

UNIVERSITY OF BIRMINGHAM

DATA-DRIVEN CLASSIFICATION APPROACHES FOR RADAR BIRD-DRONE SURVEILLANCE

by

DANIEL WHITE

A thesis submitted to the University of Birmingham for the degree of

DOCTOR OF PHILOSOPHY

Microwave Integrated Systems Laboratory

School of Electrical, Electronic and Systems Engineering

College of Engineering and Physical Sciences

University of Birmingham

April 2024

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

© Copyright by DANIEL WHITE, 2024

All Rights Reserved

Abstract

This thesis addresses the topic of drone classification using radar spectrograms to provide warning of unauthorised drone intrusions. Radar is an excellent detection sensor, but the similar size, speed and flying height of drones and birds necessities the development of robust methods for target classification. Deep learning approaches may allow classification of drone targets in benign and challenging circumstances, but in a security critical scenario where lives may be at stake, several advances in their training and validation are required. This thesis seeks to accelerate the reliability of deep learning approaches, and to substantiate such developments a collection of 564 drone flights and 932 bird tracks were gathered over ranges up to 5km by two L-band staring radar systems in a high-clutter, densely urban environment at the heart of Birmingham, England. Raw-data spectrograms crafted from these flights were used to train and test a convolutional neural network for the classification of real targets in a challenging operational condition, where we show drones possessing rotor micro-Doppler can be classified to >98% accuracy, while drones without any micro-Doppler are still able to be classified ~90%.

As radar data is usually a scarce resource, this thesis also proposes approaches for synthetically increasing the quantities of available data, and additionally, a method to use such limited data more effectively in the training of classifiers. Initially, this work shows the production of highly-realistic synthetic micro-Doppler spectrograms that novelly uses motor speed recordings extracted from drone flight logs. This approach vastly increases the ease of creation and visual fidelity of simulated drone returns. The resampling of the recorded motor speed signal to fit the radar's temporal reference was shown to have a strong effect on the results achieved with synthetic training. Finally, unsupervised deep learning implemented with an elementary convolutional autoencoder trained with a modest amount of synthetic and real radar data, was demonstrated to have both performance and interpretability advantages over the use of an optically trained,

transfer learned classifier.

To Megan and our Families

Acknowledgements

The first and greatest thanks goes to those behind the Future Aviation Security Solutions (FASS) programme that funded and enabled this work. Special thanks to T. Hodges and A. Holt for their time and interest in all the project updates, and indeed all individuals involved within the Department for Transport (DfT), Home Office, and at Connected Places Catapult (CPC) that are unnamed who altogether provided enriching and meaningful experiences, as well as the many insightful conversations and presentations that ensured the work conducted was always grounded in the real world.

An equal portion of gratitude and appreciation goes to my supervisors. To M. Jahangir, I will be forever thankful for the guidance and inspirational example set in every day of the project. To M. Antoniou, I am thankful for all our catch-ups that were always centring. To C. J. Baker, I am thankful for your incisive questions, constructive thoughts and general curiosity that I hope to keep with me in any future endeavour. To J. Thiyagalingam, I am very grateful for you for always giving your effort and interest that was essential to over coming the challenges of the project. To Y. Singh, and those at or who have now left the Quantum Hub, it is thanks to you and my experiences at undergraduate that have allowed me to enjoy this project from the first day.

I would like also to extend this to all at UoB that have made my experience here one that I am not sick of after the 8 years since undergraduate. To my fellow students (D. Griffiths, X. Ren, J. Kannanthara, G. Donlan), those who have now left (G. Atkinson, H. Dale, J. Wayman), technical staff (A. Yates, A. Dunn) and all else at and around UoB that have made me happy and comfortable here.

For my family, I can't say anything but thank you. To Megan, it is hard to comprehend who, what, where, why or how I would be without you. *Don't regret forgetfulness; forget regret-fullness.*

Contents

Abstract i

Acknowledgements v

Contents vii

List of Figures xiii

List of Tables xix

List of Acronyms xxi

List of Terms xxv

1 Introduction 1

1.1 Project Background 1

1.1.1 Motivations for Radar-Based Drone Surveillance 2

1.1.2 Occupants of Low Altitude Airspace 3

1.1.3 Operation of Modern Drones 4

1.1.4 Overview of Sensors for Drone Detection 6

1.1.5 Overview of Radar Sensing 9

1.2 Objectives and Contributions 16

1.2.1 Problem Statement 16

1.2.2 Publications . . . 17
1.2.3 Contributions . . . 19
1.2.4 Thesis Outline . . . 20

2 Radar Essentials **21**
2.1 Introduction of Radar Capabilities . . . 22
2.1.1 Signal and Noise . . . 22
2.1.2 Measurement of Range . . . 22
2.1.3 The Radar Range Equation . . . 23
2.1.4 Introduction to Coherent Radar . . . 26
2.1.5 Angular Measurement . . . 27
2.1.6 The Doppler Effect and Radial Velocity Measurement . . . 28
2.2 Pulse-Doppler Processing and Phenomenology . . . 29
2.2.1 Down Converting Radar Signals to Baseband . . . 29
2.2.2 Fourier Analysis of Time-Series Signals . . . 33
2.2.3 Radar Signal Processing Choices for Drone Detection . . . 35
2.3 Radar Employed in this Project . . . 44
2.3.1 Staring Radar Principle . . . 44
2.3.2 Spectrogram Formation . . . 48
2.4 Chapter Summary . . . 51

3 Machine and Deep Learning Essentials **53**
3.1 Review of Machine Learning Techniques . . . 54
3.1.1 Evaluating Classifiers . . . 54
3.1.2 k-Nearest Neighbours . . . 57
3.1.3 Dimensionality Reduction . . . 58
3.1.4 Other common methods . . . 59
3.2 Review of Deep Learning Techniques . . . 61
3.2.1 Introduction to Neural Networks and Deep Learning . . . 61

3.2.2 Convolutional Neural Networks . . . 64
3.2.3 Other Common Neural Networks . . . 66
3.2.4 Optimising Neural Networks . . . 67
3.2.5 Unsupervised Learning with Autoencoder Neural Networks . . . 70
3.3 Review of Deep and Machine Learning in Radar . . . 71
3.3.1 Classification Using Tracker Output . . . 71
3.3.2 Handcrafted Feature Extraction of Spectrograms . . . 72
3.3.3 Automatic Feature Extraction in Radar classification . . . 73
3.3.4 Automatic Feature Extraction in Bird-Drone Radar classification . . . 74
3.4 Chapter Summary . . . 78

4 Experimental Data Collection 83
4.1 Control Drone Trials . . . 84
4.1.1 Motivation . . . 84
4.1.2 Drone Inventory . . . 85
4.1.3 Trial Procedure . . . 85
4.1.4 Drone Signatures Collected . . . 91
4.2 Control Bird Trials . . . 98
4.2.1 Trials Undertaken . . . 98
4.2.2 Control Bird Signatures Collected . . . 101
4.3 Opportune Bird Target Collection . . . 104
4.3.1 Motivation . . . 104
4.3.2 Collection and Filtering . . . 105
4.3.3 Opportune Signatures Collected . . . 107
4.3.4 Opportune Labelling . . . 111
4.4 Collected Data Summary . . . 112

5 Supervised Spectrogram Classification 115
5.1 Deep Learning Classification with Radar Bird-Drone Signatures . . . 116

5.1.1 Review of Drone and Bird Radar Classification . . . 116
5.1.2 Introduction to AlexNet . . . 125
5.2 Supervised Learning with CNNs . . . 127
5.2.1 Dataset Curation . . . 127
5.2.2 Classifier Training Approach . . . 128
5.2.3 Bench-marking Large Drones against Birds . . . 129
5.2.4 Bench-marking Small Drones against Birds . . . 131
5.2.5 Bench-marking All Drones against Birds . . . 132
5.3 Chapter Summary . . . 134

6 Creating Synthetic Drone Data 137
6.1 Simulation Approach . . . 138
6.1.1 Point Target Simulation . . . 138
6.1.2 Helicopter Rotor Micro-Doppler Simulation . . . 139
6.1.3 Multi-Rotor Drone Micro-Doppler Simulation . . . 143
6.2 From Simulation to Synthetic Data . . . 146
6.2.1 Injection into Realistic Background . . . 146
6.2.2 Phenomenological Sideband Attenuation . . . 147
6.3 Motor Speed Sampling for Micro-Doppler Simulation . . . 150
6.3.1 Introduction to the Effect of Motor Speeds . . . 150
6.3.2 Re-sampling Raw Recorded Motor Speeds . . . 152
6.4 Chapter Summary . . . 157

7 Classification with Synthetic Data 159
7.1 Creating a Synthetic Micro-Doppler Dataset . . . 160
7.1.1 Motivation . . . 160
7.1.2 Creation of Benchmark Synthetic Data . . . 161
7.2 Synthetic Training for Real Classification . . . 166
7.2.1 Synthetic Micro-Doppler Training Results . . . 166

7.2.2 Effect of Synthetic Data Fidelity . . . 168
7.2.3 Training with Real and Synthetic Data . . . 171
7.3 Chapter Summary . . . 175

8 Unsupervised Spectrogram Classification 179
8.1 Introduction to Autoencoders . . . 180
8.1.1 Architecture and Outputs . . . 180
8.1.2 Extensions and Radar Applications . . . 184
8.2 Autoencoders with Bird-Drone Radar Data . . . 190
8.2.1 Radar Classification with Unsupervised Pre-training . . . 190
8.2.2 Latent Representation of Bird and Drone Data . . . 200
8.2.3 Latent Representation vs Classification Performance . . . 216
8.3 Chapter Summary . . . 221

9 Conclusions and Future Work 223
9.1 Conclusions . . . 223
9.2 Future Work . . . 227

References 231

A Appendix 273
A.1 CNN Hyper-parameters . . . 273
A.2 ADB Latent Representations with 4-32 Latent Variables . . . 275
A.3 CAE Decoder Performance on ADB Dataset . . . 280

List of Figures

1.1 Flight Restriction Zone (FRZ) around airports in the 'DJI Fly' app. 5

2.1 The coherent I/Q detector. A carrier frequency with a small modulation due to range ($-\frac{4\pi}{\lambda}R_0$) is input and two orthogonal signals are output. From [65]. . . . 32

2.2 A post FFT (no window) real sinusoid. On left, 512 pulses were used and on right, 4096. 34

2.3 STFT of (drone sized) helicopter with 2 blades returns gathered by CW C-band radar. Ω is the drone's motor rate; $\frac{1}{\Omega}$ is the time period for a full rotation. From [87]. 37

2.4 The effect of number of rotor blades on a short-window STFT spectrogram. From [110]. 38

2.5 Comparison of long and short window FFT lengths in a low SNR scenario. From [87]. 40

2.6 RCS of a planar propeller blade at different view angles. From [111]. 40

2.7 Identified chopping and spool lines from a high SNR small helicopter Doppler axis with $N_b = 2$. From [87]. 41

2.8 The pictorial principle of staring radar. One wide transmit beam (red) digitally cut off at the edges (black), and many narrow receive beams 'created' in post processing (blue). 45

