow leg movements as they shift position.



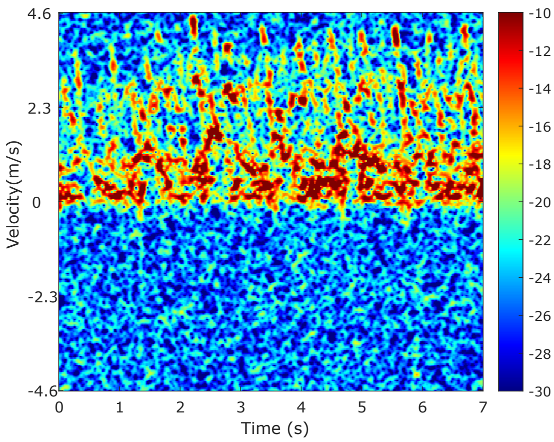
(a) Person Standing and Waving One Hand

(b) Horses Grazing and Swinging Tails

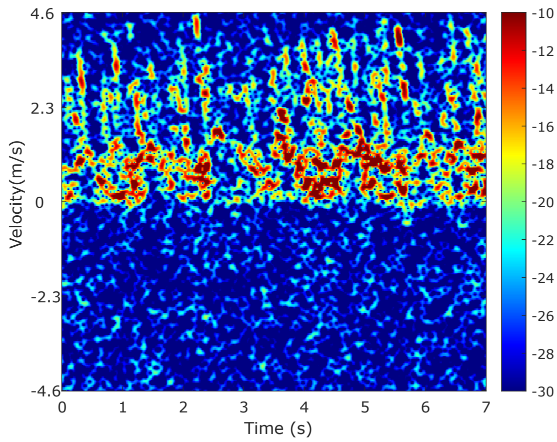
Fig. 6.2. Comparative Micro-Doppler Patterns: Human Hand-Waving Versus Horse Tail-Swinging

6.1.3 Data Augmentation Technique and Visualisation

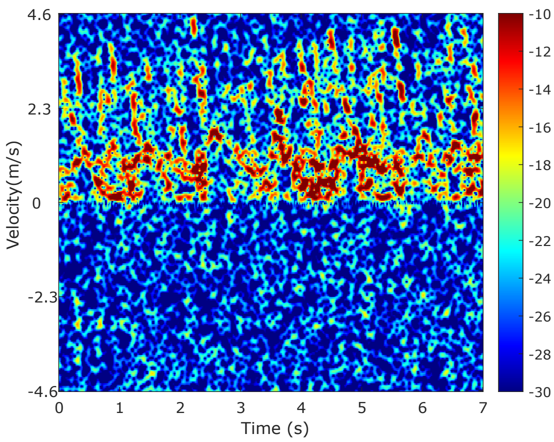
Figure 6.3 displays the result of the data augmentation strategy (as described in Section 5.1), using the same data for spectrogram generation. The spectrogram created with the Kaiser window function is shown in Figure 6.3(a), which presents data solely from channel 2 and expands the target signal by incorporating the adjacent range bins where the horse is detected. In Figure 6.3(b), the spectrogram is produced using only data from channel 2, employing the Hamming window function. Figure 6.3(c) illustrates the spectrogram that combines data from both channels using the Hann window function. Lastly, Figure 6.3(d) presents the spectrogram generated from channel 1 data using the Blackman window function. The spectrograms generated in Figures 6.3(b), 6.3(c) and 6.3(d) focus on processing the target in a single range bin.



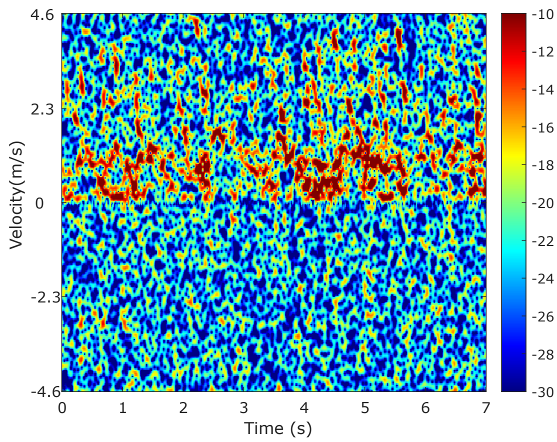
(a) Kaiser: Channel 2 (Added Range Bins)



(b) Hamming: Channel 2 (Single Range Bin)



(c) Hann: Channels 1 + 2 (Single Range Bin)



(d) Blackman: Channel 1 (Single Range Bin)

Fig. 6.3. Spectrograms of Horse Walking Towards Radar Illustrating Data Augmentation Technique

6.2 Dataset Composition

The breakdown of the augmented dataset into its constituent classes and the number of samples for each class is presented in Table 6.2.

TABLE 6.2

Dataset Composition by Target Type

Target Type	Number of Samples
Human	2681
Cows	1001
Dogs	398
Horses	1115

To directly address the research objectives outlined in Section 1.3, three datasets are constructed. Each dataset contains the same spectrogram data, but the class labels are specifically tailored to evaluate classification performance against these objectives. Table 6.3 presents the grouping of classes.

TABLE 6.3

Composition of the Datasets

Dataset	Classes
1	Human, Animal
2	Human, Horse, Dog, Cow
3	Horse, Dog, Cow

6.3 Radar Hardware Degradation

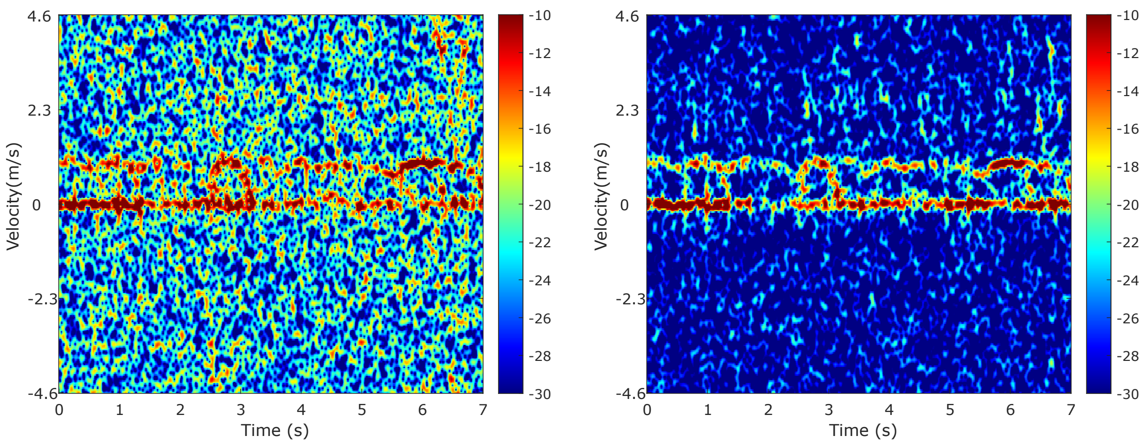
The radar hardware experienced degradation for unspecified reasons, complicating the data collection process during the project. Over time, one of the radar's receiver channels (channel 1) began to exhibit an increasingly high amount of noise for unknown reasons, resulting in a reduced signal-to-noise ratio and thereby diminished the radar's ability to detect targets clearly.

Initially, the impact was manageable and did not significantly affect the results, as targets were still clearly visible on the range-Doppler map. The data captured remained usable for further processing.

However, as the noise levels became severe, it became difficult to observe targets on the detection maps. This posed challenges both during data acquisition and in the post-processing stage, ultimately leading to a complete halt in data capture activities. While channel 2 remained unaffected and provided clean data, channel 1 limited the overall scope and volume of usable data for the study.

Figures 6.4(a) and 6.4(b) present spectrograms of a horse walking, with data captured using the degraded radar hardware, where channel 1 experiences high noise levels. Channel 1 has a lower signal-to-noise ratio when compared with channel 2.

Some of the radar data captured with the degraded radar toward the end of the project still contributed to the database, as part of the data augmentation strategy discussed in Section 5.1 and presented in Section 6.1.3.



(a) Channel 1 (b) Channel 2

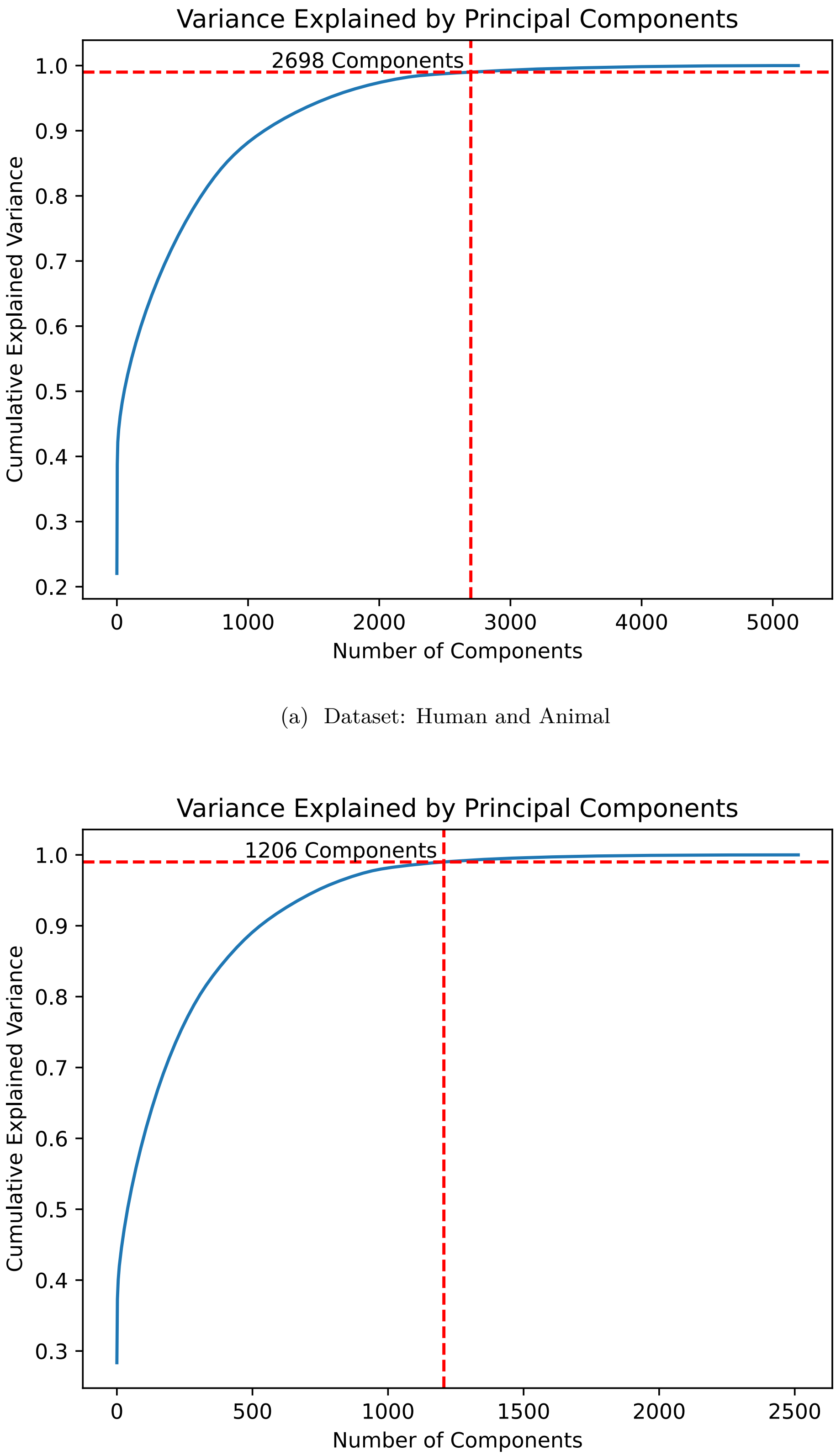
Fig. 6.4. Spectrograms of Horse Walking

6.4 Principal Component Analysis

The spectrogram matrix is compressed into a one dimensional vector containing 159 744 samples. PCA is performed in order to reduce this dimension to contain 99% of its variance. Retaining 99% variance with significantly fewer components allows for more efficient, easier data processing and analysis with minimal loss of information.

As depicted in Figure 6.5, the cumulative explained variance show that the number that accounts for 99% of the variance in the combined human and animal dataset is 2 698 components. For the animal-only dataset, 1 206 components captures 99% of the variance.

Figure 6.6 illustrates the original spectrogram of a person walking alongside its reconstructions using 10, 100, 500, 2 000 and 2 698 principal components. The reconstructions demonstrate the effect of dimensionality reduction on the representation of the spectrogram data. As the number of principal components increases, the micro-Doppler features represented by the reconstructed spectrogram improves. The reconstructed spectrogram from 2 698 principal components closely approximates the original spectrogram, which accounts for 99% of the variance in the dataset.



(a) Dataset: Human and Animal

(b) Dataset: Horses, Dogs and Cows

Fig. 6.5. Cumulative Explained Variance Showing the Number of Principal Components Capturing 99% of the Variance

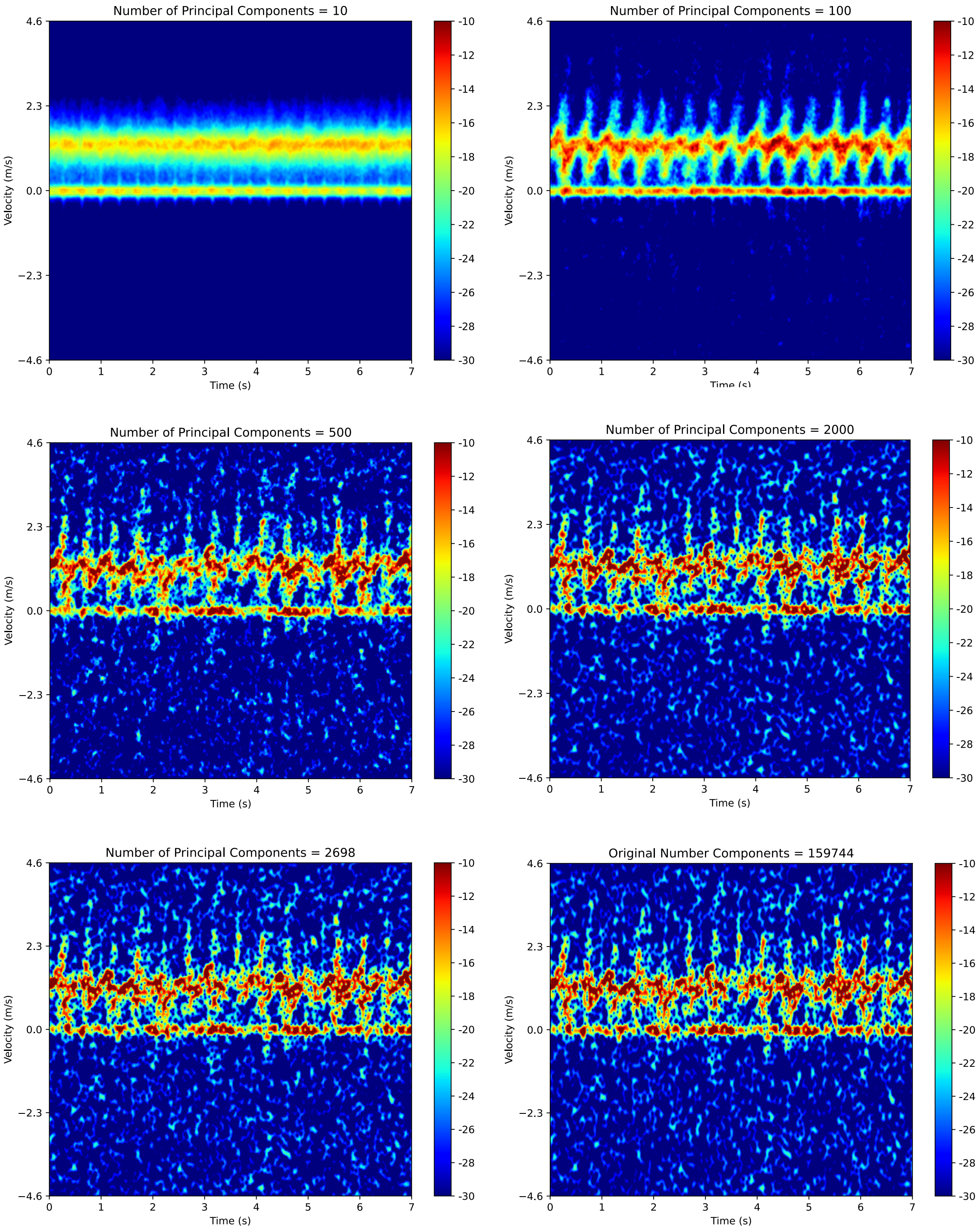


Fig. 6.6. Comparison of Original and Reconstructed Spectrograms of a Person Walking Using PCA

6.5 Classifier Parameter Optimisation

When running the `GridSearchCV` function across the datasets for the SVM, kNN and RF classifiers, the parameter values obtained to provide the highest accuracy are presented in Table 6.4 [126].

TABLE 6.4

Parameter Optimisation: Results from GridSearchCV

	Original Dataset			**PCA-Reduced Dataset**		
Classifier Parameters	1	2	3	1	2	3
SVM						
C	0.1	0.1	0.1	0.1	0.1	0.1
Gamma	1	1	1	1	1	1
Kernel	linear	linear	linear	linear	linear	linear
kNN						
k	1	1	1	1	1	1
RF						
Number of Estimators	300	300	100	300	300	200
Maximum Depth	40	30	40	10	30	30
Minimum Samples Split	10	2	2	2	2	5
Minimum Samples Leaf	1	2	1	4	1	1
Maximum Features	'log2'	'sqrt'	'sqrt'	'sqrt'	'sqrt'	'sqrt'

6.5.1 Analysis of Parameters

For the SVM and RF classifiers, a 3-fold cross-validation with accuracy as the scoring metric, is employed in the `GridSearchCV` due to computational constraints. The kNN classifier `GridSearchCV` uses a 10-fold cross-validation.