2.9 The Gamekeeper Staring Radars as installed on campus roofs. 46

2.10 Illustration of spectrogram construction from a series of datacubes organised per CPI. 48

2.11 A drone and bird as acquired by the UoB L-Band staring Radars. 50

3.1 A confusion matrix, from [131]. 56

3.2 A 2-dimensional graphical example of kNN. 57

3.3 Node schematic of a Neural Network. 62

3.4 The inputs and output of a generic perceptron. 63

3.5 Representations of overfitting in classifier training. 67

3.6 Illustration of overfitting in SVM. From [108]. 68

4.1 Drone Target Inventory 86

4.2 Plots summarising locations of drone data collected. 90

4.3 Spectrogram of the *Matrice* using Radar#1. 92

4.4 Spectrogram of the *Inspire* using Radar#1 at various heights with bird presence at low velocities. 94

4.5 Spectrograms of the small, μDoppler-less drones using Radar#1. 95

4.6 Spectrogram of the *Matrice* using Radar#2. 96

4.7 GPS Truth from Cranfield Trial. 99

4.8 GPS Truth from Cranfield Trial. 100

4.9 GPS data collected of all Pigeons with both radars' fields of view. 101

4.10 GPS and Spectrogram of Grey Buzzard Eagle that flew for an extended period in a spiral pattern at the Cranfield Trial. 102

4.11 GPS and Spectrogram of 'Pigeon-W' that flew close to radar transceiver. 103

4.12 Plots summarising locations of opportune bird data collected. 108

4.13 A selection of opportune bird spectra observed with Radar#1. 109

4.14 Gull flocking observed with Radar#1. 110

5.1 Schematic of AlexNet Layers. 126

5.2 False colour drone and bird samples as input to the classifiers of this chapter. Birds are in the top row and drones are in bottom row which are all from large drones, except for the bottom right which is of the *Mini* drone. 128
5.3 Results for Large Drones and Birds (LDB) dataset with $\mathcal{A}$CNN. 130
5.4 Results for Small Drones and Birds (SDB) dataset with $\mathcal{A}$CNN. 131
5.5 Results for Large Drones and Birds (LDB) dataset with $\mathcal{A}$CNN. 133

6.1 Top down view of a drone blade relative to radar position. 139
6.2 Doppler spectra acquired by, top: CMP ; middle: BTP; bottom: TWM, with a sampling rate of 7500 Hz for 2048 timesteps. Each model used a single rotor and $\Omega = 100\,Hz$. 142
6.4 Spectrograms produced with the three scattering choices with real motor speeds. 145
6.5 Effect of phenomenological sideband attenuation on a simulated spectra and spectrogram. 149
6.6 HERM Line spectrogram from equation 6.9 simulation model applied to a quadcopter of blade length, $L = 0.19$ m. 151
6.7 Per-CPI Sampling Approach and Result. 155
6.8 Sub-CPI Sampling Approach and Result. 156
6.9 The effect of rolling average window on Sub-CPI sideband spectral width varying the window from 1 to (1 - K), where $K = 4096$, the length of the FFT. 157
6.10 Flow diagram for the creation of a synthetic spectrogram as performed in this chapter. 157

7.1 Real and Synthetic *Matrice* spectrograms as used for Classifier Training, with (a-c) Radar#1 at 1.4 km, (d-f) Radar#1 at 1.4 km), (g-i) Radar#1 at 1.4 km; (a,d,g) Sub-CPI, (b,e,h) Per-CPI, (c,f,i) Real. 164
7.2 Summary of ADB synthetic training results. 167
7.3 Fifty-Six seconds of Real and Synthetic Spectrograms for the *Inspire* at 2.5 km range with Radar#1. 169

7.4 Summary of ADB synthetic training results with all synthetic data fidelities. . . 169

7.5 Synthetic Training comparison with half real and half synthetic simulated data with different numbers of best models assessed. 172

7.6 Side-by-side real (left) and synthetic Sub-CPI (right) spectrogram of a drone manoeuvre of the *Inspire*. 176

8.1 An example symmetrical CAE schematic. 180

8.2 Visualisations of datasets and latent representations in existing radar literature. . 188

8.3 CAE and $\mathcal{A}$CNN Classifier Schematics. 193

8.4 $\mathcal{A}$CNN compared with 5 depths of similar $\mathcal{R}$CNN. 196

8.5 Comparison of optically pre-trained $\mathcal{A}$CNN and 4 real and synthetically Radar (pre-trained) Convolutional Neural Network Classifier ($\mathcal{R}$CNN) classifier results. 199

8.6 Various Representations of LDB Training Dataset Decomposition. Birds (Blue), *Inspire* (Red) and *Matrice* (Black) and Radar#1 (Green) and Radar#2 (Orange). 201

8.7 *Simple*CAE architecture used to create the representations shown in this section. 203

8.8 CAE on LDB test set with 2 latent variables. *Matrice* is Black, *Inspire* is Red and Birds are Blue, while Radar#1 is green and Radar#2 is orange. 204

8.9 CAE on LDB test set with 2 latent variables. 206

8.10 Reconstructions of spectogram and central spectra with LDB dataset. Originals are left and black, with *Simple*CAE_2 reconstructions on right and in red. 208

8.11 LDB trained decoder uniformly sampled along each latent dimensional axis. . . 209

8.12 ADB labelled latent scatter plots of test set at epoch 30. *Mini* (pink), *Inspire* (red) *Matrice* (black) and birds (blue). 212

8.13 Comparison of *Simple*CAE_8+UMaP and UMaP with opportune bird test set at the 70^{th} epoch of training. 'Flock' (red), 'Single' (blue) and 'Unspecified' (pink) labels as described in Section 4.3.4. 215

8.14 Labelled Latent Variables for ADB after a number of epochs of training.*Mini* (pink), *Inspire* (red) *Matrice* (black) and birds (blue). 218

8.15 Mean kSS Evaluation of CAE pre-training plotted against supervised fine-tuning classification. 219

A.1 Latent variable visualisations of LDB using tSNE. Target labelled latent scatter plots on the left and radar origin on the right. 276

A.2 Reconstructions of Drones for 4 to 32 Latent Variables. 277

A.3 Reconstructions of Birds for 4 to 32 Latent Variables. 277

A.4 Latent Axis Decoder Sampling of 4, 8 and 16 Latent Variables. 278

A.5 Latent Axis Decoder Sampling of 32 Latent Variables. 279

A.6 ADB test set reconstructions against training epoch. 280

A.7 Decoder sampling of each latent axis against training epoch. 281

List of Tables

1.1 Brief comparison of UAV surveillance techniques 7

1.2 IEEE Standard Frequency Bands used in Electronic and Radar Engineering . . 10

1.3 Some CUAS Commercial Radar Manufacturers and Systems 15

2.1 List of specifications for the L-band staring radar 47

3.1 Selection of recent radar literature utilising deep learning. Att., using attention mechanism; XAI, explainable artificial intelligence; BNN, Bayesian Neural Network; CRNN, Convolutional-Recurrent Neural Network 78

4.1 Drones used in data collection trials. 85

4.2 Overall quantities of collected data 113

5.1 Radar and RNN classification approach reported in [94] 117

5.2 Radar and CNN classification experiment reported in [124] 119

5.3 Radar and CNN classification experiment reported in [250] 121

5.4 Radar and NN classification approach reported in [127] 122

5.5 Selection of recent radar literature utilising deep learning for classification using real bird and drone targets. *-Indicates work originating from this thesis. TL, Optically Transfer Learned. 124

5.6 Composition of Large Drones and Birds (LDB) data subsets 130

5.7 Composition of Small Drones and Birds (SDB) data subsets 131

5.8 Composition of All Drones and Birds (ADB) data subsets 132

5.9 Results for the three datasets using $\mathcal{A}$CNN in Section 5.2 134

7.1 Composition of All Drones and Birds (ADB) data subsets. *Simulated Data . . 161

7.2 Simulation Parameters Used in Benchmark Synthetic Dataset Creation ($2E^{-3} \equiv 2 \times 10^{-3}$) 165

7.3 Summary of ADB synthetic μDoppler training results for all synthetic data fidelities. 171

7.4 Composition of All Drones and Birds (ADB) data subsets. *Synthetic Data . . 172

7.5 Summary of ADB half real and synthetic training results with all synthetic data fidelities. 173

7.6 Results of synthetic training in Section 7.2.. 175

8.1 Quantities of data used in the LDB (left) and ADB (right) for training of $\mathcal{A}$CNN and $\mathcal{R}$CNN. 191

8.2 Configurations of the five convolutional depths of CAEs trained 194

8.3 Best accuracies of the optically and radar data pre-trained models. Subscript denotes number of convolutional layers. 195

8.4 Best accuracies of the optically, real and synthetically radar data pre-trained models 198

8.5 Table of results with LDB dataset comparing unsupervised kSS results 211

9.1 Summary of key results presented in the thesis. 228

A.1 Typical CNN Hyper-Parameters. 274

A.2 Typical Training Hyper-Parameters. 274

A.3 Spectrogram DSP options. 274

List of Acronyms

$\mathcal{A}$CNN AlexNet (pre-trained) Convolutional Neural Network Classifier. xv, xvi, xx, 127, 129–135, 160, 166, 168, 175, 176, 190–199, 202, 213, 216, 221, 222, 226

$\mathcal{R}$CNN Radar (pre-trained) Convolutional Neural Network Classifier. xvi, xx, 190, 191, 193, 195–199, 211, 213, 216, 219–221, 226

μDoppler Micro-Doppler. 36, 85, 137, 160

σ_{RCS} Radar Cross Section. 10, 24, 57, 72

***Alexnet-like* CAE** AlexNet-like Convolutional Autoencoder. 192–195, 221, 228

***Simple*CAE** Simple Convolutional Autoencoder. xvi, 201–203, 205, 207–211, 213–216, 218, 222, 275, 276, 280

AC Alternating Current. 31

AdaM Adaptive Moment Estimation. 64, 121, 129, 131, 195, 274

ADB All Drones and Birds. xvi, xx, 129, 134, 161, 166, 180, 191, 195, 196, 198, 199, 212, 213, 216, 221, 222, 224, 226

AE Autoencoder. 200

AMSL Above Mean Sea Level. 90, 108

ATR Automatic Target Recognition. 2

BCE Binary Cross Entropy. 274

CAE Convolutional Autoencoder. xvi, 61, 70, 73, 97, 179–181, 226

CNN Convolutional Neural Network. 61, 65, 66, 73, 79, 115, 116, 137, 159, 160, 190, 224, 228