- **SVM:** The optimal parameters are consistent across all datasets for the SVM Classifier. This suggests that:
  - The SVM classifier shows stable performance across varying classes and dataset sizes.
  - The linear kernel is sufficient for separating the classes in the datasets, indicating that the data might be linearly separable, or close to it. It is therefore not necessary to select a Gamma parameter for a linear kernel.

- **kNN:** The process determined that the optimal number of neighbors, k, is 1, indicating that the closest neighbor provides the best classification outcome.
- **RF:** The variation in optimal parameters for the RF classifier, across different datasets and PCA reductions, highlight the sensitivity of ensemble methods to dataset characteristics such as size, class distribution and complexity.

The optimal parameters across classifiers remain largely unchanged when comparing the original datasets to their PCA-reduced versions. This implies that the PCA effectively captures the variance and maintains the dataset's main features required for classification.

Chapter 7

Results

This chapter presents the results of the classification algorithms applied to all dataset configurations. The performance metrics and analysis of each algorithm are detailed, highlighting their effectiveness in meeting the research objectives.

7.1 Classification Results

The classification accuracy results [126] provided in Table 7.1 reflect the performance of four different classification models (SVM, kNN, RF and CNN) on micro-Doppler spectrograms, processed as image data for the CNN and one-dimensional vectors for the other models.

Each model is evaluated by using three distinct class configurations, which are variations of the same underlying data arranged in different groupings, as described in Table 6.3. These evaluations are conducted on both the original datasets and those that have been dimensionally reduced via PCA.

TABLE 7.1
Classification Accuracy Results

	Original Dataset			**PCA-Reduced Dataset**		
Classifier	**1**	**2**	**3**	**1**	**2**	**3**
SVM	99.13%	97.50%	96.42%	98.85%	96.92%	96.62%
kNN	98.46%	95.38%	94.23%	97.59%	94.03%	93.44%
RF	97.40%	93.94%	91.25%	95.67%	90.66%	91.05%
CNN	95.38%	96.05%	94.83%	95.38%	93.07%	89.26%

7.1.1 Original Dataset Performance

In the original dataset, each model produces results with good accuracy across the different dataset configurations, which are discussed below:

- **SVM:** The SVM classifier performs impressively across all the dataset configurations. It achieves 99.13% accuracy when distinguishing between human and animal, 97.5% in the multi-class category of human versus individual animal species and 96.42% when classifying in the animal-only category.

- **kNN:** The kNN classifier demonstrates high accuracy, though slightly lower than the SVM. It scores 98.46% for differentiating human from animal, 95.38% in the multi-class human versus animal species category and 94.23% for the animal-only classification.

- **RF:** The RF classifier delivers positive results, with the scores being 97.40% for the binary category, 93.94% for the classification of human versus individual animal species and 91.25% when classifying between animals.

- **CNN:** The CNN achieves favourable results but varies more significantly across the different class configurations. It achieves 95.38% in the binary human versus animal category, scores higher accuracy with 96.05% in the multi-class human versus animal species category and 94.83% for the animal-only group.

7.1.2 PCA-Reduced Dataset Performance

When applying classification techniques to the PCA-reduced dataset, a slight general decrease in model accuracy is observed and is discussed below:

- **SVM:** The SVM maintains high accuracy with the PCA-reduced set and outperforms the other models. It achieves 98.85% for differentiating between human and animal, 96.92% for the multi-class classification and 96.62% for the animal-only category.
- **kNN:** The kNN classifier yields results indicating a slight decrease in accuracy after PCA application across the multi-class classification scenarios, achieving 94.03% in the human versus animal species category and 93.44% for animal-only category. For the binary classification, the classifier delivers 97.59% accuracy.
- **RF:** The RF classifier reports a slight drop in performance with classification on the PCA-reduced set producing 95.67% accuracy for the binary classification. Notably, the animal-only classification delivers a higher accuracy result of 91.05% to the multi-class human and animal species category which scores 90.66%.
- **CNN:** The CNN's accuracy results decreases with the PCA-reduced set to 93.07% for the multi-class human and animal species classification, and 89.26% for animal-only group. However, for the binary classification case, the result remains at the accuracy level of 95.38%.

7.1.3 Analysis of Results from Original and PCA-Reduced Datasets

The SVM classifier consistently outperforms the other models in both binary and multi-class classification scenarios. This can be attributed to its effectiveness in managing high dimensional data. This ability to identify and leverage the most critical features likely contributes to the impressive accuracy outcome, even when feature dimensions are reduced through PCA.

The performance of the kNN classifier is slightly less accurate to that of the SVM. The slight decline in accuracy on the reduced dataset is likely due to the kNN's dependency on distance metrics that could be distorted by the PCA reduction technique.

The RF classifier generally underperforms when compared to the other models, particularly in the scenarios involving multiple classes. However, its performance is slightly better in the binary classification case. The model displays sensitivity to the reduction in dimensionality, likely due to a possible reliance on a more extensive array of configurable parameters to maintain accuracy, which the PCA disrupts. The inconsistency in the multi-class classification results may also be due to the differing parameter values obtained during the classifier parameter optimization stage (refer to Table 6.4). Further optimization of the RF classifier parameters could lead to more consistent results across all categories.

When using the CNN model, the multi-class classification, which includes both human and animal species, outperforms the binary classification for the full dataset however this is not reflected in the PCA-reduced dataset. Interestingly, the results for the binary human versus animal classification were identical for both the full dataset and the PCA-reduced set, indicating that the PCA-reduced dataset retains sufficient information for accurate classification in this context. The accuracy level decrease is more pronounced for the multi-class classification in the PCA-reduced datasets, which is possibly due to the convolutional layers losing critical information during feature abstraction. The design of the CNN model requires further refinement across each dataset configuration to achieve more optimal results in multi-class classification tasks.

7.2 Multi-Stage Classification Process

The binary classification results for distinguishing between human and animal demonstrated notable accuracy levels, reaching over 95%. Similarly, the performance metrics for classification between different animal species also registered accuracy percentages in the nineties. These results led to the development of a two-stage classification approach, aiming to leverage the performance of these two categories.

The original dataset is used to perform the two-stage classification. In the first stage of this two-stage classification process, binary classification is performed and used to determine whether the target subject is a human or an animal. If the subject is classified as an animal, the second stage is initiated to identify the specific type of animal (trained by using the animal-only dataset). This approach is designed to assess if the two-stage classification process can outperform the broader multi-class category classification, where humans and all animal species are classified in a single-stage.

The findings from this approach is presented in Table 7.2.

TABLE 7.2

Two-Stage Classification Accuracy Results

Stage 1 – Stage 2	**Accuracy (%)**
SVM – SVM	97.66%
SVM – kNN	96.88%
SVM – RF	97.27%
SVM – CNN	97.08%
kNN – SVM	96.17%
kNN – kNN	95.40%
kNN – RF	95.59%
kNN – CNN	95.59%
RF – SVM	94.17%
RF – kNN	93.77%
RF – RF	93.94%
RF – CNN	93.76%
CNN – SVM	94.29%
CNN – kNN	93.50%
CNN – RF	93.70%
CNN – CNN	93.90%

7.3 Analysis of Results

7.3.1 SVM-SVM Classification

The SVM-SVM combination in the two-stage classification achieved the highest accuracy, distinguishing the SVM as the best model for both stages. The first stage records an accuracy of 99.13%, where only 1 animal (out of 506 samples) was misclassified as a human and 8 humans (out of 533 samples) as animals. Refer to Figure 7.1 for the confusion matrix from the first stage classification.

In the second stage, the model achieved high accuracy at 97.66%. Table 7.3 presents the sklearn.metrics `classification_report` from the second stage. The results indicate high precision and recall for all animal types, although there was a small number of misclassification's mostly involving a horse being predicted as a cow and cows being predicted as horses. See Figure 7.2 for the confusion matrix representing the second stage classification.

TABLE 7.3

SVM-SVM Classification Report

	Precision	**Recall**	**F1-score**	**Support**
Human (Misclassified)	0.00	0.00	0.00	8
Horse	0.98	1.00	0.99	233
Dog	0.99	1.00	0.99	76
Cow	0.97	0.98	0.98	196
Accuracy			0.98	513
Macro avg	0.73	0.75	0.74	513
Weighted avg	0.96	0.98	0.97	513

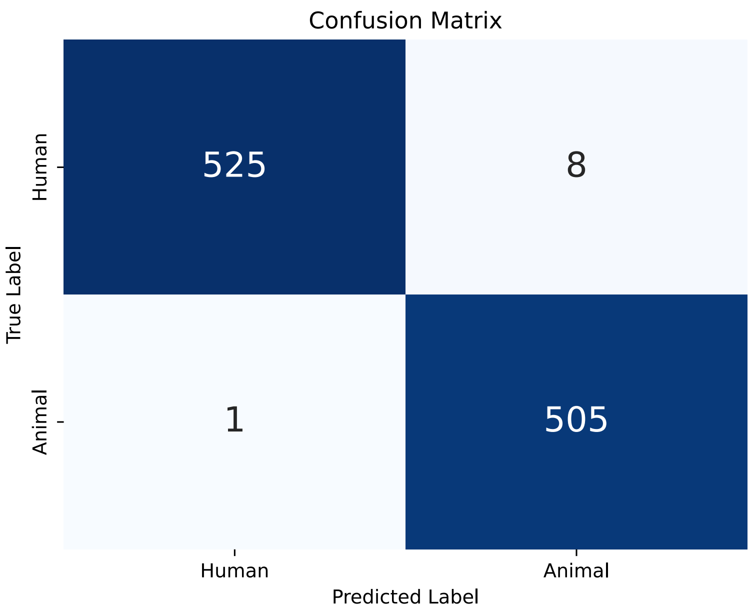


Fig. 7.1. SVM First Stage

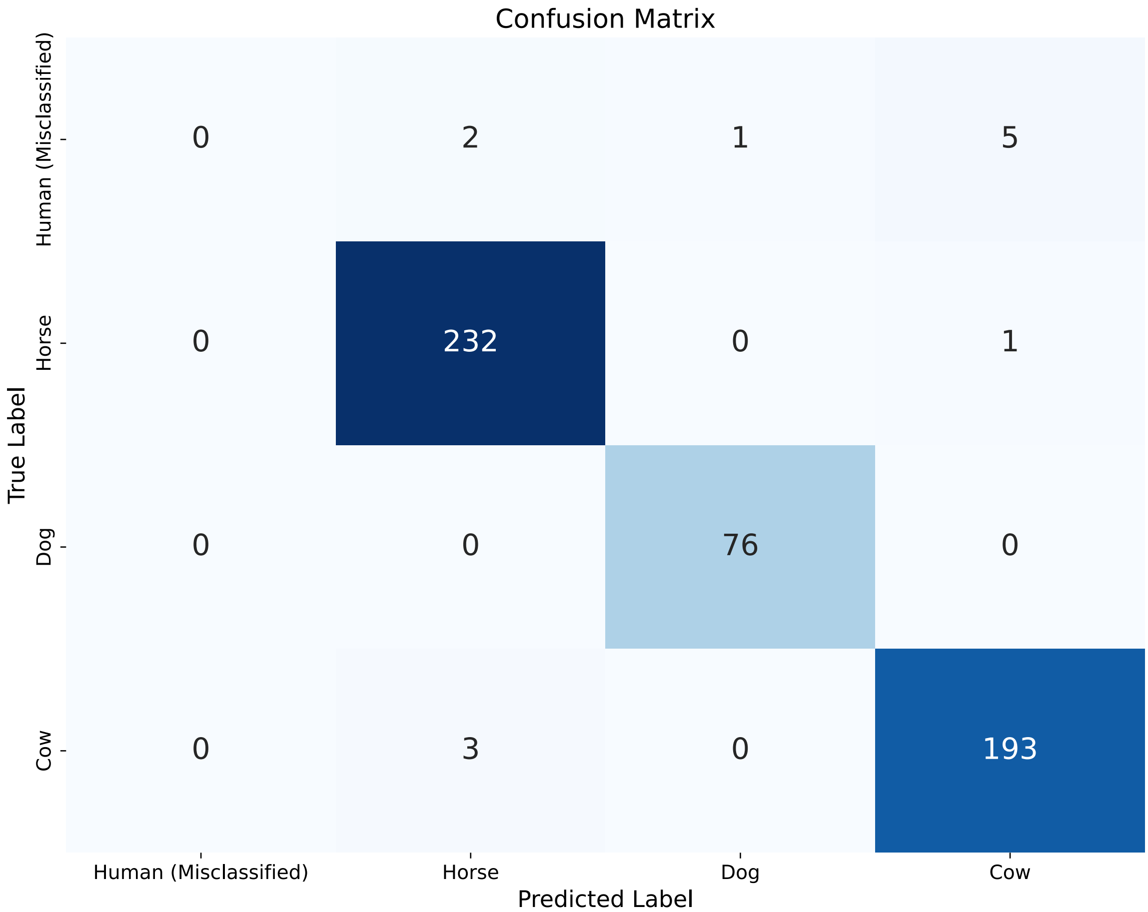


Fig. 7.2. SVM Second Stage

7.3.2 Two-Stage Classification Analysis

When the SVM is used in the first stage, the overall accuracy is highest, across all combinations, when paired with SVM in the second stage as well. The SVM-SVM combination confirms that the SVM is effective in handling both binary and multi-class classifications in this dataset. Combinations of the SVM with kNN, RF and CNN in the second stage also reveal strong results.

The combinations where kNN and RF serve as the first stage model result in satisfactory accuracies, with the highest being kNN-SVM at 96.17% and the lowest RF-CNN at 93.76%. The CNN combinations achieved the lowest accuracies among all the models when used in the first stage, with CNN-SVM scoring the highest at 94.29%.

In the original dataset configuration, the classification accuracy for SVM in the human versus animal species category is 97.50%, which is almost as accurate as the best-performing two-stage SVM-SVM model at 97.66%.

Other model combinations in the two-stage classification also perform comparably to single-stage when directly classifying in the multi-class human versus animal species category.

These results indicate that the two-stage classification model does not produce higher accuracy and could be attributed to error propagation in instances where the binary classification underperforms.

The choice between a single-stage and multi-stage classification approach should consider computational efficiency as well as practical ease of implementation for application in real systems.

Chapter 8

Conclusion

This concluding chapter summarises the key findings from the research and discusses the classification results. It reflects on how the objectives outlined at the beginning of the study are addressed, highlighting the strengths and limitations of the methodologies used. Additionally, potential directions for future research in micro-Doppler radar technology are proposed.

8.1 Research Summary

This research employs a low cost FMCW radar, to collect real-world data for classifying humans and animals, due to its availability and ease of deployment. The study focuses on capturing and analysing the movement patterns of humans and livestock animals, including horses, dogs and cows, in the micro-Doppler spectrum. Signal processing techniques such as creating range-Doppler and CFAR maps for detection and using STFT for spectrogram generation are applied to extract micro-Doppler signatures.

A comprehensive literature review identifies several models that successfully perform micro-Doppler spectrogram classification, including kNN, SVM, HMM, NB, Gaussian models, decision trees and various deep learning approaches, including traditional NNs and advanced architectures like residual networks. Based on this review, kNN, SVM, RF and CNN models are chosen for the classification task.

To increase the dataset, data augmentation techniques are employed, generating spectrograms using various window functions (Hann, Hamming, Chebyshev, Blackman and Kaiser) and channel-specific data. PCA is used for data reduction, and classification techniques are performed on both the full dataset and the PCA-reduced dataset by means of the four chosen models.

Parameter optimisation is conducted and applied to the SVM, RF, and kNN classifiers, with results evaluated mainly through accuracy scores, confusion matrices and classification performance reports. The classification is performed on three different data configurations: binary category of human versus animal, multi-class category of human versus horse, dog and cow, as well as the case with animal species only.

A two-stage classification process is developed, where stage 1 differentiates between human and animal and stage 2 classifies the specific animal species. All 16 permutations are evaluated.

8.2 Discussion of Results

The SVM model achieves an accuracy of 99.13% when classifying in the binary category of human vs animal, 97.5% when broadly distinguishing human versus dog, horse and cow and 96.42% when differentiating between animals species only. In the two-stage classification process, the SVM model achieves an accuracy of 97.66%, which is a very minor improvement over the multi-class category result of 97.5%.

The other models also perform well, registering accuracy scores in the nineties. The kNN model is second best with an accuracy of 98.46% in distinguishing between human and animal. However, the kNN and RF models perform better in the binary classification, showing sensitivity to the multi-class classification categories. The CNN model performs well in the broad multi-class category, with 96.05% accuracy, but its overall classification performance degrades slightly as the dimensionality of the data is reduced.

The results indicate that PCA can significantly reduce the dimensionality of the data while still preserving classification accuracy. This reduction is useful for speeding up the training and classification processes, which is particularly beneficial when dealing with larger datasets or limited computational resources.

Overall, this research ultimately reveals that the SVM is the best performing model for this dataset.

8.3 Challenges

Several challenges were encountered during this research. Midway through the project, the radar hardware degraded, necessitating the use of data augmentation techniques to increase the dataset. Limited computational power restricted the exploration of the HMM-GMM model and extensive parameter optimisation for the chosen models. This limitation also affected the exploration of different CNN configurations for the different dataset configurations. Additionally, the availability of the radar was affected by ongoing development work, and environmental factors such as winds and rain limiting the data capture time.