CPI Coherent Processing Interval. 34, 88, 144, 152

CVD Cadence Velocity Diagram. 75

DC Direct Current. 30

DSP Digital Signal Processing. xx, 88, 134, 185, 274

DT Decision Tree. 59

EM Electromagnetic. 7, 22, 28

FEA Finite Element Analysis. 75

FFT Fast Fourier Transform. 29, 33, 146, 147, 152

FMCW Frequency Modulated Continuous Wave. 11, 27, 185

FRZ Flight Restriction Zone. xiii, 5

GAN Generative Adversarial Network. 185

HERM HElicopter Rotor Modulation. 37, 129, 141, 142

ICBP International Centre for Birds of Prey. 99

IP Intellectual Property. 12, 45

kNN k-Nearest Neighbours. 57

kSS k-Nearest Nearest Neighbours Self-Similarity. xvii, xx, 210, 211, 216, 217, 219

LDA Linear Discriminant Analysis. 58

LDB Large Drones and Birds. xvi, xvii, xx, 129, 133, 134, 166, 191, 195–201, 203, 208, 209, 211, 213, 221, 222, 224, 226, 276, 280

Lidar Light Detection and Ranging. 7

LPF Low Pass Filter. 30

LSTM Long-Short Term Memory. 66, 122

MRμD Multi-Rotor Micro-Doppler. 152, 160, 176

MSE Mean Squared Error. 274

NN Neural Network. 61

PCA Principle Component Analysis. 58, 97, 179, 181, 200

Per-CPI per-Coherent Processing Interval. 153, 162, 166

PRF Pulse Repetition Frequency. 11, 33, 154

RADAR Radio Detection and Ranging. 21

ReLU Rectified Linear Unit. 63

RF Radio Frequency. 5

RNN Recurrent Neural Networks. xix, 66, 117

RPM Rotations Per Minute. 87

RRE Radar Range Equation. 23, 29, 138, 163

SAR Synthetic Aperture Radar. 73

SDB Small Drones and Birds. 129, 134, 224

SDK Software Development Kit. 87, 227

SGDm Stochastic Gradient Decent with Momentum. 64, 128, 274

SNR Signal-to-Noise Ratio. 23, 77, 88, 120, 146

STFT Short-Time Fourier Transform. 36, 87, 91, 103

Sub-CPI sub-Coherent Processing Interval. xvi, 154, 162, 166, 175, 176, 226

SVM Support Vector Machine. xiv, 57, 59, 68, 75, 186

SWAP Size, Weight and Power. 27, 35

tSNE t-Distributed Stochastic Neighbour Embedding. xvii, 58, 59, 65, 97, 179, 181, 200, 216, 276

UMAP Uniform Manifold Approximation and Projection. 58, 65, 97, 179, 181, 200

VAE Variational Autoencoder. 184

List of Terms

accuracy Ratio of total correct predictions to total predictions made. 55, 130

AlexNet A 8 layer deep CNN model that was trained on the large ImageNet used as a powerful feature baseline. 70, 120, 125, 160, 228

autoencoder A deep learning compression algorithm that attempts to recreate the input at the output. 8, 66, 77

balance Describing ratio of accuracies of different classes across all classes. 56, 132

bandwidth The width of frequency space that a radar operates within. 11, 25

confusion matrix A view of classifier performance on an evaluation set that is a square matrix that displays the number or ratio correct classifications along the diagonal, and simultaneously the information of mis-classifications on the off-diagonal. 56

contrastive learning Autoencoder based learning where the classifier input and loss-defining output are not the same, forcing the autoencoder to learn a representation of the image that is robust to a specific format of data input. 185

decoder A neural network structure that decompresses a latent input to produce an output that is representative of the encoded distribution. 181, 182, 200

denoising autoencoders Autoencoder based learning where the input is corrupted by noise but the loss-defining output is unaffected by the noise, forcing the autoencoder to learn a representation of the image that is robust to noise. 77, 184

Dirac delta ($\delta(\underline{z})$) function A mathematical structure that has value 0 everywhere except the location of the point, with integral over all space yielding unity. 33

encoder A neural network structure that compresses or encodes the data set to a latent representation in the form of variables or a distribution. 181, 182, 200

fine-tuning The act of training in a supervised fashion for classification a neural network that was already 'pre-trained' in a different way or on different data, that already possessed feature extracting qualities in its convolutional layers. 77, 125, 185, 190, 211, 216, 228

ImageNet An 8 layer deep CNN model that was trained on the large ImageNet. xxv, 126

Kronecker delta (δ_{ij}) For a discrete space that has multiple dimensions, the value of the Kronecker delta $\delta_{ij} = 0$ if $i \neq j$ but $\delta_{kl} = 1$ if $k = l$. 55

Mie regime Mie regieme scattering describes near-resonant electromagnetic. 10

multistatic Describing a system of radars where there are one or more receivers measuring signal from one or more transmitters. 3

optical region Where the wavelength is significantly smaller than the physical scattering object. 10

passive Referring to a radar configuration where there is a receiver only, and the measured signals reflecting off targets comes from opportune sources that were not placed by the operator, for example, 5G mobile signals. 6, 13, 184

perceptron The basic neuron in an artificial neural network, that takes many inputs, multiplies each input by a weight, adds its bias and then applies its activation (non-linearity) function. xiv, 61, 63, 185

Rayleigh region EM scattering regeime where wavelength is significantly greater than the physical reflector. 10

recall For a given semantic class, the recall is the ratio of correct classifications to the sum correct classifications and false alarms. 56, 130, 197

semantic segmentation The division of a scene in recognisable components that possess a pre-decided semantic characteristic. 8

spectrogram A time-frequency graph of signal that shows the frequency composition of a target developing over time. 36

spectrum allocation The electro-magnetic frequency bandwidth that a system must operate within to avoid measuring signals from other systems, and visa versa, to avoid polluting the spectrum that other potentially critical devices may be operating in. 7, 14

staring radar A pulsed radar system with a single flood transmit beam and 3-dimensional, software defined, post-receive digital beam forming. Also may referred to as ubiquitous or holographic radar. 44, 224

supervised learning Machine learning where labels are assigned to data instances, and the algorithm uses the labels as a metric to optimise its parameters towards a certain goal. 16

transect Observating of a sampling line to manually label radar tracks. 104

unsupervised learning Machine learning where learning takes place without prior data labelling. The algorithm learns from the data towards a specific goal, but the semantics of

the data are unknown to the algorithm. 17, 79, 179

Chapter 1

Introduction

This introductory chapter will review the motivations for researching radar-based drone surveillance techniques in Section 1.1, while Section 1.2 will summarise the technical findings and publications resulting from this project.

1.1 Project Background

The market boom of consumer unmanned aerial drones presents a difficult and urgent challenge to those looking to protect sensitive air-spaces from tresspassing. A new wave of research in stand-off automatic drone detection has thus been underway for the last decade or so, and academic and industrial efforts towards solving the drone surveillance problem is required to grow as the drone industry does. Drones have and will continue to become cheaper and more widely available as their technological components: batteries, motors, computational units, cameras and other sensors; improve independently and are mass produced at scale. The other side of the race is increasing regulations and enforcement of drone laws worldwide [12], which is not necessarily any faster than the development of drone risk. There is much growth to look forward to on the civilian and industrial fronts, but at the same time the growing ease of access to drones raises anxieties of the increased use of illegitimate drones. Organisations must be

prepared and ready to detect and neutralise rogue drones, regardless of the level of regulations present. In the defence sector, onlookers of the presently ongoing Russo-Ukrainian war have seen the reality that drones are used whereever they can to improve the tactical stance of a side. Publicly available drones have already transformed how offence and defence take place [13], with drones equipped with live feed cameras being routinely deployed for rapid, active surveillance of strategic areas. The future trend of these observations collectively paint a picture within which drone surveillance will become, if not already, amongst the most critical civilian and defence infrastructure.

1.1.1 Motivations for Radar-Based Drone Surveillance

There are numerous potential approaches to detecting airborne drones over an area of interest (positives and negatives of each are discussed in Section 1.1.4), and in practice not one single system alone will fully satisfy the end user completely as each of the methods have their downsides [14]. Interrogating the airspace with a distributed approach using multiple different systems must be the solution to keep the chances of a drone reaching sensitive areas as low as reasonably practicable [15]. A human is an effective drone detector at short ranges, but an automated system is required for 24 hour, uncompromisable surveillance. In the particular problem of drone detection, *classification* or Automatic Target Recognition (ATR) is the biggest hurdle. The reason for this hurdle is the existence of birds, that potentially exist everywhere and always in an airspace. Birds and drones have very many similar properties: size, colour, speed and manoeuvrability. At close range, classifying a target is a simpler task as features unique to a bird or a drone can be used, such as the observation of the spinning propellers and the associated noise, versus the sounds of birds and observing the wing flaps with the effect on the trajectory. At long range, using these features is less reliable and a much longer dwell time of target interrogation is required before a confident classification attempt, so that the rate of false alarms is kept as low as possible. Indeed, drones flying amongst birds may mask the drone signal enough to forfeit its detection, thus detection methods need to be robust to multi-target configurations for initial detection and also have precise classification tools to label complex

detected targets as a threat or not. The choice of sensor will dictate the fidelity of returns and classification of targets is highly dependant on this.

This thesis will comment on some of the literature achievements of drone vs bird detection in radar, and then describe developed improvements. Radar however is possibly, more appropriate as an umbrella term for lower-frequency, electromagnetic scattering and reflection measurement sensors, meaning there is a huge variation in radar systems that may be used for drone detection. The capabilities of a radar system will scale with cost, where millions of pounds could be spent on a single system. Smaller, shorter ranged radars have a lower cost and may be more beneficial to some applications. Continuous advances in networking separate radar systems together, which use the same transmitted radiation to feed multiple receivers known as multistatics, promise wider coverage and a more cost effective solution moving forward. Therefore, choosing 'radar' as a solution to this challenge is itself a question as there are many different types of radar that could be used. Sec. 2.2.3 will be dedicated to the assessment of the types of radar that exist and which are used for drone detection. It will also be discussed in more depth in Chapters 3, 5 and 8 that a host of classification techniques are available for researchers to optimising their systems detection performance.

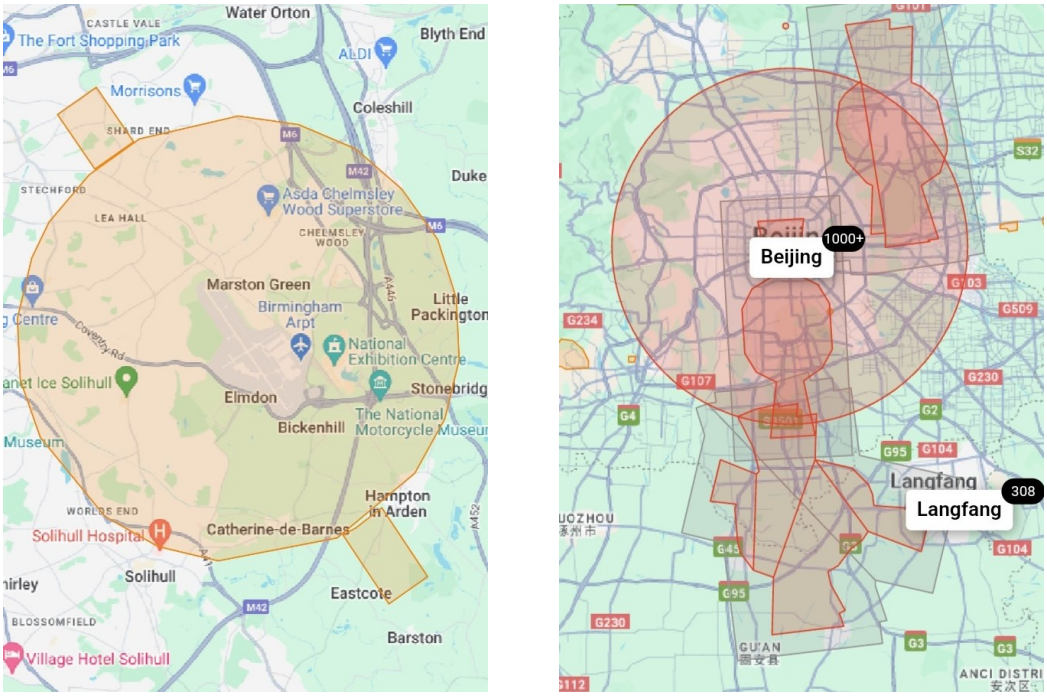
1.1.2 Occupants of Low Altitude Airspace

When surveilling for drones, birds will be detected as their airspace occupation overlaps greatly with drones. As mentioned, detected targets must be classified to whether they are a potential drone threat, or a benign bird sighting. Birds can have any flight mode [16], there are many species of bird, and any number of birds may occupy the surveilled airspace which the drone detection sensor must be able to see through and robustly classify to determine if a threat is present [17], [18]. In this work, a wide variety of bird signatures have been collected and these will be shared and analysed in Sections 4.2 and 4.3. Collected signatures, including large flocks and mixtures of species, will be compared to the returns of drones, which is essential to understanding the fundamental challenges of radar based drone detection. Drones likewise can

possess a large amount of intra-class variation which must be understood by a radar classifier [19], [20]. The key difference amongst drone targets from the radar perspective is whether or not the motion of the spinning rotor blades can be detected by the radar. Analysis and simulation of this phenomena is the greatest original contribution of this work (see Chapter 6 and [3]); however, as the drone target may or may not have visibility of the rotor blades - either from small drone size, radar used or intentional obscuration - a radar target classifier needs to be able to classify drones based of nuanced, subtle features within a wide range of possible bird returns. For this reason in this work we assess the feasibility of classifying small and large drones, separately and together in Sec. 5.2.3 to 5.2.5.

1.1.3 Operation of Modern Drones

Drones are a modern technology and there will be many improvements of their capabilities over time. This section shares prerequisite information of drones that must be considered by surveillance staff towards ensuring their airspace is defended from all types of drone. There are many vendors for both first and second hand consumer drones, the most popular manufacturer being DJI (dji.com/uk). Most of production DJI drones are marketed towards photography and videography applications, with more advanced variants employed such as users equipping them with thermal cameras for mountain and sea rescue missions. Other, typically larger models are used as flying platforms for other technologies [21], including recent research from the University of Birmingham, such as air-borne radar imaging applications [22] or hosting quantum optical cold-atoms based device [23]. This class of drones will be used more as entrepreneurs and public service officials look to apply drones for rapid delivery of agriculture [24], goods and emergency supplies. A typical drone may weight between a few hundred grams and up to 10kg, the upper end of which being less common and more highly specialised. Above 25 kg, the UK's Civil Aviation Authority requires a much greater increase in certification and permission to operate.



(a) Birmingham International Airport (b) Greater Beijing Area

Figure 1.1: Flight Restriction Zone (FRZ) around airports in the 'DJI Fly' app.

The availability of drones is wide and there is a risk of their use in domestic terrorism or smuggling. However, it is not trivial to use these drones in such a manner due to software restricting access in no-fly zones. Figure 1.1 illustrates the areas where flying is permitted and not, and using the controller alone it will not be possible to break these *geofencing* restrictions imposed from the countries' governing body. For detection of DJI drones in air-spaces not on the official no-fly list, it is possible to purchase the DJI *AeroScope*[1] that reads the transmitted signals between controller and drone, and feeds back the location of the drone, operator, and other information attainable from the signal. Although the AeroScope is no longer in active production at DJI, open source solutions [25] and other professional options[2] exist that use a large database of measured signatures to extract information from the drone. Extracting information from non-cooperative drones is an area of active research [26].

The greatest downside of passive drone-controller Radio Frequency (RF) detection and decryption, is that it cannot guarantee that it will be able to detect certain drones for the following

[1] https://www.dji.com/uk/aeroscope

[2] such as DeTect's `DroneWatcherRF` https://detect-inc.com/ or CRFS https://www.crfs.com/drone-detection

potential reasons. It is possible to build your own remote control drone for as little as £100, and so signals from a totally unseen custom model may not be in the database of known threats. Consumer drones still pose a threat as they can be 'hacked' to allow operation outside the manufacturers limitations as an open-source interest, the knowledge of which is growing due to their use in modern conflicts. With this, the geofencing restrictions can be removed, and it would be possible to have the drone fly along a predetermined flight path without any remote connection to the operator. In this scenario, passive RF identification will not be an effective counter-measure, and an active radar or another surveillance device would be the only warning of the intrusion of such as target. To overcome the threat of a drone thus requires a method of actively detecting a physical object approaching a sensitive area with classification of drone or not. Similarly but separate to passive radio-frequency detectors just discussed, passive radar techniques show capabilities to detect drones without a transmitted signal using common illuminators of opportunity, such as mobile or television signals, and will be included in the discussion in Sec. 2.2.3.

1.1.4 Overview of Sensors for Drone Detection

Before reviewing the known radar techniques that have been used for drone detection and discrimination, radar will be initially discussed in context with other counter drone sensors that have been implemented in literature and used industrially. Operators will need to understand their environment and then consider if using a multitude of solutions in parallel is required to increase their security above the required threshold. For the majority of cases, the passive RF detection method from the last section would detect the majority of drones - but would totally fail if the connection between the controller and operator was intentionally cut. Table 1.1 shows a list summarising the strengths and weaknesses of common counter drone sensors which will be elaborated in the remaining text, with reviews existing in [27], [28].

Beginning with radar, the main downside is that its cost scales with effectiveness. Radars

Table 1.1: Brief comparison of UAV surveillance techniques

Method	Strengths	Weaknesses
Radar	Long range, all weather, 24hr operation	Line of sight requirements, cost, power, spectrum allocation
Lidar	Very effective at target discrimination	Poor at medium range & in certain weather conditions
Video	Excellent short range & multiple sensor distribution	Poor classification at long range & low light
Audio	Classification & accurate target locating	Short range, forgeable returns, weather and background noise
Human	Accurate & verifiable observations	Costly, opportune and compromisable

provide a long-range early warning detection where there is a direct line of sight. The effective range of reliable classification will decrease rapidly with cost to a point where multiple radar sensors operating in unison would become a more appealing solution [29], [30]. The two key, unmatched strengths of radar is that it is the most weather resistant sensor (operation and classification is minimally affected by fog, rain and snow), and operates in all lighting conditions as it provides its own illumination. No other of the sensing solutions discussed in the following can provide both of these advantages, and thus radar could be considered the priority system to install for a surveillance application, with other sensors making up for any of the deficiencies of their radar installation in any. An issue unique to radar that can prevent a possible deployment is spectrum allocation. Radio frequency based communication technologies require an allocated spectrum to operate within, which is a problem in developed countries, in particular the United States of America where spectrum is allocated to the highest bidder. Classification in radar of similarly sized, difficult to distinguish targets has become a highly active area of research in recent decades as its excellent detection and tracking qualities manifest the desire to also produce reliable feature extraction and subsequent target classification.

Radar and Light Detection and Ranging (Lidar) are very close cousins as both measure range by timing the round trip of returning EM signals. In Lidar, the typical frequency range of operation is in the visible or near visible part of the spectrum. This much higher frequency makes it difficult to extract phase information from the return light, unlike what is commonplace

in radar (see Section 2.2), however continuous advancements mean now Doppler lidar is possible but expensive. Otherwise, lidar is capable of measuring velocity by comparing the range of successive measurements. Typically, lidar sensors are very expensive beyond the simple range-finders and laser measures. Due to the high cost of quality lidar sensors, and given their frequency use in more civil applications such as landscape imaging or conservation efforts (for example, Google Earth or other forest monitoring efforts), raw lidar data exists in a standardised .las format so open collaboration may occur on expensive and hard to acquire datasets [31]. Range is measured extremely precisely in lidar, and cross range measurements are done by integrating multiple light pulses from different directions. Lidar thus is a very expensive option for drone detection, but as it is a high resolution sensor, clear line of sight to the target at modest range will allow a confident classification as it will detect many detailed features of targets [32]. The biggest downside are its limited range, its price, and that its operation will diminish rapidly in poor weather, particularly in fog, rain and snow.

Optical cameras are another very mature technological choice for drone surveillance. The greatest advantage of optical image or video based surveillance and classification is that the majority of machine learning research exists for optical images, and so optical classifiers can more conveniently draw upon the power of deep learning for this task. There exists many examples of drone and bird differentiation with optical images and videos. Modern image classification is now accompanied with semantic segmentation which is the learned ability to draw a digital enclosing box around objects that pass the threshold of categorical classification. This precise technique is based on an autoencoder framework (introduced in this work for radar in Chapter 8), and requires a huge amount of data for training to be effective thus restricting occurrences of this method mainly to the optical domain with many open source implementations [33], [34]. Similar demonstrations of this approach within radar have been performed [35], [36]. This segmentation method is essential for camera operation as it permits a method to track the target digitally through the surveillance space. Radar and lidar have azimuthal and elevation measurement through beamforming and point clouds respectively, which would not be expected to be as precise or reliable as camera image segmentation [37]. Target range measurement on

the other hand would be crude with cameras when compared to radar and lidar, where ranging is already implemented to well defined resolution. The networking of multiple surveillance cameras is a cost effective solution for wider coverage drone surveillance, but at longer range and in low light or poor weather conditions its reliability suffers greatly.

A notable difference between drones and birds is their emitted sounds. Birds are either silent or make their calls, while drones come with a loud 'buzzing' sound from the air displacement of the rapidly rotating propellers. Thus, sound based classification has been explored for drone detection and identification using deep learning automatic target classification methods [38]. This method is largely unreliable as the effective range is short, and quality of data will be broken with strong winds, rain and other ambient uncontrollable noises. Further technological developments of drones to reduce their ambient noise, such as toroidal propellers, will make audio based detection even harder.