8.4 Further Work

To improve the efficacy of target classification in poaching applications, additional micro-Doppler data of targets such as humans, vehicles, UAV's and various animal species is required. The animal dataset should be expanded to include both individual animals and groups of animals. This will facilitate the deeper exploration of movement patterns and the extraction of micro-Doppler features, such as the frequency of limb movements, tail movements and main body oscillations.

Tracking algorithms should be developed in order to track animal targets in their natural environments. This will depend on the type of radar deployed (scanning, surveillance, or staring) and the specific movement patterns of the animals. For instance, livestock tend to cluster together and varying grazing patterns can be challenging to account for in the algorithm. Designing and developing sophisticated tracking algorithms for such diverse and dynamic animal movements within groups will significantly enhance future data processing efforts for micro-Doppler animal classification.

While PCA has proven to be an effective data reduction technique in this research, it is essential to explore whether it will remain effective as the dataset diversity grows. Future work should investigate other techniques such as LDA, ICA and pure feature extraction (such as the methods used by Shrestha *et al.* [36, 37] for animal lameness detection). This is important for scalability, as factors like storage, computational power and speed become crucial in a real-life system, making the size of the dataset, training and classification times all critical considerations.

This research uses a low cost radar system due to the high expense and limited availability of more high-performance radars. While the low cost system proves to be a viable option for shorter range target classification, expanding research with high-performance radars could improve Doppler resolution and range coverage, especially for larger parks or organizations with the necessary resources. For smaller parks or farms, where such high technology equipment is not feasible, low cost systems remain a valuable tool for target classification and is worth the further exploration.

8.5 Concluding Remarks

Despite challenges faced, this research indicates that it is possible to classify between humans and animals using micro-Doppler spectrogram signatures generated from using a low cost radar system.

The research successfully demonstrates the potential of using radar with micro-Doppler classification technology for anti-poaching and conservation-based initiatives.

References

[1] Geeks For Geeks Authors. Getting started with Classification. [Online]. Available: https://www.geeksforgeeks.org/getting-started-with-classification/

[2] Simplilearn. What is Epoch in Machine Learning? [Online]. Available: https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learning#:~:text=Machine%20learning%20models%20are%20trained,training%20data%20through%20the%20algorithm.

[3] C. Wolff. Frequency-Modulated Continuous-Wave Radar (FMCW Radar). [Online]. Available: https://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html

[4] V. C.Chen, *The Micro-Doppler Effect in Radar Second Edition.* Boston, London: Artech House, 2019.

[5] IFAW. What is poaching? [Online]. Available: https://www.ifaw.org/international/journal/what-is-poaching

[6] W3 Schools Authors. Tensors. [Online]. Available: https://www.w3schools.com/ai/ai_tensors.asp

[7] The European Space Agency Authors. Satellite Frequency Bands. [Online]. Available: https://www.esa.int/Applications/Connectivity_and_Secure_Communications/Satellite_frequency_bands

[8] The ZeroMQ Authors. ZMQ Documentation - Get Started. [Online]. Available: https://zeromq.org/get-started/

[9] S. Bhega. Rhino poaching declines in Kruger, but poachers are on the rampage in KZN. [Online]. Available: https://mg.co.za/environment/2023-02-08-rhino-poaching-declines-in-kruger-but-poachers-are-on-the-rampage-in-kzn/

[10] Department of Forestry, Fisheries and Environment. Minister Barbara Creecy on fight against rhino poaching in national parks. [Online]. Available: https://www.gov.za/speeches/minister-barbara-creecys-relentless-fight-against-rhino-poaching-national-parks-6-feb-2023

[11] African Wildlife Foundation. [Online]. Available: https://campaign.awf.org/poaching-infographic/

[12] Save The Rhino. 548 rhinos poached in 2022. [Online]. Available: https://www.savetherhino.org/africa/548-rhinos-poached-in-2022/

[13] Wildlife ACT. Black Rhino Release of Four Orphans Back into the Wild. [Online]. Available: https://www.wildlifeact.com/blog/black-rhino-release-back-into-the-wild

[14] The Things Network. What are LoRa and LoRaWAN? [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/

[15] T. Walters. Turning the tide? 'We've lost just one rhino in 350 days' — Sabi Sand. [Online]. Available: https://www.dailymaverick.co.za/article/2023-12-20-turning-the-tide-weve-lost-just-one-rhino-in-350-days-sabi-sand/

[16] K. Zeeman. South African rhinos: Poaching on the rise in new hotspot. [Online]. Available: https://www.bbc.com/news/world-africa-68414362

[17] Reunert. Radars used to protect rhinos in Kruger National Park. [Online]. Available: https://reunert.com/media-release-article.php?articleID=14168

[18] M. Lenhoff. Reutech Radar unit an integral component of Meerkat surveillance system. [Online]. Available: https://www.defenceweb.co.za/office/reutech/PressRelease.php?StoryID=46298

[19] Reunert. Anti-poaching Surveillance Radar video. [Online]. Available: https://www.reunertae.com/RAE-interactive-presentation/business-areas/rrs/

[20] Getaway Authors. U.S. military radar detection equipment to be used in the fight against rhino poaching. [Online]. Available: https://www.getaway.co.za/environment/u-s-military-radar-detection-equipment-fight-rhino-poaching/

[21] C. Wolff. Basics of radar technology. [Online]. Available: https://www.radartutorial.eu/

[22] V. C.Chen, *The Micro-Doppler Effect in Radar.* 685 Canton Street Norwood, MA 02062: Artech House, 2011.

[23] Carnegie Mellon University- Electrical and Computer Engineering, College of Engineering. Short-Time Fourier Transforms. [Online]. Available: https://course.ece.cmu.edu/~ece491/lectures/L25/STFT_Notes_ADSP.pdf

[24] National Instruments,Inc. Short-Time Fourier Transform (Advanced Signal Processing Toolkit). [Online]. Available: https://www.ni.com/docs/en-US/bundle/labview-advanced-signal-processing-toolkit-api-ref/page/lvasptconcepts/aspt_stft.html

[25] National Instruments, Inc. Using the MUSIC Method (Advanced Signal Processing Toolkit). [Online]. Available: https://www.ni.com/docs/en-US/bundle/labview-advanced-signal-processing-toolkit-api-ref/page/lvasptconcepts/aspt_music.html

[26] I. Daubechies, J. Lu, and H.-T. Wu, "Synchrosqueezed Wavelet Transforms: an Empirical Mode Decomposition-like Tool," *Applied and computational harmonic analysis*, vol. 30, no. 2, pp. 243–261, 2011.

[27] X. Wu and T. Liu, "Spectral decomposition of seismic data with reassigned smoothed pseudo Wigner–Ville distribution," *Journal of Applied Geophysics*, vol. 68, no. 3, pp. 386–393, 2009.

[28] M. M. Hasan, A. Pandey, and A. A. B. Raj, "UAV Classification from Micro-Doppler Signature-based Time Frequency Images using SVM," in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 2022, pp. 1225–1230.

[29] V. C. Chen, "Evaluation of Bayes , ICA , PCA and SVM Methods for Classification," in *RTO NATO*, 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:19011008

[30] Y. Zhao, H. Zhou, S. Lu, Y. Liu, X. An, and Q. Liu, "Human Activity Recognition Based on Non-Contact Radar Data and Improved PCA Method," *Applied Sciences*, vol. 12, no. 14, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/14/7124

[31] G. Ciattaglia, L. Senigagliesi, D. Alidori, L. Cipriani, G. Iadarola, S. Spinsante, and E. Gambi, "Drone classification using mmWave micro-Doppler radar measurements," in *2022 IEEE 9th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2022, pp. 259–264.

[32] B. Jokanovic, M. Amin, and F. Ahmad, "Radar fall motion detection using deep learning," in *2016 IEEE Radar Conference (RadarConf)*, 2016, pp. 1–6.

[33] A.-K. Seifert, A. M. Zoubir, and M. G. Amin, "Radar-based human gait recognition in cane-assisted walks," in *2017 IEEE Radar Conference (RadarConf)*, 2017, pp. 1428–1433.

[34] Analytics Vidhya Content Team. What Is Principal Component Analysis and How It Works? [Online]. Available: https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/

[35] V. Singh, S. Bhattacharyya, and P. K. Jain, "Micro-Doppler classification of human movements using spectrogram spatial features and support vector machine," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 30, no. 8, p. e22264, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/mmce.22264

[36] A. Shrestha, C. Loukas, J. Le Kernec, F. Fioranelli, V. Busin, N. Jonsson, G. King, M. Tomlinson, L. Viora, and L. Voute, "Animal Lameness Detection With Radar Sensing," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 8, pp. 1189–1193, 2018.

[37] A. Shrestha, J. Le Kernec, F. Fioranelli, J. F. Marshall, and L. Voute, "Gait analysis of horses for lameness detection with radar sensors," in *International Conference on Radar Systems (Radar 2017)*, 2017, pp. 1–6.

[38] K. Hanifi and M. E. Karsligil, "Elderly Fall Detection With Vital Signs Monitoring Using CW Doppler Radar," *IEEE Sensors Journal*, vol. 21, no. 15, pp. 16 969–16 978, 2021.

[39] J. Lee, J. Kwon, J. Bae, and C. Lee, "Classification Algorithms for Human and Dog Movement Based on Micro-Doppler Signals," *IEIE Transactions on Smart Processing and Computing*, vol. 6, pp. 10–17, 02 2017.

[40] H. Okinaka, K. Saho, M. Fujimoto, S. Go, M. Masugi, K. Sugano, K. Uemura, and M. Matsumoto, "Gait Classification of Healthy Young and Elderly Adults Using Micro-Doppler Radar Remote Sensing," in *2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS)*, 2018, pp. 1222–1226.

[41] P. Molchanov, K. Egiazarian, J. Astola, R. I. A. Harmanny, and J. J. M. de Wit, "Classification of small UAVs and birds by micro-Doppler signatures," in *2013 European Radar Conference*, 2013, pp. 172–175.

[42] S. Björklund, T. Johansson, and H. Petersson, "Evaluation of a micro-Doppler classification method on mm-wave data," in *2012 IEEE Radar Conference*, 2012, pp. 0934–0939.

[43] F. Fioranelli, H. Li, J. Le Kernec, V. Busin, N. Jonsson, G. King, M. Tomlinson, and L. Viora, "Radar-based evaluation of lameness detection in ruminants: preliminary results," in *2019 IEEE MTT-S International Microwave Biomedical Conference (IMBioC)*, vol. 1, 2019, pp. 1–4.

[44] C. Ding, Z. Ding, L. Wang, and Y. Jia, "A Fall Detection Method Based on K-Nearest Neighbor Algorithm with MIMO Millimeter-Wave Radar," in *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, 2021, pp. 624–628.

[45] Y. Sun, T. Fei, F. Schliep, and N. Pohl, "Gesture Classification with Handcrafted Micro-Doppler Features using a FMCW Radar," in *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 2018, pp. 1–4.

[46] R. Coppola, S. Ahmed, and M.-S. Alouini, "Users Classification Based on Bi-Frame Micro-Doppler With 24-GHz FMCW Radar," *Frontiers in Signal Processing*, vol. 2, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frsip.2022.864538

[47] N. Aldabashi, S. M. Williams, A. Eltokhy, E. Palmer, P. Cross, and C. Palego, "A Machine Learning Integrated 5.8-GHz Continuous-Wave Radar for Honeybee Monitoring and Behavior Classification," *IEEE Transactions on Microwave Theory and Techniques*, pp. 1–11, 2023.

[48] J. Hou, G. Chen, Q. Zhou, C. Liu, X. Zuo, Y. Tang, and C.-T. Cheng, "Indoor human detection based on micro-Doppler features in the presence of interference from moving clutter sources," *Physical Communication*, vol. 58, p. 102037, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187449072300040X

[49] T. K. Huang X, Patel N, "Application of mmWave Radar Sensor for People Identification and Classification," *Sensors(Basel, Switzerland)*, vol. 23,8, 2023. [Online]. Available: https://doi.org/10.3390/s23083873

[50] Scikit-Learn Authors. sklearn.ensemble.RandomForestClassifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[51] J. Hui. Speech Recognition — GMM, HMM. [Online]. Available: https://jonathan-hui.medium.com/speech-recognition-gmm-hmm-8bb5eff8b196

[52] C. Bento. Markov models and Markov chains explained in real life: probabilistic workout routine. [Online]. Available: https://towardsdatascience.com/markov-models-and-markov-chains-explained-in-real-life-probabilistic-workout\-routine-65e47b5c9a73

[53] A. Kumar. Hidden Markov Models: Concepts, Examples. [Online]. Available: https://vitalflux.com/hidden-markov-models-concepts-explained-with-examples/

[54] M. O. Padar, A. E. Ertan, and C. Candan, "Classification of human motion using radar micro-Doppler signatures with hidden Markov models," in *2016 IEEE Radar Conference (RadarConf)*, 2016, pp. 1–6.

[55] W. van Eeden, J. De Villiers, R. Berndt, W. Nel, and E. Blasch, “Micro-Doppler radar classification of humans and animals in an operational environment,” *Expert Systems with Applications*, vol. 102, 02 2018.

[56] M. Jahangir, K. Ponting, and J. O’Loghlen, “A robust Doppler classification technique based on hidden Markov models,” in *RADAR 2002*, 2002, pp. 162–166.

[57] I. Bilik, J. Tabrikian, and A. Cohen, “GMM-based target classification for ground surveillance Doppler radar,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 267–278, 2006.

[58] A. Mesloub, K. Abed-Meraim, and A. Belouchrani, “Ground moving target classification based on micro-Doppler signature using novel spectral information features,” in *2017 European Radar Conference (EURAD)*, 2017, pp. 255–258.

[59] Z. Zhang and A. G. Andreou, “Human identification experiments using acoustic micro-Doppler signatures,” in *2008 Argentine School of Micro-Nanoelectronics, Technology and Applications*, 2008, pp. 81–86.

[60] K. Kalgaonkar and B. Raj, “Acoustic Doppler sonar for gait recogination,” in *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, 2007, pp. 27–32.

[61] A. Singh. Build Better and Accurate Clusters with Gaussian Mixture Models. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/

[62] J. Brownlee. A Gentle Introduction to Expectation-Maximization (EM Algorithm). [Online]. Available: https://machinelearningmastery.com/expectation-maximization-em-algorithm/

[63] J. C. Olamendy. Understanding Gaussian Mixture Models: A Comprehensive Guide. [Online]. Available: https://medium.com/@juanc.olamendy/understanding-gaussian-mixture-models-a-comprehensive-guide-df30af59ced7

[64] S. Saha. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. [Online]. Available: https://medium.com/towards-data-science/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[65] M. Mandal. Introduction to Convolutional Neural Networks (CNN). [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

[66] Y. D. Dadon, S. Yamin, S. Feintuch, H. H. Permuter, I. Bilik, and J. Taberkian, “Moving Target Classification Based on micro-Doppler Signatures Via Deep Learning,” in *2021 IEEE Radar Conference (RadarConf21)*, 2021, pp. 1–6.

[67] K. Saho, K. Shioiri, M. Fujimoto, and Y. Kobayashi, “Micro-Doppler Radar Gait Measurement to Detect Age- and Fall Risk-Related Differences in Gait: A Simulation Study on Comparison of Deep Learning and Gait Parameter-Based Approaches,” *IEEE Access*, vol. 9, pp. 18 518–18 526, 2021.

[68] S. Biswas, B. Bartlett, J. E. Ball, and A. C. Gurbuz, “Classification of Traffic Signaling Motion in Automotive Applications Using FMCW Radar,” in *2023 IEEE Radar Conference (RadarConf23)*, 2023, pp. 1–6.

[69] G. Atkinson, M. Jahangir, D. White, J. Wayman, C. Baker, J. Sadler, J. S. Reynolds, and M. Antoniou, “The Role of Target Signatures in Bird-Drone Classification,” in *2023 IEEE Radar Conference (RadarConf23)*, 2023, pp. 1–6.

[70] A. Amran, E. Purwantini, A. Wijayanto, R. T. Widodo, B. Nur Iman, E. Satriyanto, E. Puspita, and H. Oktavianto, “Implementation of Fall Detection for The Elderly Home Support using 24-GHz Radar Sensor,” in *2022 International Electronics Symposium (IES)*, 2022, pp. 166–170.

[71] D. T. Victor C. Chen and W. J. Miceli, *Radar Micro-Doppler Signature Processing and Applications*. Michael Faraday House Six Hills Way, Stevenage Herts, SG1 2AY, United Kingdom: The Institution of Engineering and Technology, 2014.

[72] Robin Radar Systems. Do You Need Different Radar Solutions to Detect Bats And Birds at Wind Farms? [Online]. Available: https://www.robinradar.com/blog/do-you-need-different-solutions-to-detect-bats-and-birds-at-wind-farms

[73] Robin Radar. Max avian radar. [Online]. Available: https://www.robinradar.com/products/max-radar

[74] R. Berndt, L. Wabeke, V. J. Van Rensburg, F. Potgieter, and K. Kloke, “Ground-based Surveillance and Classification Radar for Wildlife Protection,” in *2023 IEEE International Radar Conference (RADAR)*, 2023, pp. 1–5.

[75] S. Birch. ’Boaring’ in on vehicle-to-animal road safety. [Online]. Available: https://www.sae.org/news/2020/10/micro-doppler-radar-for-adas-and-avs

[76] Spectrum Instrumentation. Intelligent Road-Radar to detect wild animals. [Online]. Available: https://spectrum-instrumentation.com/news/202005_Intelligent_Road_Radar_Detects_Wild_Animals.php

[77] C. Karabacak, S. Z. Gurbuz, A. C. Gurbuz, M. B. Guldogan, G. Hendeby, and F. Gustafsson, “Knowledge Exploitation for Human Micro-Doppler Classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 10, pp. 2125–2129, 2015.

[78] A. Alzogaiby, “Using micro-Doppler radar signals for human gait detections,” Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2014.

[79] M. G. Amin, Y. D. Zhang, F. Ahmad, and K. D. Ho, “Radar Signal Processing for Elderly Fall Detection: The future for in-home monitoring,” *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 71–80, 2016.

[80] S. Liu, Z. Zeng, Y. D. Zhang, T. Fan, T. Shan, and R. Tao, “Automatic human fall detection in fractional fourier domain for assisted living,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 799–803.

[81] Y. Balal, N. Balal, Y. Richter, and Y. Pinhasi, “Time-Frequency Spectral Signature of Limb Movements and Height Estimation Using Micro-Doppler Millimeter-Wave Radar,” *Sensors*, vol. 20, no. 17, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/17/4660

[82] J. Duan, L. Zhang, Y. Wu, Y. Zhang, Z. Zhao, and X. Guo, “Classification of birds and drones by exploiting periodical motions in Doppler spectrum series,” *Journal of Systems Engineering and Electronics*, vol. 34, no. 1, pp. 19–27, 2023.

[83] S. Yoon, S. Kim, J. Jung, S. Cha, Y. Baek, B. Koo, I. Choi, and S. Park, “Efficient Protocol to Use FMCW Radar and CNN to Distinguish Micro-Doppler Signatures of Multiple Drones and Birds,” *IEEE Access*, vol. 10, pp. 26 033–26 044, 2022.

[84] X. Zhang, V. Mehta, M. Bolic, and I. Mantegh, “Hybrid AI-enabled Method for UAS and Bird Detection and Classification,” in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 2803–2807.

[85] Y. Zhao, X. Zhang, and F. Fioranelli, “Initial results of Radar-based classification of commercial drone carrying small payloads,” in *2019 International Radar Conference (RADAR)*, 2019, pp. 1–4.

[86] MathWorks, *MATLAB*, MathWorks, Natick, Massachusetts, United States, 2023.

[87] MathWorks. Hann Window. [Online]. Available: https://www.mathworks.com/help/signal/ref/hann.html

[88] D. Ivkovic, B. Zrnic, and M. Andrić, “Fusion CFAR detector in receiver of the software defined radar,” 01 2013, pp. 762–767.

[89] J. Jen, *A Study of CFAR Implementation Cost and Performance Tradeoffs in Heterogeneous Environments*. California State Polytechnic University, Pomona, 2011. [Online]. Available: https://books.google.co.za/books?id=WChRtwAACAAJ

[90] T.-T. V. Cao, J. Palmer, and P. E. Berry, “False alarm control of CFAR algorithms with experimental bistatic radar data,” in *2010 IEEE Radar Conference*, 2010, pp. 156–161.

[91] The MathWorks, Inc. Constant False Alarm Rate (CFAR) Detection. [Online]. Available: https://www.mathworks.com/help/phased/ug/constant-false-alarm-rate-cfar-detection.html

[92] V. Bhangod. Spectrogram plot without using spectrogram command. [Online]. Available: https://www.mathworks.com/matlabcentral/answers/104488-spectrogram-plot-without-using-spectrogram-command

[93] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[94] F. Chollet, “Keras: The Python Deep Learning API,” https://keras.io/.

[95] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: Large-scale machine learning on heterogeneous systems,” https://www.tensorflow.org/, 2015.

[96] A. Saleh, M. Sheaves, D. Jerry, and M. Rahimi Azghadi, “Applications of deep learning in fish habitat monitoring: A tutorial and survey,” *Expert Systems with Applications*, vol. 238, p. 121841, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423023436

[97] A. AG. Federated Learning. [Online]. Available: https://vocal.media/education/federated-learning?

[98] MathWorks. Chebyshev Window. [Online]. Available: https://www.mathworks.com/help/signal/ref/chebwin.html

[99] MathWorks Authors. Blackman Window. [Online]. Available: https://www.mathworks.com/help/signal/ref/blackman.html

[100] MathWorks. Hamming Window. [Online]. Available: https://www.mathworks.com/help/signal/ref/hamming.html

[101] MathWorks Authors. Kaiser Window. [Online]. Available: https://www.mathworks.com/help/signal/ref/kaiser.html

[102] Scikit-Learn Developers. sklearn.model selection train test split. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[103] Scikit-Learn Authors. sklearn.decomposition.PCA. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[104] Scikit-Learn Developers. sklearn.model GridSearchCV. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[105] Scikit-Learn Authors. Support Vector Machines. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html

[106] M. Aibin. The C Parameter in Support Vector Machines. [Online]. Available: https://www.baeldung.com/cs/ml-svm-c-parameter

[107] Scikit-Learn Authors. sklearn.neighbors.KNeighborsClassifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[108] W. Koehrsen. Hyperparameter Tuning the Random Forest in Python. [Online]. Available: https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

[109] Databricks Authors. TensorFlow. [Online]. Available: https://www.databricks.com/glossary/tensorflow-guide#:~:text=As%20you%20now%20know%2C%20Google,theoretical%20understanding%20of%20deep%20learning.

[110] TensorFlow Authors. tf.data: Build TensorFlow input pipelines:Batching dataset elements. [Online]. Available: https://www.tensorflow.org/guide/data#batching_dataset_elements

[111] F. Chollet. The Sequential model. [Online]. Available: https://www.tensorflow.org/guide/keras/sequential_model

[112] Z. Keita. Convolutional Neural Networks (CNN) with TensorFlow Tutorial. [Online]. Available: https://www.datacamp.com/tutorial/cnn-tensorflow-python

[113] Geeks For Geeks Authors. Introduction to Convolution Neural Network. [Online]. Available: https://www.geeksforgeeks.org/introduction-convolution-neural-network/?ref=lbp

[114] J. Brownlee. A Gentle Introduction to the Rectified Linear Unit (ReLU). [Online]. Available: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[115] S. Sahoo. Deciding optimal kernel size for CNN. [Online]. Available: https://towardsdatascience.com/deciding-optimal-filter-size-for-cnns-d6f7b56f9363

[116] IBM. What are convolutional neural networks? [Online]. Available: https://www.ibm.com/topics/convolutional-neural-networks

[117] G. Yatish. How to calculate the number of parameters in CNN? [Online]. Available: https://www.geeksforgeeks.org/how-to-calculate-the-number-of-parameters-in-cnn/

[118] J. Brownlee. Dropout Regularization in Deep Learning Models with Keras. [Online]. Available: https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/

[119] L. Roeder, "Netron," https://github.com/lutzroeder/netron.

[120] J. Brownlee. Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. [Online]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

[121] S. Chand. Choosing between Cross Entropy and Sparse Cross Entropy — The Only Guide you Need! [Online]. Available: https://medium.com/@shireenchand/choosing-between-cross-entropy-and-sparse-cross-entropy-the-only-guide-you-need-abea92c84662#:~:text=The%20sparse%20categorical%20cross%2Dentropy,formula%20as%20Cross%20Entropy%20Loss.

[122] V. Yathish. Loss Functions and Their Use In Neural Networks. [Online]. Available: https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9

[123] S. Saldanha. Logistic regression – sklearn (sci-kit learn) machine learning – easy examples in Python – tutorial. [Online]. Available: https://savioglobal.com/blog/machine-learning/logistic-regression-sklearn-sci-kit-learn-machine-learning-python/

[124] Scikit-Learn Authors. 3.3. Metrics and scoring: quantifying the quality of predictions. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html

[125] Scikit-Learn Developers. sklearn.metrics: precision recall fscore support. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

[126] A. Manga, “MSc-Micro-Doppler-Classification,” 2024. [Online]. Available: https://github.com/AmishaManga/MSc-Micro-Doppler-Classification

[127] S. Rao. (2018) Introduction to mmwave sensing : Fmcw radars. YouTube video. [Online]. Available: https://www.youtube.com/watch?v=8cHACNNDWD8&list=PLJAlx-5DOdeMNjpg4sRO6cty3gL_PZeCE&index=6&ab_channel=AutonomyAvenue

[128] C. Iovescu and S. Rao, “The fundamentals of millimeter wave radar sensors.”

[129] S. Rao. Lectures on Radar Systems. [Online]. Available: https://adasauto.blogspot.com/

[130] M. M. Vázquez. Basics of FMCW Radar. [Online]. Available: https://www.renesas.com/us/en/blogs/basics-fmcw-radar

[131] G. Wright. Nyquist theorem. [Online]. Available: https://www.techtarget.com/whatis/definition/Nyquist-Theorem#:~:text=The%20Nyquist%20theorem%20is%20also,at%20least%20twice%20its%20frequency.

[132] M. Liu, S. Chen, F. Lu, M. Xing, and J. Wei, “Realizing Target Detection in SAR Images Based on Multiscale Superpixel Fusion,” *Sensors (Basel, Switzerland)*, vol. 21, no. 5, p. 1643, 2021, pMID: 33652908; PMCID: PMC7956173. [Online]. Available: https://www.mdpi.com/1424-8220/21/5/1643

[133] Y. Sim, J. Heo, Y. Jung, S. Lee, and Y. Jung, “FPGA Implementation of Efficient CFAR Algorithm for Radar Systems,” *Sensors*, vol. 23, no. 2, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/2/954

[134] W. Li, J. Zhang, L. Liu, J. Zhou, Q. Sui, and H. Li, “CFAR Algorithm Based on Different Probability Models for Ocean Target Detection,” *IEEE Access*, vol. PP, pp. 1–1, 11 2021.

[135] G. Dallas. Principal Component Analysis 4 Dummies: Eigenvectors, Eigenvalues and Dimension Reduction. [Online]. Available: https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/

[136] W. Fulmyk. Data Standardization — A Brief Explanation. [Online]. Available: https://medium.com/@WojtekFulmyk/standardizing-data-for-machine-learning-2cf687e621f9

[137] Z. Jaadi. Principal Component Analysis (PCA): A Step-by-Step Explanation. [Online]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis

[138] F. Azimi and J. Jung, "Automated crop residue estimation via unsupervised techniques using high-resolution uas rgb imagery," *Remote Sensing*, vol. 16, no. 7, 2024. [Online]. Available: https://www.mdpi.com/2072-4292/16/7/1135

[139] A. Kumar. Support Vector Machine (SVM) Python Example. [Online]. Available: https://vitalflux.com/classification-model-svm-classifier-python-example/

[140] IBM Authors. What are support vector machines (SVMs)? [Online]. Available: https://www.ibm.com/topics/support-vector-machine#:~:text=What%20are%20SVMs%3F,in%20an%20N%2Ddimensional%20space.

[141] F. Luo, "Human activity classification using micro-Doppler signatures and ranging techniques," Ph.D. dissertation, Queen Mary University of London, 2020. [Online]. Available: https://qmro.qmul.ac.uk/xmlui/handle/123456789/68294

[142] N. Kumar. K-Nearest Neighbor(KNN) Algorithm. [Online]. Available: https://www.geeksforgeeks.org/k-nearest-neighbours/

[143] T. Srivastava. A Complete Guide to K-Nearest Neighbors (Updated 2023). [Online]. Available: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

[144] IBM. What is random forest? [Online]. Available: https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.

[145] G. Calmattes. Random Forests. [Online]. Available: https://brookewenig.com/RandomForests.html#/11

[146] Sruthi E R. Understand Random Forest Algorithms With Examples. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

[147] T. Gupta. Gaussian Mixture Model. [Online]. Available: https://www.geeksforgeeks.org/gaussian-mixture-model/?ref=lbp

[148] Raman. ML | Expectation-Maximization Algorithm. [Online]. Available: https://www.geeksforgeeks.org/ml-expectation-maximization-algorithm/

[149] M. Rupp. Artificial Intelligence: Hidden Markov Model Classifiers and RADAR Objects Classification by Machine Learning 2/2. [Online]. Available: https://www.skyradar.com/blog/artificial-intelligence-hidden-markov-model-classifiers-and-radar-objects-classification-by-machine-learning-2-2

[150] V. Caruana. What is a neural network and how many types are there? [Online]. Available: https://www.algolia.com/blog/ai/what-is-a-neural-network-and-how-many-types-are-there/

[151] Phung and Rhee, "A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets," *Applied Sciences*, vol. 9, p. 4500, 10 2019.

Appendix A

FMCW Radar and Signal Processing Techniques

Details of the LCR system used in this research are provided. Thereafter, a brief overview of the basic operational principles of FMCW radar, focusing on range and velocity estimation, is presented. These insights draw from a combination of sources, including a video tutorial [127], lecture materials and further content [128] provided by Rao [129]. Furthermore, the techniques used for target detection and spectrogram generation are explained.

A.1 LCR System

Table A.1 presents the radar parameters of the LCR.

TABLE A.1

Parameters of LCR system

Transmit Power	17	dBm
Start Frequency	9.5	GHz
SPI Average Data Rate	20.48	Mbps
Receiver Gain	59	dB
Maximum Measurable Target Velocity	4.93	m/s
Number of Range Bins	1008	Bins
Chirp Bandwidth	80	MHz
ADC Sampling Rate	1.4	MSps
Data Stream Sample Rate	700	kSps
Chirp Repetition Time	1.6	ms
Chirp Repetition Frequency	625	Hz
Range Resolution	1.875	m
Doppler Resolution	36.56	mm/s
Burst Size (Number of Doppler Bins)	256	Chirps
Maximum Range	1	km
Horizontal Half Power Beam Width	70°	Azimuth
Vertical Half Power Beam Width	20°	Elevation

A.2 FMCW Radar

FMCW radar systems use a frequency modulation technique where a continuous wave signal, termed a chirp, is used. The chirp is a continuous wave with a frequency that undergoes linear modulation, thereby giving rise to the term: Frequency-Modulated Continuous Wave [130].

In an FMCW system, a synthesiser is typically used to generate the chirp signal that is transmitted by the Tx Antenna. This chirp gets reflected off a target and the receive signal is then picked up by the Rx antenna as a reflected chirp signal [130]. A mixer is used to combine these signals, which results in the baseband signal. This signal is a sinusoid with frequency and phase determined by the difference of the transmitted and received signals [128]. Figure A.1 [127] illustrates the Tx and Rx chirp signals, along with the overlap time interval representing the baseband signal [127]. The frequency of this signal is directly proportional to the range of the detected target [128].

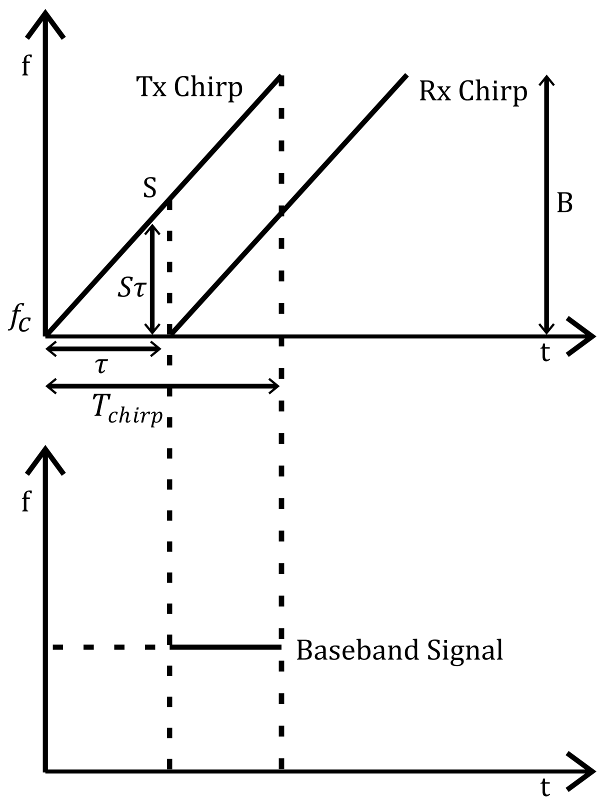


Fig. A.1. Transmitted and Received Chirp Frequencies as a Function of Time

The chirp signal can be characterised by its start frequency (f_c), bandwidth (B) and duration (T_{chirp}) [128]. The slope (S) determines the rate at which the frequency increases and tau (τ) represents the time delay between the transmitted signal and reflected signal when received back by the radar [128].

$$\tau = \frac{2d}{c} \tag{A.1}$$

Where d refers to the distance of the target and c, the speed of light.

A.2.1 Range Measurement

Range resolution is the smallest distance at which two targets of equal strength can be distinguished as separate entities [127]. Two targets can be distinguished based on their frequency if the difference between their frequencies satisfies the following condition [128]:

$$\Delta f > \frac{1}{T} \tag{A.2}$$

By considering the time delay derivation from A.1, the range resolution (Δd) can be derived as follows [127]:

$$\frac{S2\Delta d}{c} > \frac{1}{T} \Rightarrow \Delta d > \frac{c}{2ST} \tag{A.3}$$

$$\Delta d > \frac{c}{2B} \quad \text{as} \quad S = B \times T \tag{A.4}$$

Equation A.4 indicates that the range resolution is purely reliant on the bandwidth.

The cut-off frequency of the low pass filter is determined by the maximum frequency of the baseband signal, since the sampling rate of the ADC must exceed twice this frequency [131] in order to prevent aliasing while digitising the signal. The baseband bandwidth is therefore constrained by the ADC's sampling rate (F_s), which places a limitation on the radar's maximum measurable range (d_{max}).

$$F_s \geq \frac{S2d_{\text{max}}}{c} \tag{A.5}$$

$$d_{\text{max}} = \frac{F_s c}{2S} \tag{A.6}$$

A.2.2 Velocity Measurement

The baseband signal in the frequency domain, i.e. after performing the range-FFT, is complex, with each value representing as a phasor containing amplitude and phase components [128]. The phase of the baseband signal at the mixer output is the difference between the phases of the two input signals [127, 128].