Lastly, using human security guards as a means for drone detection often has been one of the only lines of defence aside security cameras in previous drone incursions on civilian infrastructure (Gatwick, 2018 [39]). They cannot be relied on alone for drone detection as they have human limitations of eyesight and opportune positioning, as well as the cost and practicalities of human availability on a 24/7 basis.

1.1.5 Overview of Radar Sensing

In this section, a brief review of existing radar sensor systems is followed by a brief summary of radar target classification techniques to preface the relevance of this works contributions. Later, Chapter 2 will in greater detail review the relevant radar principles used in this work, within which Sec. 2.2.3 will contain a focused review of the phenomenology the radar radiation and drone interaction. When constructing a radar system, there are many considerations. The first consideration could be the choice of operating frequency. Table 1.2 shows the separation of the spectrum into bands alongside their frequency, wavelength and typical civil and radar applications.

Table 1.2: IEEE Standard Frequency Bands used in Electronic and Radar Engineering

Band	**Frequency**	**Wavelength**	**Civil Application**	**Radar Application**
HF	3-30 MHz	10-100 m	Radio Broadcasting	Over-the-Horizon radar
VHF	30-300 MHz	1-10 m	Radio & TV Broadcasting	Very Long Range Surveillance
UHF	0.3-1 GHz	0.3-1 m	$5G^{Low}$, Ship and Aircraft Communication	Long Range Surveillance Radar
L-band	1-2 GHz	15-30 cm	EPR (Medical) Imaging [40]	Weather-proof surveillance radar
S-band	2-4 GHz	7.5-15 cm	WiFi, Bluetooth, LTE [41]	NexRad Weather Radar [42]
C-band	4-8 GHz	3.75-7.5 cm	Satellite Communications	Wind Measurement [43] & Early Warning Radar [44]
X-band	8-12 GHz	2.5-3.75 cm	Terrestrial & Space Broadband	Longer Rage Imaging Radar
Ku-band	12-18 GHz	1.7 - 2.5 cm	Down-link Satellite Communications	Various Short-Medium Range
K-band	18-27 GHz	11 - 17 mm	Short-Range Specialist Applications	Water Vapour Absorption Detection
Ka-band	27-40 GHz	7.5-11 mm	$5G^{High}$, High-bandwidth Space Communication	Police Vehicle Speed Measurement
V-band	40-75 GHz	4-7.5 mm	GB/s Rate Wireless Communications	Compact, High-bandwidth Applications
W-band	75-110 GHz	2.3 - 4 mm	Concealed Weapon Detection	Beyond-Atmospheric Measurements
mm wave/ Sub-THz	0.1-0.3 THz	1-3 mm	Experimental, High-Speed Wireless Comms [45]	Experimental Space Radar Detection and Imaging [46]

The maturity of each radar frequency band is not equal, with particularly Sub-THz being a modern experiment trend. To understand and have a valid intuition of how the transmitted radiation will interact with the target, it is useful to compare the radiation wavelength to the physical features of expected targets. This is because each photon of coherent EM radiation oscillates with the same phase, meaning that if the dimensions of a feature are similar to the wavelength, currents will be induced on the target that will be re-enforced with further incoming radiation leading to resonance of the currents that will provide a very strong return, which is known as Mie regime scattering. The scattering of EM radiation with targets where the wavelength is less than the physical dimensions is in the Rayleigh region, and conversely where the spatial frequency of the target is less than the frequency of the radiation is called optical region scattering. As will be formally introduced in 2.1.2, the Radar Cross Section (σ_{RCS}) is the mathematical representation of the strength of returns / the radar reflectiveness of a target. Thus the choice of operational wavelength is important, and depends on the application. For instance, sub-THz radar with wavelengths in the order of a millimetre makes it feasible for through-clothes imaging in body scanning security applications at low ranges. The radar system used in this work that we will expand upon in Section 2.3 that was designed for drone detection has an operational wavelength of $\sim 24\ cm$ which is very close to the blade length of a large rotary winged drone, thus returns

from the blades themselves are very strong.

With operational frequency decided, then follows the operating mode of the radar. For range measurement capable radars, three possible choices are pulse-Doppler, Frequency Modulated Continuous Wave (FMCW) radar or passive radar. The experimental results and validation of all research of this thesis took place with a industrial, airport-grade L-band pulse-Doppler radar described in Section 2.3. While the experimental results in this thesis are formed with data from the L-band system, the efforts of this PhD were to find advances of drone detection for all radar types. In this introductory section the key differences between pulsed and FMCW radar systems are described. As pulse-Doppler is the grounds for development in this work, a thorough background of the mechanisms and signal processing of pulse-Doppler radar is presented later in Sec. 2.2. In the following paragraph, the basic differences of the two techniques are overviewed.

Typically, FMCW radars are cheaper, more compact and better suited to shorter range detection. Pulse-Doppler radars are better suited to very long ranged operation due to the ease of integrating multiple returns to build the strength of genuine signals above the noise floor, provided the Pulse Repetition Frequency (PRF) is sufficiently high. As FMCW are transmitting all the time, a relatively lower power is used at a constant rate. Pulsed radars on the other hand only transmit for a fraction of its duty cycle, thus the peak power needs to be much greater which is why they are typically more bulky for a given power budget. The maximum unambiguous range (see 2.1.2) of a pulsed radar system is determined by the listening duration after a pulse is transmitted, whilst in a FMCW system it is determined by the chirp duration. Due to the high PRFs used in Doppler-prioritising pulsed radar systems, FMCW systems have a greater maximum unambiguous range - however, due to their lower peak power, rarely would you expect to detect signals near the edge of the maximum unambiguous range, whereas in a pulsed system these two values (maximum unambiguous range and peak power with respect to desired minimum target size) will be set in harmony with one another. The range resolution of a radar system is determined by its bandwidth, B. For FMCW radar, this is simply the difference of the minimum and maximum values of the chirp. For pulsed radar, this is the reciprocal of the pulse width. In either case, the

range resolution, ΔR, is given by $\Delta R = \frac{1}{2}cB^{-1}$, where c is the speed of light. It is common to see FMCW range resolutions much smaller than that of a pulsed system, which may be in the order of centimetres. Pulsed radars may have very small range resolutions if their application is a very short ranged one, dictated by a shorter pulse with anticipated accompanying smaller power. The range resolution of the radar will impact the quality of tracking delivered, which in turn will support a superior trajectory based classification. The other key factor to enabling moving target classification in radar is velocity measurement. The velocity spectrum of a target or scene will provide the most instantaneous features that may be leveraged for classification, so is an important consideration when applying this technology to drone detection. Both FMCW and pulse-Doppler radar can increase their Doppler resolution through longer continuous integration. Ultimately, the choice of operating mode *and* operational frequency are important decisions to be made by the radar engineer designing a solution to meet the operational desires and budget.

Both pulsed and FMCW radars allow for target tracking. This thesis did not attempt track based classification as a research topic. Alternatively, analysis of the spectra of targets was performed as the main focus[3]. A radar will require both tracker and spectral based classification procedures, and operators need to ensure these are well optimised for their environmental setting. Many works and research groups have focused on using the flight profiles of drones and birds to support classification, such as,

- Various works from the Thales Group [47]–[49],
- From University College London on multi-static radar tracking [50], [51],
- From the University of Cambridge regarding multi-target tracking [52], [53].

The combination of tracker based classification and feature extraction alongside information gained from the velocity spectrum of a target will provide the most robust feature set for accurate

[3]Research in tracking algorithms is an extremely important area of study, particularly for small manoeuvrable targets such as drones and birds. Two key reasons for not investigating tracking in this work were, (A) the tracking algorithm installed onto the COTS staring radar was protected by their IP, (B) to process staring radar data into detections, then tracks, then subject to further analysis would require a greatly increased computational burden, that at the project outset was decided not to be the focus of the thesis given time constraints.

and reliable classification of unknown, detected targets. A major challenge in this area of track-based classification is appropriately dealing with multi-target scenarios. For instance, if a bird was to closely follow a drone, the track quality and subsequent classification efforts would be potentially compromised.

Spectrum based classification of coherent radar returns is an area of ongoing active research. Modern radars produce so much information from the environment and targets beyond simple detection and tracking, that extracting as much semantic information from the already collected data is a priority. Of course, extracting reliable information from radar spectral data has not historically been easy due to the unpredictability of return nature, target configuration and other considerations such as unintentional radar artefacts, or intentional jamming from foes. The combination of possessing high-quality, detailed measurements from a radar system alongside the modern trends of machine learning enabling automatic target recognition that is protected from minor anomalies means now radars can be highly optimised to have excellent operational standards for a particular application. The greatest challenge in this domain, is the availability of large amounts of labelled data that is required to train and validate performance improvements over existing methods. Efforts are needed in both, the ability to create large labelled datasets and also to intelligently process large amounts of unlabelled data - both of which will be addressed in this work.

Another domain of radar not mentioned thus far are passive radar systems. This receive-only method allows detections, tracking and classification to be performed without transmitting any radiation. The source of EM energy from which to extract situational awareness are here other radiation sources, referred to as *illuminators of opportunity* from the local environment. This illumination could be (as already used for passive drone detection) DVB-T [54] or telecommunication signals such as LTE [55] and 5G [56]. This work will not directly consider the drone detection using passive radars, although some of the research findings later in the thesis may indirectly be of benefit to passive systems. As passive systems rely on opportune illumination, their deployment is situational depending highly on an area's geography and passive signal

availability, which provides an opportunity to deploy radar where spectrum allocation is an issue. Also, the returns measured in passive systems are very weak compared to those from active systems where a powerful transmitter is used. The proposed advancements of synthetic data in Chapter 6 are extendable to passive systems, and the unsupervised learning approach that is the topic of Chapter 8 has been applied to passive systems previously [57]. Passive radar is an exciting research trend that may be very useful for specific environments and applications, but this thesis will only directly consider drone detection with active radar systems that will be more appropriate in the majority of cases.

Table 1.3 shows a selection of existing counter-drone radar products that exist to detect the presence of drones in an airspace. Some details of their operation within the scope of this section are shared that contains information garnered from their publicly available datasheets.

Table 1.3: Some CUAS Commercial Radar Manufacturers and Systems

Image	Manufacturer & System	Operation
	Hensoldt Spexer 2000 3D Mk2	• Pulse-Doppler • X-Band, AESA • $120° * 15°$ Coverage • 9km Drone Detection Range • > 300 Targets Simultaneously Tracked
	Blighter A800 Mk 2	• FMCW • Ku-band (15.7-17.2 GHz) AESA • <7km Drone Detection Range • <700 Targets Simultaneously Tracked • $0.1ms^{-1}$ Doppler Resolution
	Robin Radar Systems IRIS	• FMCW • X-band, Mechanical Scan • 0.6m Range Accuracy • 4 km Drone Detection Range • <700 Targets Simultaneously Tracked • $0.1ms^{-1}$ Doppler Resolution
	Aveillant Ltd. Now a part of the Thales organisation Gamekeeper 16U *The radar system used in this work*	• Pulse-Doppler • L-band, Staring Radar • $90° \times 30°$ Coverage • 5 km $0.01m^2$ Drone Detection Range • <700 Targets Simultaneously Tracked • $0.1ms^{-1}$ Doppler Resolution • See also Table 2.1

1.2 Objectives and Contributions

The overarching aim for this project is to improve general radar capabilities of distinguishing potentially unsafe drones from birds in the sky. Following on from the previous section where the drone detection challenge and possible sensor solutions were introduced, this section will explicitly state the tasks conducted in this project.