If the object moves slightly by $\Delta\tau$, the phase difference ($\Delta\phi$) can be derived as [127, 128]:

$$\Delta\phi = 2\pi f_c \Delta\tau = \frac{4\pi\Delta d}{\lambda} \quad \text{as} \quad c = \lambda \times f \tag{A.7}$$

The baseband signal for a target detected at a certain range, d, from the radar, can be represented mathematically as a sinusoid as shown below [128]:

$$A\sin(2\pi f t + \phi_0) \quad \text{where} \quad f = \frac{S2d}{c} \quad \text{and} \quad \Delta\phi = \frac{4\pi\Delta d}{\lambda} \tag{A.8}$$

In order to measure a moving targets velocity, the transmitter is designed to transmit frequency modulated chirps separated by a fixed time period, T_{chirp} [128] as depicted in Figure A.2.

A range FFT is performed on each of the reflected chirp signals in order to detect the range. The phase difference per chirp repetition interval, ω, corresponds directly to the targets velocity, v, and can be derived as follows [128]:

$$\omega = \frac{4\pi v T_{\text{chirp}}}{\lambda} \Rightarrow v = \frac{\lambda\omega}{4\pi T_{\text{chirp}}} \tag{A.9}$$

The velocity measurement will be ambiguous [128], as it originates from the phase difference, ω. In the case of positive velocity, the phase shift occurs in the counter clockwise direction. For negative velocity, the shift moves in a clockwise direction.

For an unambiguous velocity measurement, the phase difference must be less than π radians [128]:

$$\frac{4\pi v T_{\text{chirp}}}{\lambda} < \pi \Rightarrow v < \frac{\lambda}{4T_{\text{chirp}}} \tag{A.10}$$

The maximum relative speed, v_{max}, that can be measured after transmitting two chirps is [128]:

$$v_{\text{max}} = \frac{\lambda}{4T_{\text{chirp}}} \tag{A.11}$$

The above phase comparison has limitations when calculating the velocities of multiple targets that move within the same range because the frequencies from the reflected signals are expected to be almost identical. However, the phase components will be intertwined or superimposed within the reflected signals of each target [127, 128].

To measure velocities of multiple targets, a series of equally spaced chirps, N, often referred to as a burst, is transmitted. Refer to Figure A.2 [127] for an illustration. The range FFT performed on this set of chirp signals creates N frequency peaks [127, 128]. The discrete sequence associated with each frequency peak contains a unique phase element, which includes the phase contributions from the reflected targets [127, 128].

A second FFT, commonly referred to as the Doppler FFT, is then performed on these N phasors to calculate the corresponding discrete angular frequencies $\omega_1, \omega_2, \ldots, \omega_N$ [127]. The corresponding velocities $v_1, v_2, \ldots, v_N$ are then deduced from these angular frequencies, as seen in Equation A.12, which corresponds to the phase difference between consecutive chirp signals for the respective targets [127, 128].

$$v_i = \frac{\lambda \omega_i}{4\pi T_{\text{chirp}}} \tag{A.12}$$

In order to distinguish these angular frequencies, $\omega_1, \omega_2, \ldots, \omega_N$, the separation $\Delta\omega = \omega_j - \omega_i$ must be greater than $\frac{2\pi}{N}$ radians per sample [128, 131], therefore:

$$\Delta\omega > \frac{2\pi}{N} \quad \text{and} \quad \Delta\omega = \frac{4\pi v T_{\text{chirp}}}{\lambda} \tag{A.13}$$

$$\Delta v > \frac{\lambda}{2NT_{\text{chirp}}} \quad \text{as} \quad v = \frac{\lambda\omega}{4\pi T_{\text{chirp}}} \tag{A.14}$$

where $N \times T_{\text{chirp}}$ refers to the burst time (T_{burst}). The velocity resolution (v_{res}) of the radar is inversely proportional to the burst time and is represented by [128]:

$$v_{\text{res}} = \frac{\lambda}{2T_{\text{burst}}} \tag{A.15}$$

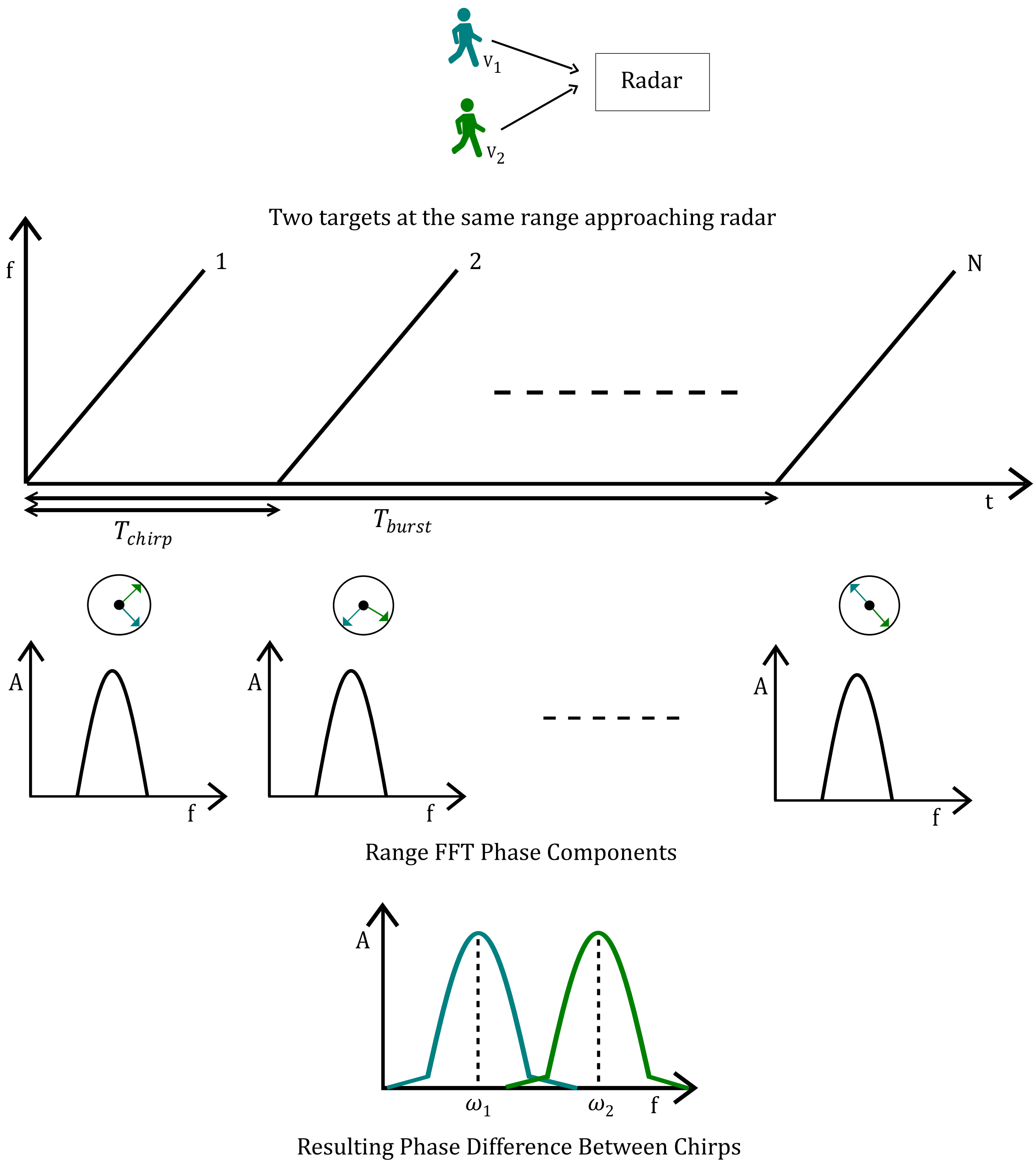


Fig. A.2. Velocity Measurement for Multiple Objects Moving at the Same Range

A.2.3 Range-Doppler Map

The range-Doppler map provides a useful visual representation of the detected targets, as it combines information from two dimensions: range, which is the distance between the radar and the target, and Doppler frequency, which represents the target's radial velocity [128,130].

Figure A.3 [127, 130] provides an illustration of multiple chirps corresponding to one burst. The captured ADC samples, relating to each individual chirp, can be represented as rows within a matrix. A range FFT is performed on each individual row of the matrix in order to identify the target range [128, 130].

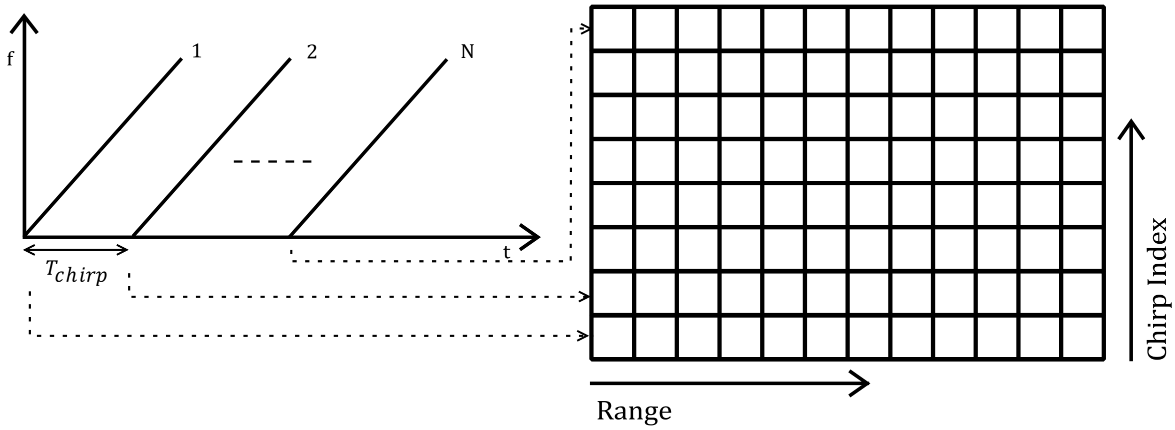


Fig. A.3. Range FFT

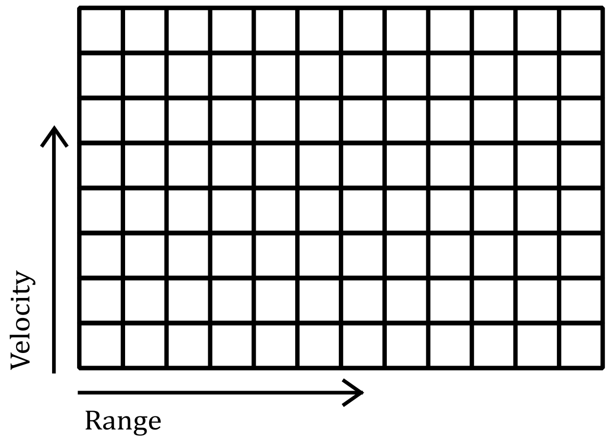


Fig. A.4. Range-Doppler FFT Map

Following the range FFT, the Doppler FFT is performed along the columns of the range FFT output matrix. The Doppler FFT can only be done once the range FFT has been performed across the full burst. The y-axis represents the discrete angular frequencies from the Doppler FFT, which are directly proportional to the target velocity [128].

Performing the range FFT followed by the Doppler FFT results in what is known as the range-Doppler map, as shown in Figure A.4 [127, 130]. Each point on this map corresponds to a specific target and its associated range-velocity information [130].

A.3 CFAR Target Detection

As radar systems are often deployed in non-homogeneous environments, the detection of targets against unwanted radar returns, commonly referred to as clutter and noise, can present a challenge.

Due to the benefits of low cost computation and adaptive thresholds, the CFAR method is favoured among the numerous types of target detection techniques available [132]. CFAR algorithms, firmly grounded in statistical analysis, demonstrate high adaptability in adjusting detection thresholds based on the local statistical properties of the surrounding clutter [88, 133]. The CFAR's dynamic threshold adjustment focuses on maintaining the desired Probability of Detection, P_{D} and reduces the Probability of False Alarm, P_{fa}, rate [89, 133].

CFAR algorithms work on the principle of evaluating the energy or power levels in the radar return signals. By statistically analysing the vicinity of the CUT, which represents the target detection location, a threshold is computed to discriminate true targets from clutter [133]. Signal amplitudes exceeding the threshold are identified as targets, while those below it are classified as noise. Refer to Figure A.5 for an illustration of this.

The target detection threshold is adapted by analysing the local statistical data within a moving window across the range or Doppler dimensions. This approach surveys the neighbourhoods of power returns in the noise or clutter mean estimate [89]. By proportionally adjusting this estimated mean with a predetermined multiplier, the threshold is set to restrict false alarms to a desired rate [89].

Mathematically, the detection threshold (T) is determined as [91]:

$$T = \alpha \cdot P_n \tag{A.16}$$

where P_n is the noise power estimate, with α representing the CFAR threshold factor. Proper α selection maintains a CFAR despite varying external interference intensities [134].

One of the ways that the threshold factor can be calculated is by "assuming the background clutter conforms to a Gaussian distribution model" [134]. The Gaussian distribution is a popular choice among probability distribution models due to its frequent utilisation [134]. CFAR detectors aim for high probability density, typically exceeding 80%, while keeping the P_{fa} in the range of 10^{-5} to 10^{-6} [89, 90]. Optimising α requires a careful balance between minimising false alarms and maximising target detection [88–90, 134].

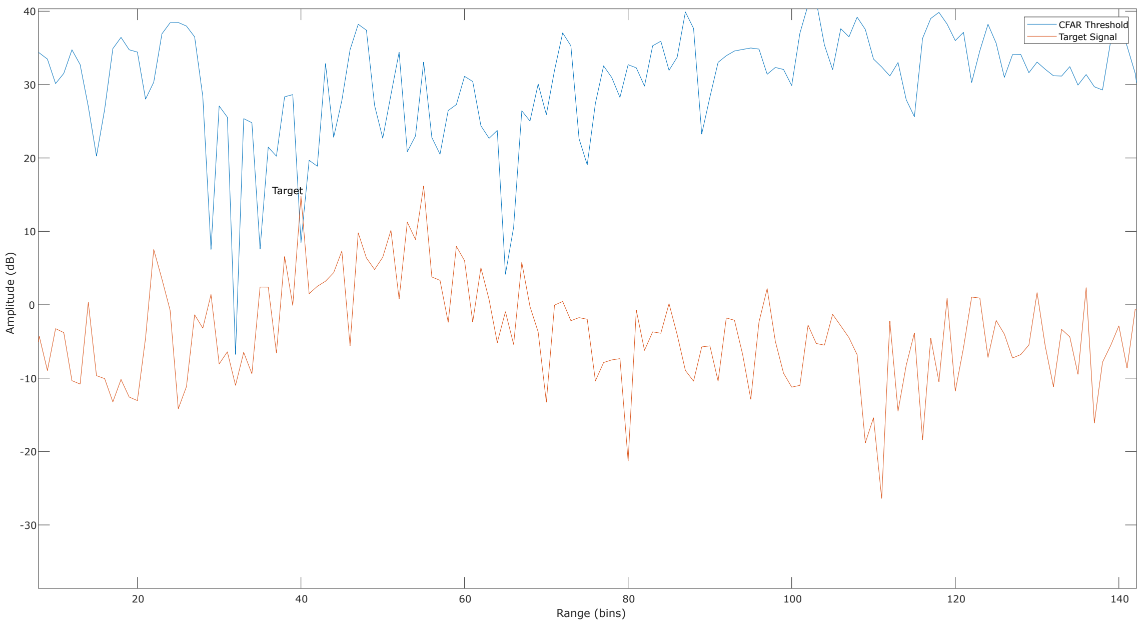


Fig. A.5. CFAR Detection with Range Threshold and Target Signal

A.3.1 CA-CFAR

One of the commonly employed CFAR techniques is the CA-CFAR. This detector involves the estimation of noise power in the vicinity of the CUT by examining surrounding cells [89]. The noise samples are extracted from both leading and lagging windows around the CUT [89]. The window of neighbouring cells consists of guard cells surrounded by training cells (N), as illustrated in Figure A.6 [91]. These training cells serve as references for the noise characteristics of the surrounding clutter, enabling the radar's processing framework to adjust its threshold accordingly [134]. Additionally, guard cells are strategically placed adjacent to the CUT in order to prevent interference from signal components leaking into the training cells [91].

Once the background noise power is estimated, the final threshold is computed by scaling the background noise power to achieve the desired false alarm rate. The output of the CA-CFAR detector is this threshold, which is then used to make the decision regarding whether or not there is a target [91, 134].