1.2.1 Problem Statement

Due to the proliferation of consumer drone systems, efforts to improve detection and distinguishment of these targets are required to protect civilian infrastructure from terrorism and other threats. This thesis seeks to improve a radar's ability to classify the difficult to distinguish targets of birds and drones. Radar is the focus of this work, as it is a mature sensing technology that possesses useful qualities (operating over long ranges, in all lighting and most weather conditions). The small size of drone and bird targets, compared with traditional radar targets (such as planes, helicopters) motivates further investigations into improving signal processing to optimise the detection of these targets. It is desired to use the modern suite of deep learning classification tools for this challenge in radar. However, as these methods rely on a large dataset, there is a difficulty to apply and validate these algorithms in real operational environments. This work constructed a large dataset of real drone and bird radar signatures taken from a pulse-Doppler radar in a highly challenging, urban environment (Chapter 4), and implemented a classification benchmark using a chosen pre-trained deep learning classifier, AlexNet, evaluating the effectiveness of the approach. The classifiabilility of four different drone models is investigated alongside a wide variety of real bird signatures (Chapter 5).

This preliminary work is then used to validate a number of new approaches in the field. Firstly, to overcome the lack of availability of drone signatures, a new simulation approach for creating highly realistic synthetic drone micro-Doppler returns is introduced (Chapter 6) and this is assessed in its ability to be used as training data for classification of real drones (Chapter 7). These approaches take place in the Supervised learning domain, where efforts to label large

amounts of data was necessary.

To overcome the difficult and expensive process of labelling a large dataset, this work also provides an initial investigation of a foundational Unsupervised learning approach, the autoencoder (see Chapter 8) that is closely analogous to the CNN classifier used in the supervised investigation. Section 8.1 describes the autoencoder principle, then explains several common extensions to the approach and reviews where radar works have used these extensions for target classification and other applications. Section 8.2 uses a simple convolutional autoencoder on the collected bird-drone radar dataset to explore its efficacy in a number of regards. This method is shown to be as effective as transfer learning approaches that allows a limited radar dataset to effectively train a classifier for these airborne targets. In addition, unsupervised clustering and exploration with interpretation of the unsupervised performance is had in-and-of-itself, as well as with the relationship with subsequent supervised classification results.

1.2.2 Publications

This work has resulted in the following first-authorship publications, starting with the most recent:

- [4] - D. White, M. Jahangir, C. J. Baker, *et al.*, "Comparing optical transfer learning and autoencoder pretraining for radar bird-drone discrimination," Unpublished - To appear at 2024 IEEE Radar Conference (RadarConf24), Denver, Colarado, USA, 2024

- [3] - D. White, M. Jahangir, C. J. Baker, *et al.*, "Urban Bird-Drone Classification With Synthetic Micro-Doppler Spectrograms," *IEEE Transactions on Radar Systems*, vol. 2, pp. 167–179, 2024, ISSN: 2832-7357. DOI: 10.1109/TRS.2023.3326317

- [2] - D. White, M. Jahangir, J. P. Wayman, *et al.*, "Bird and Micro-Drone Doppler Spectral Width and Classification," in *2023 24th International Radar Symposium (IRS)*, vol. 2023-May, IEEE, May 2023, pp. 1–10, ISBN: 9783944976341. DOI: 10.23919/IRS57608.2023.10172408

- [1][4] - D. White, M. Jahangir, M. Antoniou, *et al.*, "Multi-rotor Drone Micro-Doppler Simulation Incorporating Genuine Motor Speeds and Validation with L-band Staring Radar," in *2022 IEEE Radar Conference (RadarConf22)*, IEEE, Mar. 2022, pp. 1–6, ISBN: 978-1-7281-5368-1. DOI: 10.1109/RadarConf2248738.2022.9764352

Additionally, the author was involved in the writing or publishing of the following publications, in order of decreasing involvement:

- [5] - G. Atkinson, M. Jahangir, D. White, *et al.*, "The Role of Target Signatures in Bird-Drone Classification," in *2023 IEEE Radar Conference (RadarConf23)*, San Antonio: IEEE, May 2023, pp. 1–6, ISBN: 978-1-6654-3669-4. DOI: 10.1109/RadarConf2351548.2023.10149665

- [6] - X. Ren, M. Jahangir, D. White, *et al.*, "Estimating physical parameters from multi-rotor drone spectrograms," in *International Conference on Radar Systems (RADAR 2022)*, Edinburgh: Institution of Engineering and Technology, 2023, pp. 20–25, ISBN: 978-1-83953-777-6. DOI: 10.1049/icp.2022.2285

- [7] - M. Jahangir, G. M. Atkinson, D. White, *et al.*, "Networked staring radar testbed for urban surveillance: status and preliminary results," pp. 471–476, 2023. DOI: 10.1049/icp.2022.2363

- [8] - M. Jahangir and D. White, "Good practices and approaches for counter UAV system developments – an industrial perspective," in *Radar Countermeasures for Unmanned Aerial Vehicles*, Institution of Engineering and Technology, Oct. 2021, pp. 363–383. DOI: 10.1049/SBRA543E{_}ch12. [Online]. Available: https://digital-library.theiet.org/content/books/10.1049/sbra543e_ch12

- [9] - M. Jahangir, D. Griffiths, D. White, *et al.*, "Development of a networked photonic-enabled staring radar testbed for urban surveillance," *IET Radar, Sonar and Navigation*,

[4]This publication won the author 3rd place at the student paper competition at the IEEE Radar Conference 2022, New York.

no. October, 2023, ISSN: 17518792. DOI: 10.1049/rsn2.12524

- [10] - D. Griffiths, M. Jahangir, D. White, *et al.*, "Urban Clutter Analysis for Drone Detection using L-band Staring Radar," *2023 IEEE International Radar Conference (RADAR)*, pp. 1–6, 2023. DOI: 10.1109/radar54928.2023.10371017

- [11] - J. P. Wayman, G. Atkinson, M. Jahangir, *et al.*, "L-band radar quantifies major disturbance of birds by fireworks in an urban area," *Scientific Reports*, vol. 13, no. 1, pp. 1–11, 2023, ISSN: 20452322. DOI: 10.1038/s41598-023-39223-1

1.2.3 Contributions

The three key outcomes of the work to benefit drone detection capabilities are:

- The collection (Chapter 4) and benchmarking (Chapter 5) of real-world drone and bird datasets measured in a range of weather conditions, locations and severity of environment using a range of modern classification approaches). This work was published in [3] and supported the following.

- The proposal (Chapter 6) of a new drone μDoppler simulation technique incorporating drone motor speeds that was originally published in [1]. Subsequently, this simulation was extended to provide an approach for creating realistic synthetic data using real backgrounds and power spectral attenuation of sidebands. This synthetic data was used as training data (Chapter 7) with an investigation of resulting accuracy as a function of synthetic data fidelity, which was published in [2].

- A thorough investigation of unsupervised learning using a simple convolutional autoencoder is presented (Chapter 8) - first by reviewing existing radar demonstrations, and then showing in detail the many available outputs and analysis of the model (including classification) for the collected drone and bird dataset. The findings from this effort enables flexible training of classifiers to be better optimised on modest amounts of radar data. The

work also illustrates the unsupervised method to be effective for learning automatically from the dataset through its latent representations and reconstructions. Part of this work was published in [4].

1.2.4 Thesis Outline

Chapter 2 reviews the essential radar techniques from its basic operation to precise drone tracking and enabling of classification. Following this, Chapter 3 separately reviews the fundamentals of modern classification techniques, finishing with a review of their application in radar and for radar bird-drone classification. Chapter 4 presents the detail for how a large dataset of bird and drone radar data was collected in the urban Birmingham city airspace, highlighting the practices and targets that were used for data collection and in the remainder of the work. Chapter 5 reviews available literature reports of deep learning classification achievements applied to various radar applications including that of bird and drones; then, benchmarks of a deep learning classifier are created using different target configurations to show that drone classification is possible even in the absence of the characteristic μDoppler returns drones are often observed to produce. The method that created these classification benchmarks are repeated in following chapters where novel augmentations to the dataset and classifier training process are made. Chapter 6 addresses the problem of achieving large datasets of radar-drone signatures for deep learning through the demonstration of a multi-speed multi-rotor simulation technique that produces highly realistic synthetic drone μDoppler spectrograms [1]. Chapter 7 applies the new simulation approach for the synthetic training of a CNN with comparisons to the real benchmarks created in Chapter 5 [2], and explores the benefits of the approach beyond the benchmarking performed. Chapter 8 explores unsupervised learning with a convolutional autoencoder to overcome the shared difficulty of data labelling and to learn a latent representation of radar targets that naturally clusters and reproduces synthetic samples. Finally, insights are presented of the quality of a latent representation's relationship with subsequently achieved classification results when using the pre-trained convolutional layers. This work will be summarised in Chapter 9 containing conclusions of the work presented as well as directions for future work.

Chapter 2

Radar Essentials

Radio Detection and Ranging (RADAR) today is one of the most mature forms of electrical engineering, as its simple principles enabled it to be demonstrated before 1900 and its applications wide-ranging due to the contactless nature of its operation. Developments of radar capabilities were accelerated by transport, war, radio, meteorology, aviation, yet it is still used to break new ground in the challenges of self-driving cars, extraterrestrial touchdown of spacecraft and medical monitoring of patients privately at home. Modern radars gather a vast amount of data, and a great deal of time can be spent, and progress made, in carefully analysing the radar returns to solve the radar specific problems of detection, tracking and classification. Radar has now long been capable of distinguishing cars and aircraft, and a current challenge is to differentiate alike moving targets and tracking very small and highly manoeuvrable objects. This work's focus is to offer advancement on the techniques for the discrimination of bird and drone targets via classification of spectrograms, that is a representation of the raw data captured by coherent radar. Section 2.1 will present the basic theory of the radar as relevant to the undertaking of this project. Then, Section 2.2 shows how coherent radar returns are mathematically processed in a pulse-Doppler format to yield the velocity spectrum of the environment. Finally, Section 2.3 will discuss the details of the radar used in this project, and how its attributes are used in the research following.

2.1 Introduction of Radar Capabilities

This first subsection will explain the key enabling phenomena that allow radar to be employed as a surveillance sensor discussing signal, range measurement, noise, velocity and angular measurements.

2.1.1 Signal and Noise

Radar transmits large amounts of EM energy, either continuously or in discrete time limited pulses, and receives faint echoes permitting conclusions of the environment to be made if the return echo power is greater than the noise signal power. There are many types of noise in radar systems, with the most basic and unavoidable noise in any radar system being thermal noise. Any object not at absolute zero will possess an amount of thermal energy, that is going to cause charged particles to move, thereby induce electric and magnetic fields according to Gauss's laws, the Biot–Savart law or in general according to Maxwell's equations [58]. Thermal noise will be mathematically compared with target echo signals in the following section. Other forms of noise of particular interest to the radar community are phase noise [59], which is internal and becomes a limiting factor when strong clutter (stationary objects in a radars field of view, e.g. the ground and buildings) is present masking weak target echos, and interference noise; where external EM sources are undesirably received by a radar (intentionally [60] or unintentionally [61]) and compete with desired application specific signals.

2.1.2 Measurement of Range

Radar at its core is a range measurement instrument. For pulsed operation, measuring the round trip time, $\mathcal{T}$, of each short pulse of light duration, τ, allows the range, R, to be determined. Two targets will not be resolved if they are closer together than the distance covered by the pulse of light in its transit, and so exists a fundamental range resolution, ΔR, on range measurements

with pulsed radar. This is expressed mathematically as equation 2.1,

$$R = \frac{c\mathcal{T}}{2} \pm \Delta R = \frac{c}{2}(\mathcal{T} \pm \tau) \ . \tag{2.1}$$

Reducing the pulse width, τ, increases the range resolution. There is also a maximum range of radar, beyond which it will be ambiguous to what pulse caused that return. If a pulse returns from a target beyond half the distance that light travels between pulses, $\mathcal{T}$, it will appear as being only so far away as the distance from this maximum range, $\frac{c\mathcal{T}}{2}$. Thus the maximum unambiguous range is given by $R_{max} = \frac{c(\mathcal{T}-\tau)}{2}$. The Signal-to-Noise Ratio (SNR) of the returning signal is maximised by convolving it with the complex conjugate, time-reversed received signal. This is the theoretical best filter that could be applied to maximise signal strength amongst presence of additive noise [62]. The resultant analogue signal is then sampled repeatedly in time, which produces a 1-D array of digitised samples which make up the *range gates*, or fast-time axis, of the received signal.

2.1.3 The Radar Range Equation

Generally, the *transmitted* power of a radar is a very important design parameter as it sets the limits of what the minimum target size at maximum range is. A target will not be detected when the power induced in the *receiving* sensor from a target is comparable with the power induced from thermal or other noise sources. If there is a detectable presence of signal, it can be measured relative to the noise floor with a SNR. The Radar Range Equation (RRE) predicts the SNR of a point target in space using target and radar hardware parameters. An antenna emits EM radiation with a transmission power, P_t. The transmitted power density, Q_i, of a radar depends on the area over which it is cast. An isotropic antenna is scaled by the surface area of the sphere of propagation. Radar antennas are usually designed to efficiently direct most of the transmitted energy in a specific direction and this is quantified as the transmit gain, G_t. The

total transmit power of the radar is given by equation 2.2,

$$Q_i = \frac{P_t G_t}{4\pi R^2} \,. \tag{2.2}$$

A small portion of the output energy reflects off a target and returns to the antenna. The power radiated back towards the receiver is the transmitted power density multiplied by the target's Radar Cross Section (σ_{RCS}), σ_{RCS} which has units of m^2 and a value that depends on the target characteristics relative to radar EM radiation. The resultant value is a complicated relationship of material, position and aspect angle of the target. The RCS of a target is either measured experimentally (such as in an anechoic chamber), theoretically derived in the case of a simple geometry and homogeneous material, or numerically calculated via EM propagation simulation tools such as finite element analysis or Yee cells [63], [64]. The power density (Wm^{-2}) at the receive antenna after scaling again for the surface area of propagation is

$$Q_r = \frac{Q_i \sigma}{4\pi R^2} = \frac{P_t G_t \sigma}{(4\pi)^2 R^4} \,. \tag{2.3}$$

The power received P_r at the antenna is equation 2.5 scaled by the receive gain of the antenna, G_r, which is a value defined by the antenna design and geometry with the well known relation [65], [66],

$$A_e = \eta_a A = \frac{G_r \lambda^2}{4\pi} \,, \tag{2.4}$$

where $0.5 \lessapprox \eta_a \lessapprox 0.8$ is the *antenna efficiency*, λ is the EM radiation wavelength, A is the radar antenna area and A_e is the effective area of the antenna. The target signal power on receive of a single pulse is,

$$P_r = Q_r A_e = \frac{P_t G_t A_e \sigma}{(4\pi)^2 R^4} = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R^4} \,. \tag{2.5}$$

Thus, the received power is dependant on several radar design aspects. The operational wavelength is one of the key choices that requires consideration of the size and structure of the

expected targets, which in turn effects the operational range, size and cost of the system. Operating frequency is so important in radar that it is used as a method of categorisation. Radars that operate in a certain frequency band are given a letter to which they can are categorised [67], but this convention goes beyond radar and is used to describe the frequency of light used in many EM applications. The various radar bands as used in real applications were shown earlier in Table. 1.2 and discussed in Sec. 1.1.4.

In the absence of an external signal, the power induced in the antenna is read as noise, due to EM fields induced by thermal movement of charges at ambient temperatures. This type of noise is uniformly distributed over all measured frequencies - its power spectral density is constant/uniform and is commonly referred to as *White Noise*. As noise signals only affect the radar performance if their frequency falls in the receiver bandwidth, B, we have the total noise power,

$$P_n = k_B T_s B = k_B T_0 F B \ , \tag{2.6}$$

with $T_s = F\, T_0$ is the system noise temperature, with the noise factor, $0 < F < 1$ and $T_0 = 290K$. Comparing the power of the signal P_r and of the noise, P_n gives us the SNR of the target,

$$SNR = \frac{P_r}{P_n} = \frac{P_t G_t G_r \lambda^2 \sigma \tau}{(4\pi)^3 R^4 k_B T_0 F} \ , \tag{2.7}$$

where we have assumed an uncompressed pulse, $B = \tau^{-1}$. This equation is the simplest end-to-end equation for SNR of a point target, and in practice many additional factors can be added to compensate for losses commonly including effects of the transmitter, receiver, propagation through atmosphere and losses that occur in signal processing from the discretisation of analogue signals to digital, such as scalloping [65].

2.1.4 Introduction to Coherent Radar

A laser, a light bulb or a LED all emit EM radiation, which is a source of energy that does not require any propagation medium. Maxwell's equations show that an oscillation of an electric field will induce a magnetic field, which will induce an electric field etc, perpetually, until the propagation direction intercepts a zone with a different relative electric or magnetic permittivity. This property allows for controlled propagation of energy by a user. A powerful or bright source of EM radiation will have many 'photons', instances of a perpetual EM oscillation. If all of the EM oscillations from a source occur with identical phase, which requires something more than having the same wavelength, frequency and propagation speed (of light, c), then the source is called *coherent*, and a simple mathematical model of a sinusoid is enough to describe the light enabling a huge range of applications and measurements to be performed. In pulsed radar, the phase of each pulse is continuous with respect to the pulses before and after it, as if a continuous coherent beam had a fan placed immediately in front of it. As will be discussed in the following subsection, a coherent radar beam can be electronically steered in cross-range by controlling the initial phase shift of each radiating element in an array. On the radial axis, a coherent interrogation of the environment allows for the measurement of Doppler shift in the radar radiation that is used to precisely measure target velocity which is to be explained fully in Section 2.2.

In addition, a fine control of phase in a radar system beyond coherency can provide further opportunities to improving the performance of a given hardware. In pulsed radar, flipping the phase midway through a pulse or sweeping the frequency of a pulse can provide a finer range resolution and increase the maximum unambiguous range by providing an additional reference of frequency to range measurements from time delay duration. Equation 2.1 stated that a pulsed radars range uncertainty ΔR is proportional to the pulse width, which is true for a fixed frequency pulse. If there is a frequency modulation, the bandwidth should be used: $\Delta R = \frac{c}{2B}$. With a coherent deterministic frequency modulation, pulses are not required for a range measurement thus providing some advantages in certain applications. A CW radar

without frequency modulation is not able to measure a target's range, as there is no time delay to be measured with a constant phase input, and its uses are limited to applications of velocity measurement. In Frequency Modulated Continuous Wave (FMCW) radar, a comparison of emitted and detected frequencies allows an instantaneous range measurement, with a resolution dictated by the bandwidth of the waveform as above. FMCW commonly operates at high frequency for short to medium range applications that allows for a fine Doppler measurement and coherent beamforming approaches for angular measurements. Together with their low Size, Weight and Power (SWAP) FMCW systems now sees active research such as for the automotive industry [68], and for radar measurements from drone platforms [22].

2.1.5 Angular Measurement

Section 2.1.2 described the pulsed radar approach for measuring range of a target. Now, how radar can provide azimuth and elevation information of a target will be briefly reviewed. Originally, mechanical antennas producing a fan or pencil beam were mechanically steered with measurements processed with the sensor facing direction to produce 3-D detections. The *beamwidth* of the antenna is the cross-range extent between which the transmitted or received EM energy is non-zero, and is dictated by the antenna geometry. Measurements more fine than the absolute beam width could be made by comparing the power received while factoring intentional changes in the antenna direction. However, modern radars use electronically steered phase arrays for measurement of radar directionality that has many advantages beyond greatly reducing the mechanical engineering burden at the transceiver. Typically, a phased array antenna has equally spaced radiating apertures each separated by $\frac{\lambda}{2}$ so that the radiation from one aperture is at null at every other aperture site. This arrangement produces nulls at angles where destructive interference occurs. Imperfections in the spacing or the transmitted radiation contribute to reducing the antenna efficiency, η_a. The beamwidth of a phased array antenna is dictated by the aperture length. For a 1-D or square antenna of side length, D, the first nulls of the beams fall at $\pm sin^{-1}(\frac{\lambda}{D})$, with the width of the signal within 3dB of the peak approximately $0.87\frac{\lambda}{D}$. For a circular antenna, the first nulls of the beams fall at $\pm 1.22\frac{\lambda}{D}$ with a 3dB beamwidth of $\pm 1.03\frac{\lambda}{D}$

[69]. The appearance of λ in the numerator dictate that higher frequency systems can form finer beams. These values can be adjusted by tapering or thinning the element placement, to achieve advantages in certain applications but not without some compromise. Target detection can be made more precise than the beamwidth with techniques by comparing the strength of returns near a target with small offsets to measure an amplitude response, known as monopulse. The phased array beam pattern can be steered by adjusting the phase shift incrementally along the array surface. The desired angle propagation, θ, can be achieved by adding an additional phase shift, $\Delta\phi$, to every successive aperture in the array,

$$\Delta\phi = \frac{2\pi D sin\theta}{\lambda} \tag{2.8}$$

that will be in the direction of phase shift increase. This concept applies identically to both the steering of the transmitted beam *and* the direction of the received beam. The phase shift across the face of the array causes increased constructive interference in the desired direction angle, while inducing increased destructive interference at the antenna site for angles it is looking away from.