The noise estimate, P_n, is computed as follows [91]:

$$P_n = \frac{1}{N}\sum_{m=1}^{N} x_m \tag{A.17}$$

where x_m is the sample in each training cell, N [91]. By assuming the data entering the detector is from a single burst, the threshold factor is calculated as follows [91]:

$$\alpha = N\left(P_{fa}^{-\frac{1}{N}} - 1\right) \tag{A.18}$$

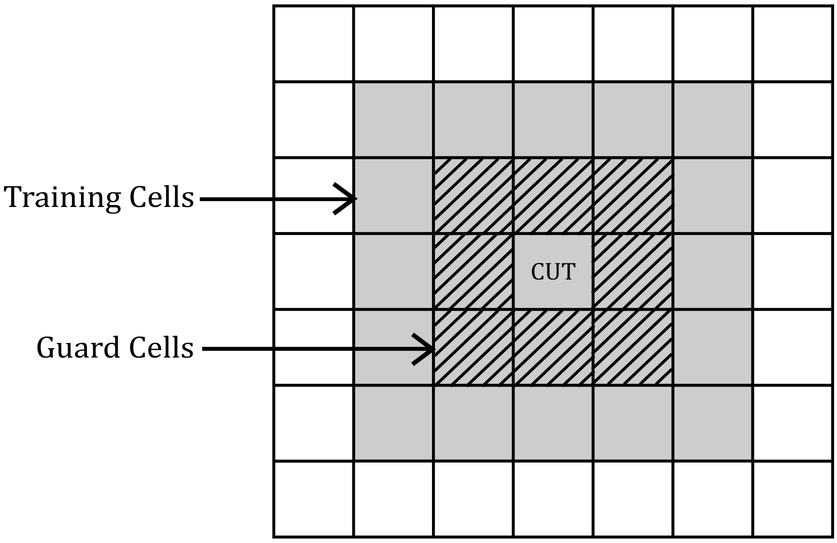


Fig. A.6. CA-CFAR Detection for Range-Doppler Maps

A.4 Spectrogram Generation

The following code listing presents a high-level overview of the signal processing workflow, starting with reading data from the log file to preparing the data in order to perform the STFT on the signal.

```
% While the log file is open
while feof(fileId)~= 1
        % Extract timestamp and read data continously from file 1 burst at time
        timestamp = uint8(fread(fileId, 12, 'uint8'));
        rawData = uint32(fread(fileId, (burstLength), 'uint32'));
        % Reshape data and extract data from channels
        rawData = reshape(rawData, 1024, 256)';
        RPDSheader = rawData(:,   1:16);
        Ch1Data  = rawData(:,  17:520);
        Ch2Data  = rawData(:, 521:1024);
        % Add channels after range FFT: Can simply add, no phase shift
        rangeFFT1 = doRangeFFT(Ch1Data);
        rangeFFT2 = doRangeFFT(Ch2Data);
        rangeFFT = rangeFFT1 + rangeFFT2;
        % Extract signal from range bin where there is a target
        % "Bin" is the predefined range bin data corresponding to the log file
        signalWithTarget = rangeFFT(:,Bin(targetIdx));
        % Prepare input signal for STFT by concatenating the signal
        inputSignal = cat(1, inputSignal, signalWithTarget);
        % Calculate the number of bursts
        numberOfBursts = numberOfBursts +1;

        % Interval for STFT analysis
        if (numberofBursts == burstCountInterval)
                % Do spectrogram then reset the spectrogram matrix
                SpectrogramMatrix = Specgram(inputSignal);
                SpectrogramMatrix = [];
                % Reset the no. of bursts count
                numberOfBursts = 0;
        end

        % Iterate through predefined range bins containing target
        targetIdx = targetIdx + 1;
end
```

The following code listing demonstrates the function for computing the STFT and the method used to generate the spectrograms in this research.

```
% Determine the total length of the input signal
signalLength = length(inputSignal);
% Initialize the STFT result as an empty matrix
STFT = [];
% Calculate the total number of segments to analyse, accounting for overlap
if (overlap == 0)
        totalSegments = floor(signalLength/windowLength);
else
        segmentsFromOverlap = floor(signalLength/overlap);
        totalSegments = segmentsFromOverlap  - floor(windowLength/overlap);
end
% Loop through each segment to compute its Fourier Transform
for segIndex = 1: totalSegments
        % Calculate the start and end indices of the current segment
        segStart = (segIndex-1)*overlap + 1;
        segEnd = segStart + windowLength -1;
        % Extract the segment of the signal
        fftSeg = inputSignal(segStart:segEnd);
        % Apply the window function to the segment: use other window functions
        % for data augmentation here:
        windowedSegment = Hann(length(fftSeg))';
        windowedSegment = windowedSegment / sum(windowedSegment);
        % Compute the Fourier Transform of the windowed segment and centre the zero frequency
        STFT(:,segIndex) = fftshift(fft(fftSeg.*windowedSegment, nfft));
end
% Assign the computed STFT
computedSpectrogram = STFT;
% Compute the magnitude of the STFT
magnitudeSTFT = abs(computedSpectrogram);
% Find the maximum magnitude value for normalization
maxMagnitudeSTFT = max(max(magnitudeSTFT));
% Apply logarithmic scaling and normalization, then plot the spectrogram using imagesc
spectrogramData = imagesc(20 * log10(magnitudeSTFT / maxMagnitudeSTFT));
% Adjust color scale limits to specified dynamic range
set(gca, 'CLim', [DynamicRangeLowerLimit, DynamicRangeUpperLimit]);
```

Appendix B

Classification Techniques

This appendix provides background information on the techniques employed when performing the micro-Doppler classification in this research. The basic foundational concepts of Principal Component Analysis, Support Vector Machines, k-Nearest Neighbors, Hidden Markov Models, Gaussian Mixture Models, Random Forests and Convolutional Neural Networks are explored and presented.

B.1 Principal Component Analysis

PCA is applied in the following steps [30, 34, 135]:

- **Standardisation:** The first step in PCA is to standardise the dataset. This means scaling each feature so that they all have a mean of zero and a standard deviation of one [136].
- **Covariance Matrix Computation:** The covariance matrix captures the variance and the relationship between different variables in the dataset in order to understand how much the dimensions vary from the mean with respect to each other [137].
- **Eigenvalue and Eigenvector Calculation:** The dataset is decomposed into eigenvectors and eigenvalues, which exist in pairs: each eigenvector is associated with a corresponding eigenvalue [137]. An eigenvector represents a specific direction within the data space (e.g., vertical, horizontal, or at a particular angle). The associated eigenvalue quantifies the variance in the dataset along that particular eigenvector's direction [137]. This quantification is a reflection of the data spread along the line defined by the

eigenvector. The eigenvector that corresponds to the highest eigenvalue is designated as the principal component, indicating the direction of maximum variance within the dataset [34, 137]. In PCA, the eigenvectors are orthogonal to each other.

- **Sorting Eigenvalues and Eigenvectors:** The eigenvalues are sorted in descending order and their corresponding eigenvectors are aligned in the same order. This step helps in understanding the components (directions in the feature space) that account for the most variance [137].
- **Projection Onto New Feature Space:** The data is then projected onto the new feature space. This is done by selecting the top k eigenvectors (where k is the number of dimensions to keep) and using them to transform the original n-dimensional data into a new k-dimensional feature space [34, 137].

PCA operates by transforming the data into a new coordinate system, where the axes (principal components) are oriented in the direction of maximum variance, with the first principal component capturing the most significant variance and each subsequent component capturing the next most significant variance orthogonal to the previous components [137,138].

B.2 Support Vector Machines

SVMs are designed to find an optimal hyperplane that separates two sets of data that can be linearly separated [35]. When the data is linearly separable, as illustrated in Figure B.1 [139], there are two parallel hyperplanes that effectively separate the two classes of data by maximising the distance between them. The region enclosed between these hyperplanes is known as the margin [140]. The most *optimal* hyperplane within this margin is positioned equidistant from both hyperplanes and is referred to as the maximum-margin hyperplane [141].

However, when the data is not linearly separable, SVMs employ a combination of a maximal margin strategy and a kernel method [29] to estimate the optimal boundary in a higher-dimensional feature space.

To achieve this, SVMs use kernel functions to transform each instance of input data into a higher-dimensional space [139, 140]. This transformation allows the data points to become linearly separable, enabling the identification of an optimal hyperplane [28]. The dimensionality of this feature space is determined by the number of support vectors extracted from the training data [140]. These support vectors play a crucial role in determining the decision boundaries of the SVM [29].

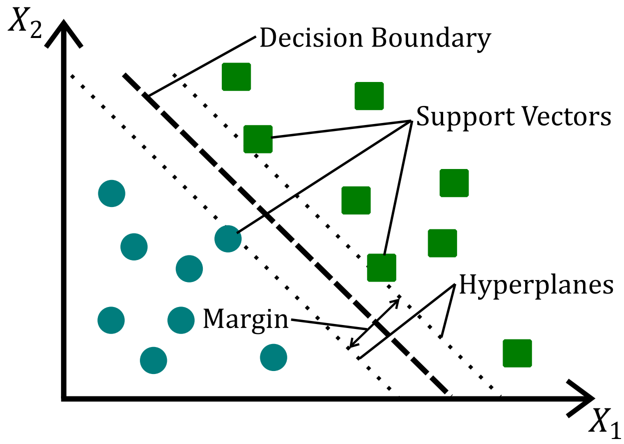


Fig. B.1. Support Vector Machine

B.3 k-Nearest Neighbors

The kNN algorithm is a non-parametric algorithm [142], as it does not make any explicit assumptions about the underlying data distribution.

In the case of kNN, the algorithm stores the full training dataset to act as a reference point [143]. When making predictions, it calculates the distance between the input data point and all the training examples based on a distance metric, which is commonly based on a Euclidean distance [142].

The number of neighbors, k, is a hyperparameter that needs to be set before making predictions, but it is not fixed across different datasets or model instances. When a new data point needs to be classified, kNN finds the nearest neighbors from the training dataset and classifies data points into groups identified by an attribute. Choosing the appropriate value of k is essential in kNN. A small k value may result in overly complex decision boundaries, leading to overfitting and unstable predictions [142]. On the other hand, a large k value may lead to overly simplified decision boundaries, reducing prediction accuracy [143]. Typically, an odd value for k is chosen in order to avoid ties when voting for the class of a data point.

The algorithm can be visualised by an example presented in Figure B.2 [143]. The data point that needs to be classified is depicted as a star. Assuming the value of k is set to 3, the 3 closest data points to the star are indicated by circles [143]. As a result, the algorithm classifies the star data point as belonging to the circle class as it is the majority class.

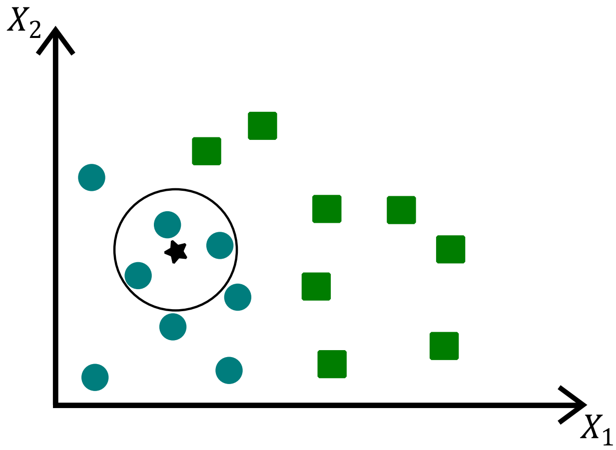


Fig. B.2. k-Nearest Neighbor Example

B.4 Random Forests

The RF algorithm is a supervised machine learning algorithm, developed by Leo Breiman and Adele Cutler [144]. It functions as an ensemble technique that combines the output of multiple decision trees in order to reach a single outcome [144].

These individual trees make decisions regarding classification of new data by leveraging features learned during the training stage. Input data for the RF classifier is processed through each constituent decision tree. Each tree autonomously predicts a classification and contributes a “vote” for the corresponding category [48]. The final classification for the classifier’s input is determined by the majority consensus of these votes [48]. This is illustrated in Figure B.3 [145].

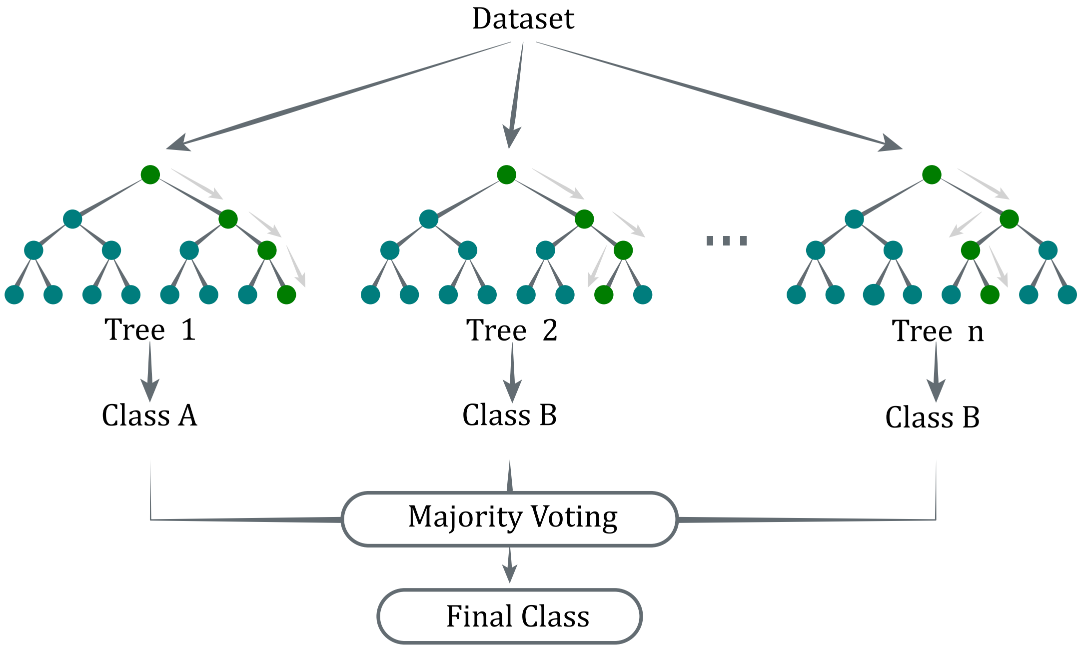


Fig. B.3. Random Forests

Ensemble learning methods combine the classification predictions from multiple models in order to facilitate better performance. Bagging, also known as bootstrap aggregation [146], allows separate decision trees to randomly select and replace data from the dataset, resulting in diverse outcomes for each tree. Instead of using all the available data, each tree uses only a portion of the data. Consequently, these trees make decisions based solely on the data they possess, predicting outcomes accordingly.

In a RF, this implies that each tree is trained on different data subsets and employs different features to make decisions [146]. This approach mitigates overfitting, bias and variance risks, thereby ensuring more precise predictions [48].

B.5 Hidden Markov Model

A Markov Model, named after Andrey Markov [53], is a probabilistic framework used to represent and analyse stochastic systems or processes, capturing dependencies that gradually approach a steady-state over time [52].

Functioning as finite-state machines [53], markov models predict the future state of a system by drawing on information from its current state. Each state holds a probability of transitioning to any other state after a single step, resulting in a sequential arrangement of states where the system dynamically shifts between them over time, governed by specific probabilities [53]. The transitions between states exhibit a “memoryless” [52] characteristic, implying that the historical path leading to the present state is discarded.

Markov models are categorised into two types: hidden and observable. Observable models, also referred to as Markov chains, focus solely on states that can be directly observed without any involvement of hidden variables [53]. It is represented as a sequential series of interconnected states, where it transitions between states with specific probabilities over time, forming a chain [52]. This is illustrated in Figure B.4 [52].

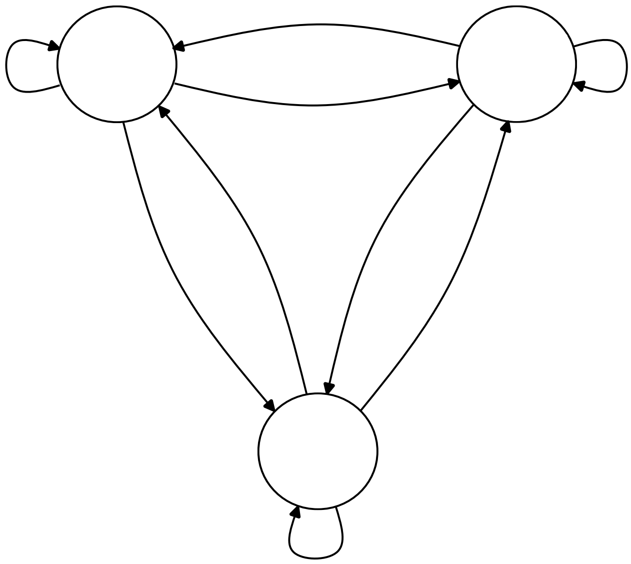


Fig. B.4. Markov Chain Example

HMMs contain hidden states that cannot be seen directly but can be inferred from other observations. Hidden states, or the subsequent potential variables that cannot be directly observed, are inferred through the Markov assumption, where a hidden variable relies solely on the preceding hidden state [53]. Mathematically, the probability of being in a state at time t depends exclusively on the state at time $t-1$ [53]. Additionally, HMMs assume that the conditional distribution over the next state, given the current state [53], remains constant over time [53].

A HMM is comprised of the following components: an initial probability distribution, one or more hidden states, a transition probability distribution represented by a matrix showing hidden state-to-hidden state transition probabilities, a sequence of observations [52], and emission probabilities reflecting the likelihood of an observation being generated from a specific state [53]. These components collectively facilitate the estimation of hidden states based on observable outputs [52, 53].

While a Markov chain deals with observable states and converging probabilities over time, HMMs are designed to model situations where the states are not directly observable [52]. Instead, they are associated with a sequence of observable events or emissions. This makes HMMs particularly well-suited for scenarios where there is uncertainty about the underlying state, but some information can be obtained through measurements or observations [52, 53].

B.6 Gaussian Mixture Models

The GMM is a probabilistic framework that is used for clustering and density estimation [61]. It operates on the principle that data arises from a blend of Gaussian distributions, each representing a unique cluster [61, 63]. GMM's distinctive feature lies in its capability to assign probabilities to data points, accommodating their simultaneous membership to multiple clusters [61].

The GMM relies on multiple Gaussian distributions to group similar data points [61]. It leverages a mix of Gaussian probability distributions, necessitating the estimation of mean and standard deviation parameters for each distribution [62, 63].

Consider, for instance, four Gaussian distributions: GD1, GD2, GD3 and GD4, each defined by its mean $(\mu_1, \mu_2, \mu_3, \mu_4)$ and variance $(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ [61]. Given a dataset, GMM calculates the likelihood of each data point belonging to these distributions [61].

The Probability Density Function (PDF) of a Gaussian Distribution in a one dimensional space is represented as [61]:

$$G(X|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{B.1}$$

where, μ denotes the mean, and σ^2 signifies the variance [61].

While the Maximum Likelihood Estimation (MLE) efficiently estimates parameters for a single distribution [147], in reality clusters often arise from multiple distinct Gaussian Distributions.

For Multivariate Gaussian Distribution, the PDF is defined as [147]:

$$G(X|\mu, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (x-\mu)} \tag{B.2}$$

where μ represents the d-dimensional mean of the distribution and Σ stands for the $d \times d$ covariance matrix [147].

In a dataset containing d features, there's a blend of k Gaussian distributions (where k represents the number of clusters), each possessing a specific mean vector and variance matrix [61]. Gupta *et al.* in [147] elaborate on why MLE becomes ineffective in estimating these parameters.

As such, the Expectation-Maximization (EM) algorithm [62,148,149] often emerges as a good approach to estimating the parameters of these distributions [62].

B.7 Neural Networks

NNs are a fundamental component of deep learning algorithms. They consist of layers of interconnected nodes, including an input layer, hidden layers and an output layer [150]. Each node processes data, applies learned weights and makes decisions based on threshold values [116]. Activation of a node occurs when its output is above a certain threshold, facilitating the transmission of data to the next layer in the network [116]. If the output remains below the threshold, data transmission to the next layer is inhibited [116]. A Basic NN Architecture is depicted in Figure B.5 [65].

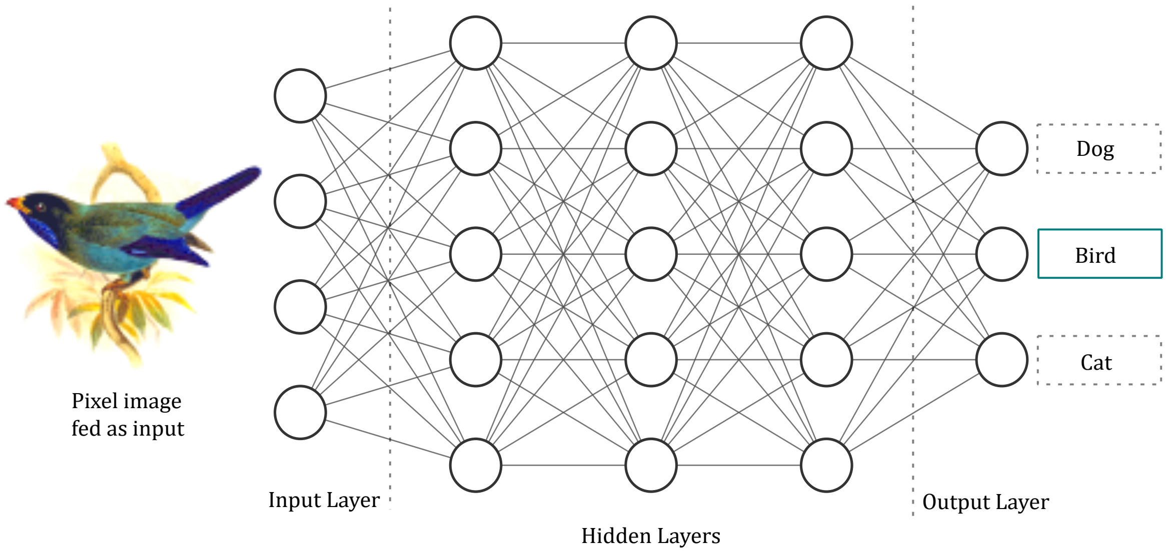


Fig. B.5. Neural Network Architecture

B.7.1 Convolutional Neural Networks

CNNs are a class of deep neural networks that have garnered considerable attention due to their performance when processing data with a grid-like topology, such as images [64]. They are able to capture the spatial and temporal characteristics in an image effectively through the application of the relevant filters [64].

CNN's are comprised of three main layers [116]:

1. **Convolutional Layer:** The convolutional layer is the fundamental component of a CNN, tasked with the bulk of the computational processing. In the initial phase of a convolutional layer, the component responsible for the convolution operation is known as the kernel or filter [64]. The objective of the convolution process is to extract high level features such as edges and textures from the input [64, 116].

   The convolutional layer applies a set of these learnable filters to the input [113]. Each filter in a convolutional layer is spatially small (along width and height), but extends through the full depth of the input volume [113].

   The filter is applied to a section of the image and a dot product is calculated between the filter and the input pixels [116]. The result of this dot product is placed into an output array [116]. Following this, the filter shifts across the image by a specific stride value and repeats the process [65], ultimately scanning the entire image [116]. The accumulated results of these dot products form what is known as a feature map, activation map or convolved feature [116].

   After the convolution process, a ReLU activation function [114] is typically applied to the feature map, introducing non-linearity and enhancing the CNN's ability to learn complex patterns.

2. **Pooling Layer:** Pooling layers, also known as down sampling, play an important role in reducing the dimensionality of the input, thereby decreasing the number of parameters [116]. Unlike convolutional layers, the filters in pooling layers are weightless [116] and employ aggregation functions like max or average pooling. Max pooling selects the highest value from the receptive field for the output array, while average pooling computes the average value [65]. This process, is useful for extracting dominant features [64] and significantly benefits the CNN by simplifying its complexity [116].

3. **Fully-Connected Layer:** The fully-connected layer is the layer where each node is connected directly to the previous layer [116] and performs the final classification task [113] using techniques like softmax classification [64, 112].

Figure B.6 provides an illustration of the basic CNN Architecture [151].

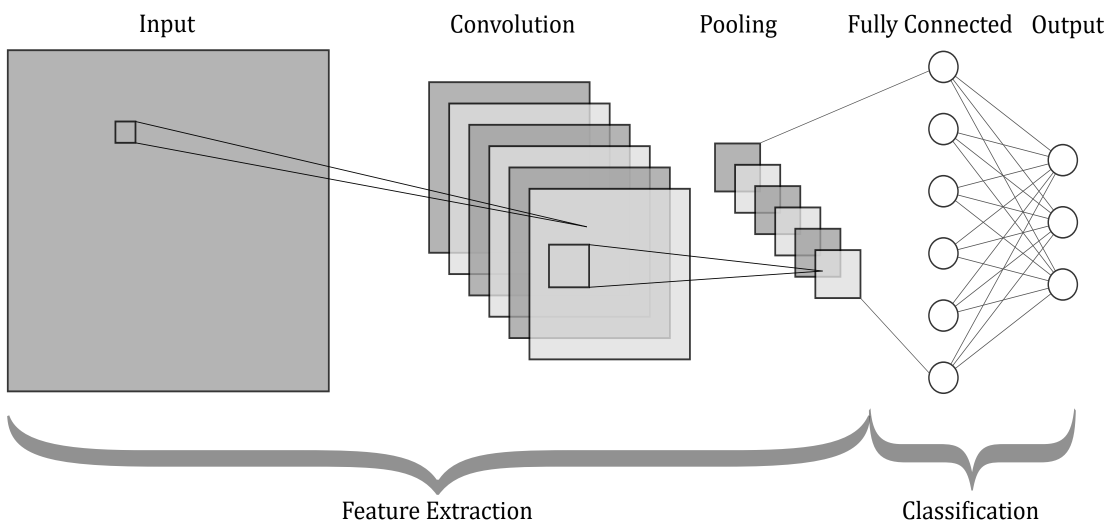


Fig. B.6. Basic CNN Architecture

Appendix C

Photographs of Data Capture Sessions

This appendix contains all the photographs from the data collection activities undertaken to compose the dataset for this study.

Fig. C.1. RRS Rooftop Radar Setup

Fig. C.2. RRS Parking Radar Setup

(a) Walking: Hand in Pocket

(b) Walking Normally

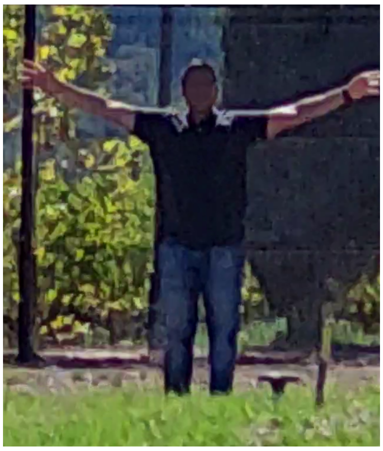
(c) Walking: Hands in Pocket

(d) Standing and Waving

(e) Walking with Hammer

(f) Ducking Up and Down

(g) Running



(h) Waving with Both Arms

(i) Walking: Swinging Arms

Fig. C.3. Snapshots from Person Dataset Captured at RRS

Fig. C.4. Radar Setup at Paardevlei Old Industrial

Fig. C.5. Golden Retriever - Link; Link Walking Toward Radar; Link Running Away from Radar

Fig. C.6. Border Collie x Staffordshire Bull Terrier - Loki; Loki Walking Away from Radar; Loki Walking Toward the Radar

Fig. C.7. Pinto Stables: Radar Recording Data of Herd of Horses

Fig. C.8. Pinto Stables: Radar Recording Data of Stallion - Seun

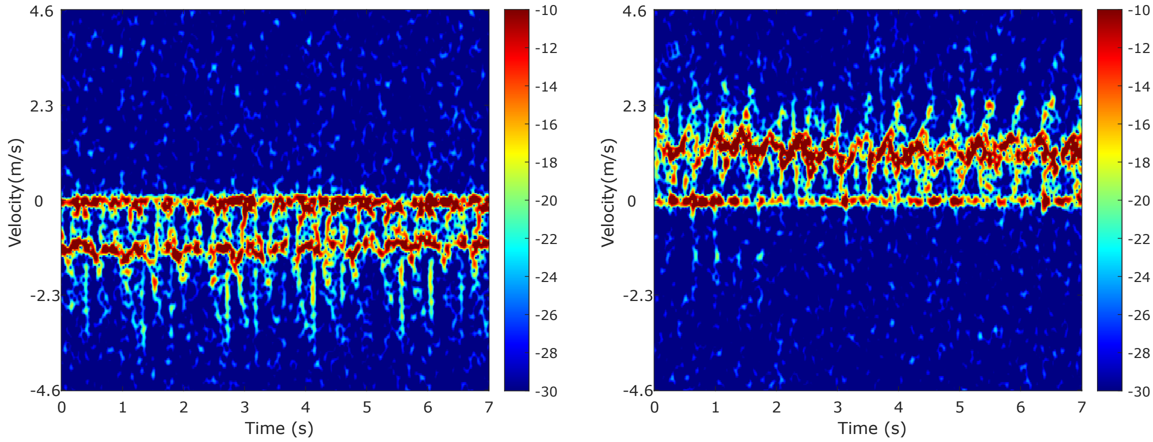
Fig. C.9. Yonder Hill Radar Setup

Appendix D

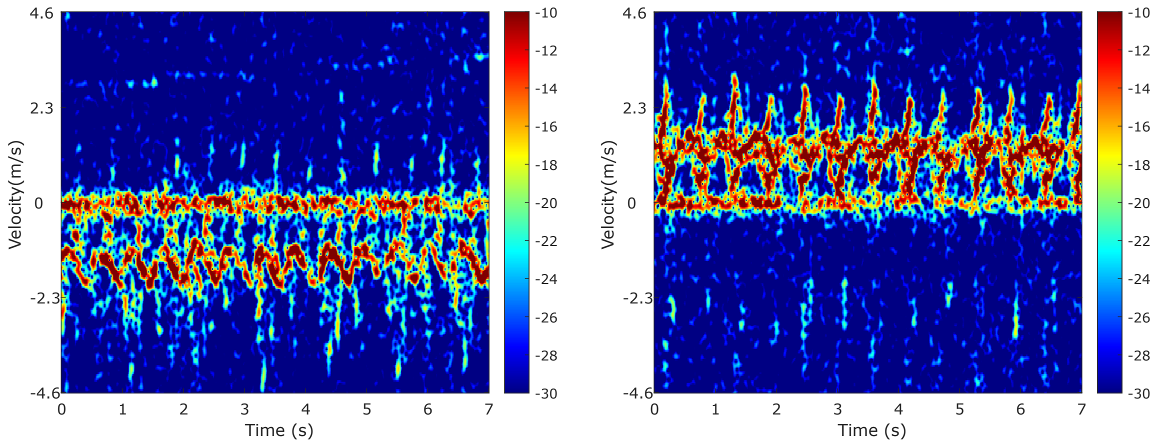
Spectrogram Images of Human and Animal Targets

This appendix presents a selection of the spectrogram images generated for each type of target subject, namely:

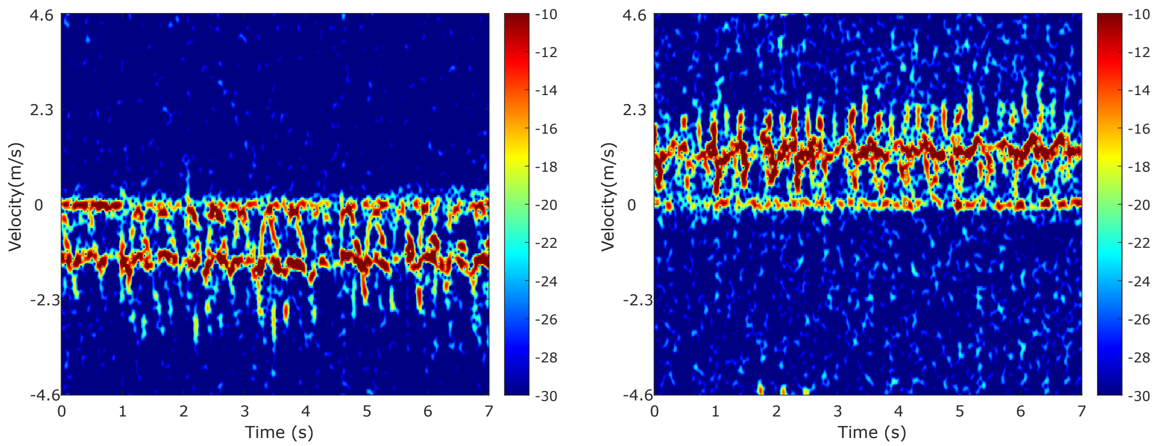
- Humans
- Dogs
- Horses
- Cows



(a) Human Walking Normally, Away From (left) and Towards Radar (right)

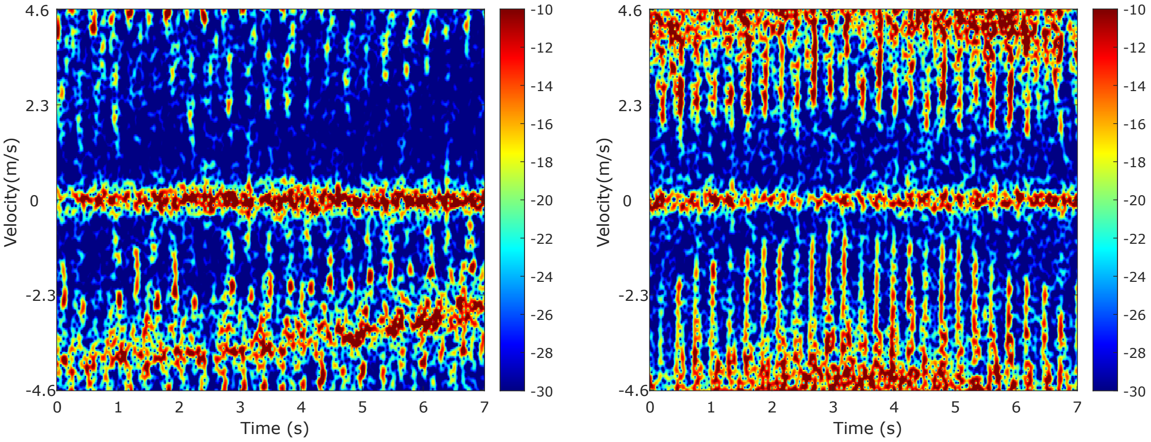


(b) Human Walking with Arms Swinging, Away From (left) and Towards Radar (right)

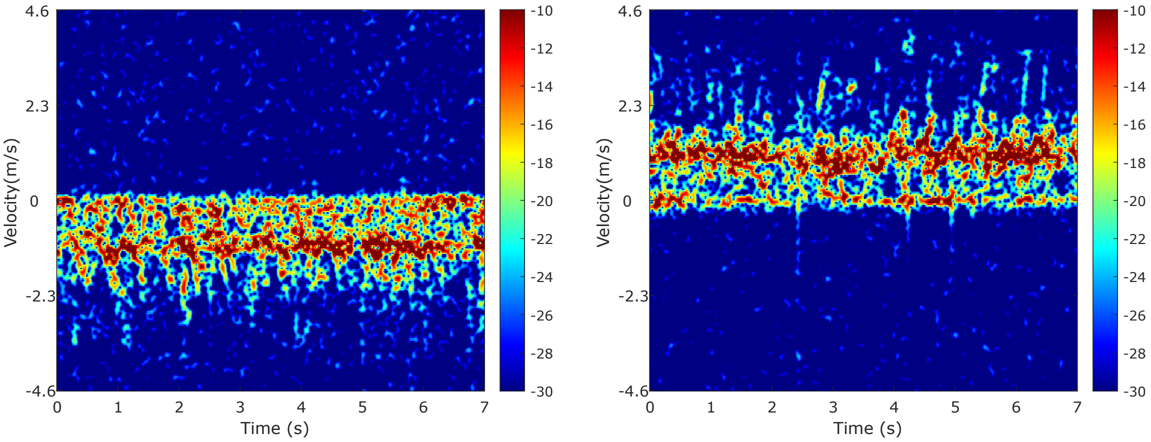


(c) Human Walking with Arms in Pocket, Away From (left) and Towards Radar (right)

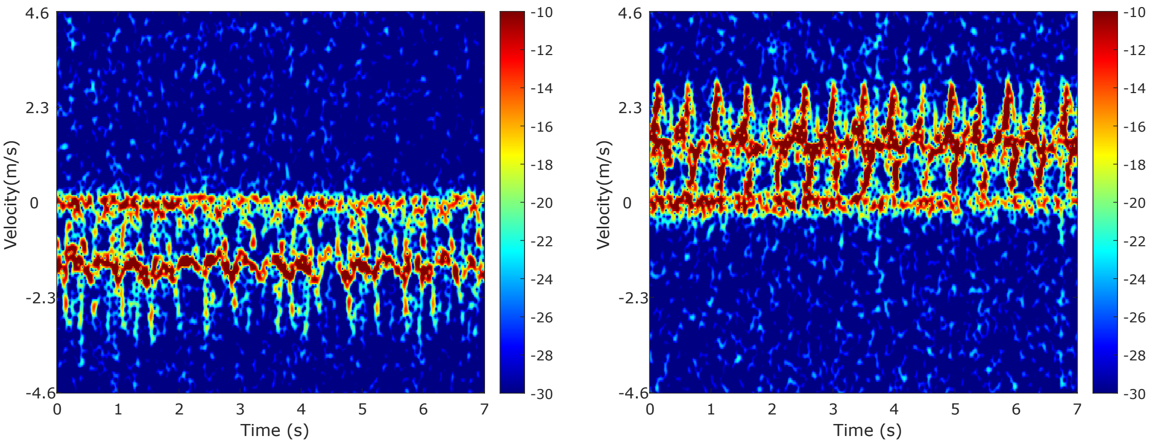
Fig. D.1. Spectrograms of Different Human Movements Part 1



(a) Human Running, Away From (left) and Towards Radar (right)

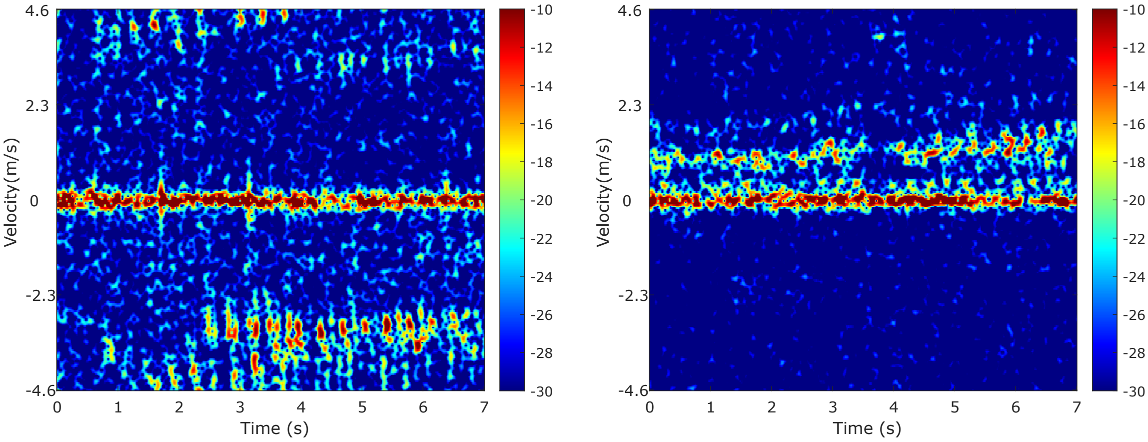


(b) Two Humans Walking, Away From (left) and Towards Radar (right)

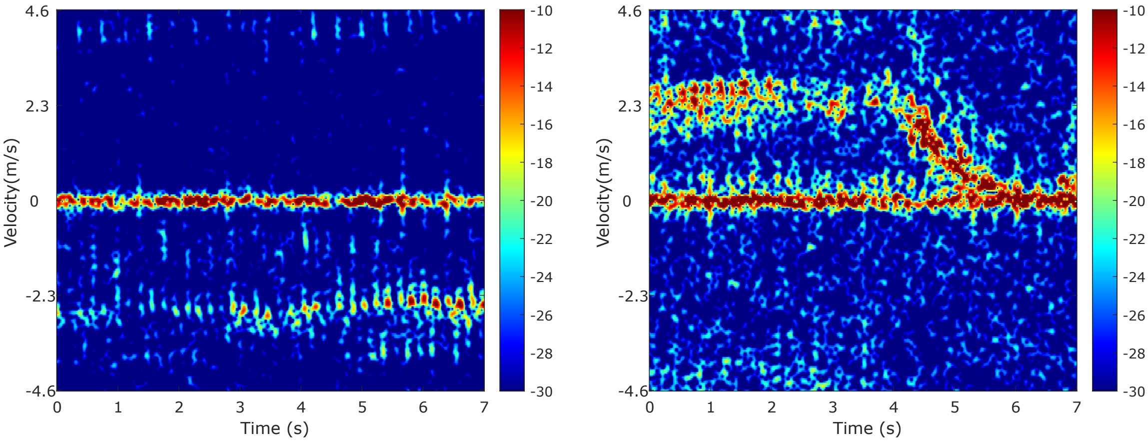


(c) Human Walking with Weapon, Away From (left) and Towards Radar (right)

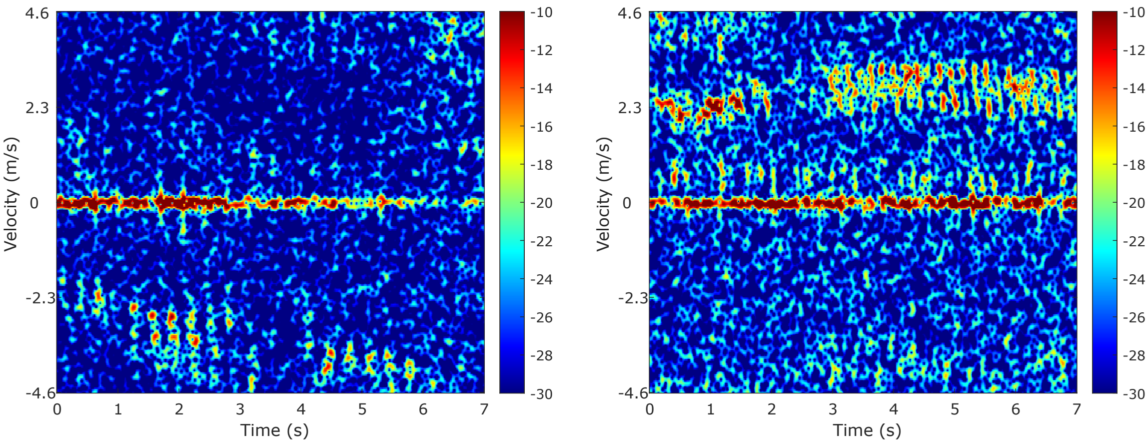
Fig. D.2. Spectrograms of Different Human Movements Part 2



(a) Link Running, Away From (left) and Towards Radar (right)

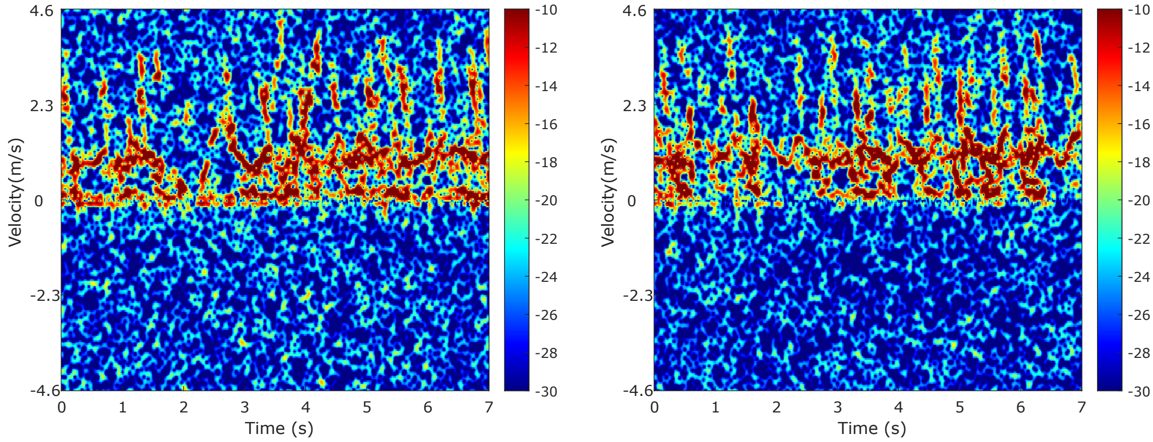


(b) Link Walking, Away From (left) and Towards Radar (right)

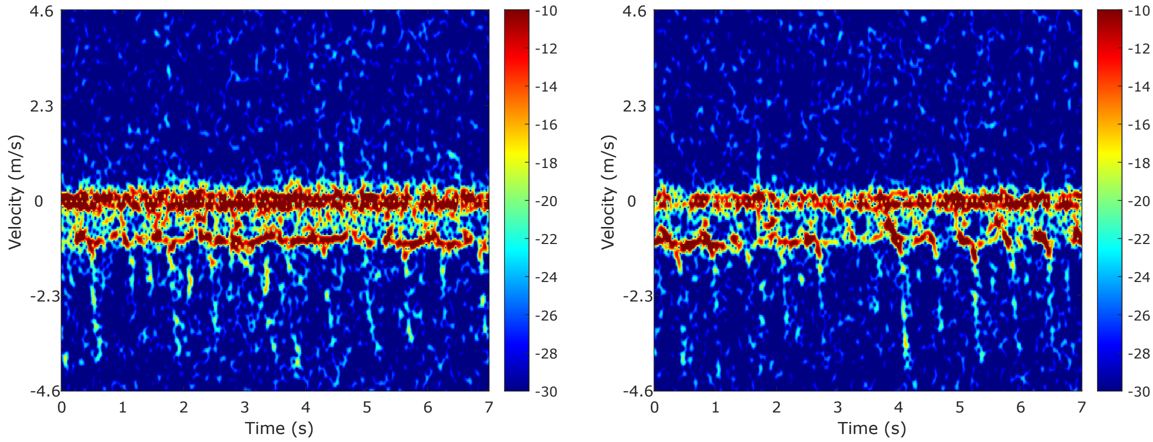


(c) Loki Walking, Away From (left) and Towards Radar (right)

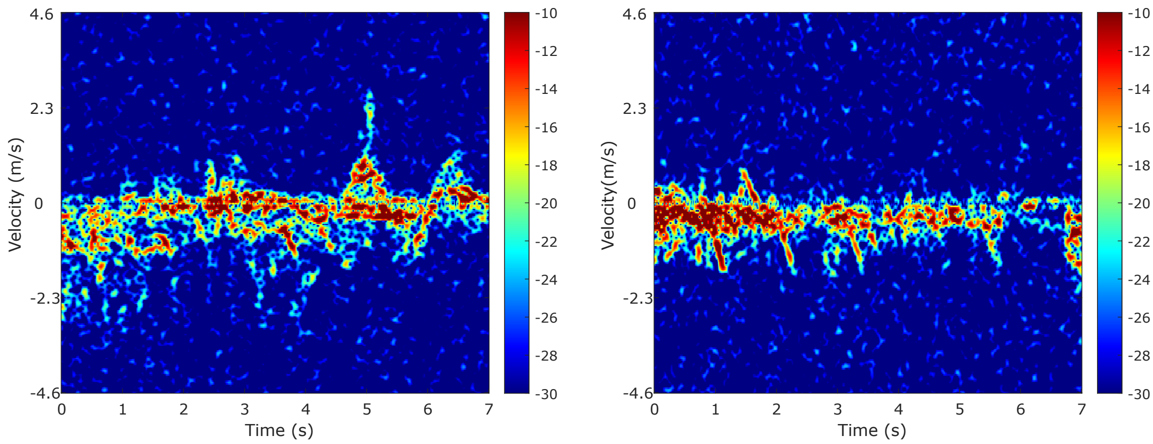
Fig. D.3. Spectrograms of Different Dog Movements



(a) Horse Walking Towards Radar

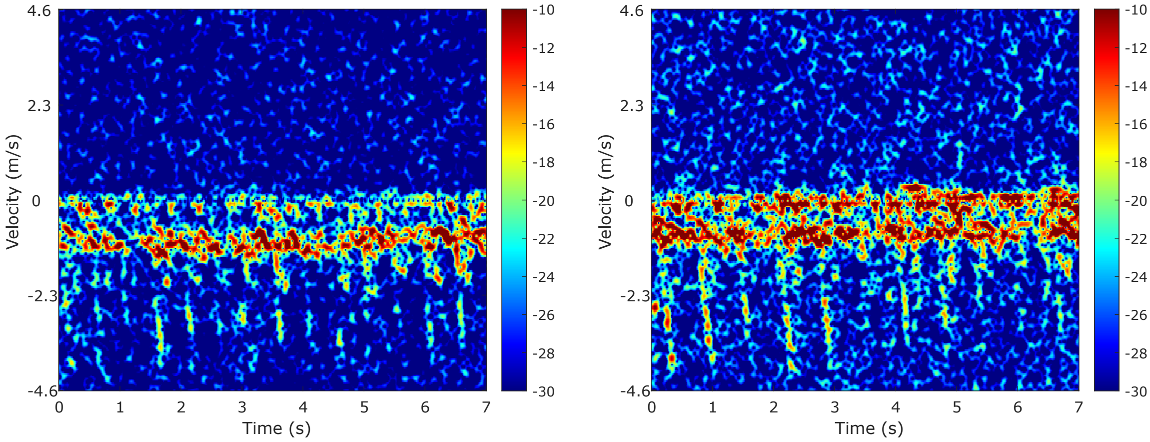


(b) Horse Walking Away From Radar

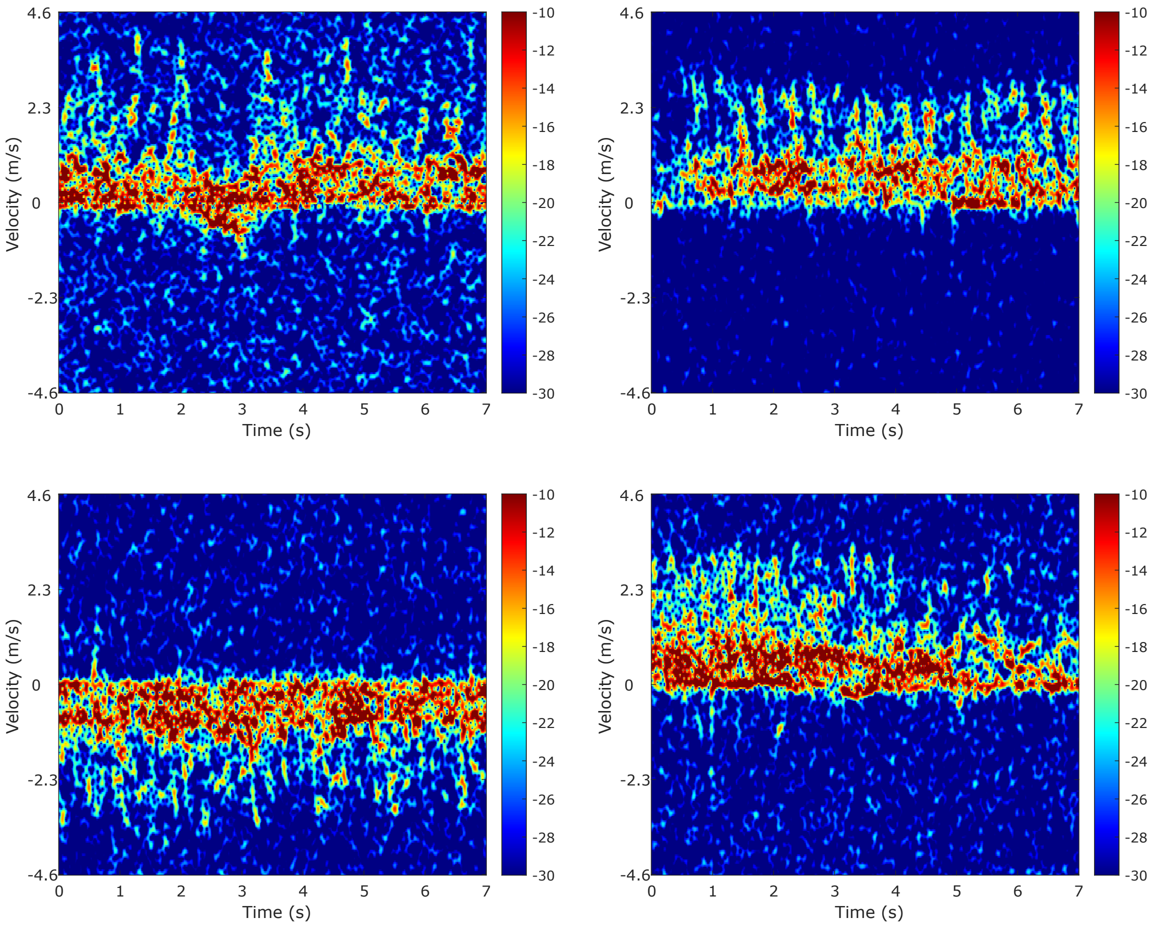


(c) Horses Grazing

Fig. D.4. Spectrograms of Different Horse Movements



(a) Single Cow Walking, Away From Radar



(b) Group of Cows Grazing and Walking

Fig. D.5. Spectrograms of Different Cow Movements