2.1.6 The Doppler Effect and Radial Velocity Measurement

The Doppler effect arises when a receiver or transmitter of a signal is moving, and the signal is observed at a higher or lower frequency than at its source. Precise measurements of velocity can only be made if the frequency of the emitter is known. The observed frequency, f, of a signal with stationary frequency, f_0 due to the Doppler effect, in Hertz, for a target of radial velocity, v is

$$f = f_0 \pm \frac{2v}{c} = f_0 \left(1 \pm \frac{2v}{\lambda}\right), \tag{2.9}$$

where it assumed that $v \ll c$ is non-relativistic. A direct measurement of received frequency with comparison to known transmitted frequency would allow a radial velocity measurement. However, measuring small shifts of high frequency EM radiation as are used in radar is not easy and so is not done directly. Instead to perform precise velocity measurements, the radar signal

processing chain in essence compares the received signal with the output transmitted signal to garner the same information, as is detailed in the following section.

2.2 Pulse-Doppler Processing and Phenomenology

In the previous section it was stated that the Doppler frequency could be measured to enable a calculation of target velocity. How this frequency is measured via pulsed radar will be shown and elaborated as the topic of this section. The interrogating EM beams of the radar exist at a specific carrier frequency, f_0. Directly measuring a 150 Hz shift of a GHz frequency would be an extremely fine and unreliable approach. It is the practice in radar to *down*convert the high frequency EM signal down to a lower intermediate frequency or to baseband, then, a Fast Fourier Transform (FFT) can be used to turn a cyclic signal intensity into a peak at the specific Doppler frequency. To do this, it is required to look at multiple consecutive pulses so the effect of the target's Doppler shift can be collected. It is thus required that the pulse train is coherent - that the phase of one pulse is continuous with the next - as this allows for a frequency measurement via the rate of change of phase.

2.2.1 Down Converting Radar Signals to Baseband

We can treat the signal of a single pulse as a sinusoid with an initial phase offset, θ,

$$x(t) = A\ cos(2\pi f_0 t + \theta) = A\ cos(\omega t + \theta) = A\ \Re\{\ e^{i\omega t + \theta}\ \} \tag{2.10}$$

where ω is the angular frequency of radiation, and t is duration of propagation starting at $t = 0$. A, is a scale factor, which will be dictated by the RRE and so will be left out of the derivation. If there was a target at a radial distance R_0 from a monostatic antenna then the returning signal will have the form

$$y(t) = cos(\omega(t - \frac{2R_0}{c}) + \theta)\ , \tag{2.11}$$

where a time delay equal to the propagation delay has been included. Now we will expand this equation into a more useful form,

$$\begin{aligned} y(t) &= cos(2\pi f_0(t - \frac{2R_0}{c}) + \theta) \\ &= cos(2\pi f_0 t - \frac{4\pi f_0}{c}R_0 + \theta) \\ &= cos(\omega t + \theta - \frac{4\pi}{\lambda}R_0) = cos(\omega t + \theta') \\ &\quad \text{defining } \ \theta' = \theta - \frac{4\pi}{\lambda}R_0, \end{aligned} \tag{2.12}$$

where $c = \lambda f_0$ has been employed. Now, the received signal is in a form similar to that of the emitted signal $x(t)$ from equation 2.10, which can be accessed by upconverting the local oscillator signal and allows for a local phase comparison which enables Doppler measurements to be made via a separation of time scales. Multiplying the received signal by two times the reference oscillator signal, $x(t)$, and using the trigonometric equation $2cos(a)cos(b) = cos(a-b) + cos(a+b)$,

$$\begin{aligned} cos(\omega t) \cdot cos(\omega t + \theta - \frac{4\pi}{\lambda}R_0) &= cos(\theta - \frac{4\pi}{\lambda}R_0) + cos(2\omega t + \theta - \frac{4\pi}{\lambda}R_0) \\ &= cos(\theta') + \cancelto{0}{cos(2\omega t + \theta')}^{LPF \to 0} \\ &= cos(\theta') \end{aligned} \tag{2.13}$$

The term that has been cancelled is as a result of a Low Pass Filter (LPF) that attenuates the twice carrier frequency term, leaving a baseband sinusoid that is sampled at θ'. So after mixing and low pass filtering, the return signal we have is a Direct Current (DC) value for one pulse reflected off a moving target,

$$y(\mathcal{T}_m) = cos(\theta - \frac{4\pi}{\lambda}(R_0 - v\mathcal{T}_m)) \ , \tag{2.14}$$

where $\mathcal{T}_m = m * PRI$ where m is the pulse number, and v is the target velocity along the radial

axis with $|\underline{v}| = v$ towards the antenna, which is assumed to be constant. Here, it is assumed that the target has constant (radial) velocity, such that target's position can be described with a fixed offset of velocity $\times$ PRI. $\mathcal{T}_m$ is equivalent to t, but discretized at the instances of radar temporal sampling. This framing relies on an additional assumption, known as 'stop-and-hop' [65], [70], where it is assumed the target does not move within the duration of the waveform, which is valid for the vast majority of cases where the target's velocity is much smaller than the speed of light. As time progresses, the distance of the target to the antenna will change and this is measured by the radar via the phase difference between successive pulses. A target's absolute position can be measured in radar via the range gates which are, generally for long range applications, in 10s to 100s of meters. By measuring the phase shift between successive pulses, it is possible to determine the relative range of a target compared to the previous pulse to a sub-wavelength accuracy. A train of many successive pulses are required to turn the DC return signal into an Alternating Current (AC) response capturing a target's dynamics. However, the relationship between $y(\mathcal{T})$ and v is not sufficient to determine the direction of a target's motion, as a trigonometric *cos* function has a time reversal symmetry. Measurements of amplitude and phase are ambiguous. To extract target directionality, an additional mixing process is done to create a signal orthogonal to the previously calculated one, and is represented pictorially in Figure 2.1. Here, a second signal is created via the same mixing process, the only change being that the mixing term is orthogonal to the previous. There are two channels leading to two outputs. Where the mixing term is in phase with the return signal, the in-phase (I) channel, and where the mixing term is orthogonal to the return signal, the quadrature (Q) channel. The mathematical rationale is the same as for the in-phase case with the equivalent *sin* formula, $2cos(a)sin(b) = sin(a + b) - sin(a - b)$.

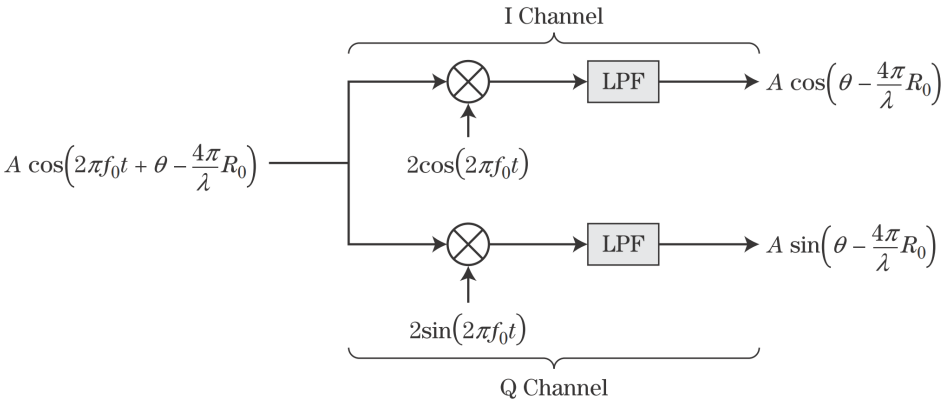


Figure 2.1: The coherent I/Q detector. A carrier frequency with a small modulation due to range $(-\frac{4\pi}{\lambda}R_0)$ is input and two orthogonal signals are output. From [65].

Adding the orthogonal in-phase and quadrature channels as a complex sum allows an independent measurement of the return signal's phase and amplitude to be done. This is called the analytic signal and is constructed like so,

$$y[0] = y_I[0] + iy_Q[0] = cos\theta' + isin\theta' \propto A'e^{i\theta'} \ , \tag{2.15}$$

where 0 corresponds to the 0^{th} pulse of a sequence with successive pulses are appropriately named, and the amplitude which will locally be a function of phase and RRE signal strength. Alternatively, a quadrature signal could be created from a single channel, real only input by applying the Hilbert transform [71] to produce an analytic signal. The Hilbert transform creates an orthogonal by 90° phase signal in the complex domain which is an essentially equivalent digital solution to the analogue I/Q demodulation described previously, of which within itself there are other proposed ways of doing this [72]. With the coherent detector conceptually set up, a radar's Doppler frequency measurement of a target moving with velocity, v, towards the radar will be described. A train of M pulses, each labelled by m, are directed towards the target. The distance to the target will be less after each pulse, $R = R_0 - v\mathcal{T}_m$. The I/Q signal in this case will be,

$$
\begin{aligned}
y[m] &= y_I[m] + iy_Q[m] \\
&\propto exp\{\, i[\, \theta - \frac{4\pi}{\lambda}(R_0 - v\mathcal{T}_m)\,]\, \} \\
&= exp\{\, i[\, \theta - \frac{4\pi}{\lambda}R_0 + 2\pi\frac{2v}{\lambda}\,\mathcal{T}_m)\,]\, \} \\
&= exp\{\, i[\, \theta - \frac{4\pi}{\lambda}R_0 + 2\pi f_D\mathcal{T}_m\,]\} \\
&= exp\{\, i[\, \theta' + 2\pi f_D\mathcal{T}_m\,]\} \,.
\end{aligned}
\tag{2.16}
$$

where $f_D = \frac{2v}{\lambda}$ is the target's measured Doppler frequency. These samples, gathered on a per-pulse basis, are referred to often as the *slow-time* axis. The arbitrary phase offset, θ, now factors in the phase difference due the *initial* range of propagation to the target which itself is arbitrary and unrelated to velocity measurement that was desired.

2.2.2 Fourier Analysis of Time-Series Signals

Treating a pulse train in this way will generate a sinusoidal at the target's Doppler frequency which will reveal a peak after a digital Fourier transform implemented with the FFT algorithm [73]. In radar data, this can be referred to as the Doppler axis. If an infinite number of pulses were used, the target maintained an indefinite constant velocity, and the PRF approaches infinity, then the peak in frequency space would be a Dirac delta ($\delta(\underline{z})$) function, where $\underline{z}$ is all space. The more pulses coherently integrated together in this fashion will give a taller and narrower peak, as the discrete Fourier transform will have a longer sinusoid to which it can specify its Fourier amplitude, and so has less intrinsic uncertainty over what the Doppler frequency is. See Figure 2.2 below, where the FFTs of a 512 and 4096 length *sin* function are displayed side by side.

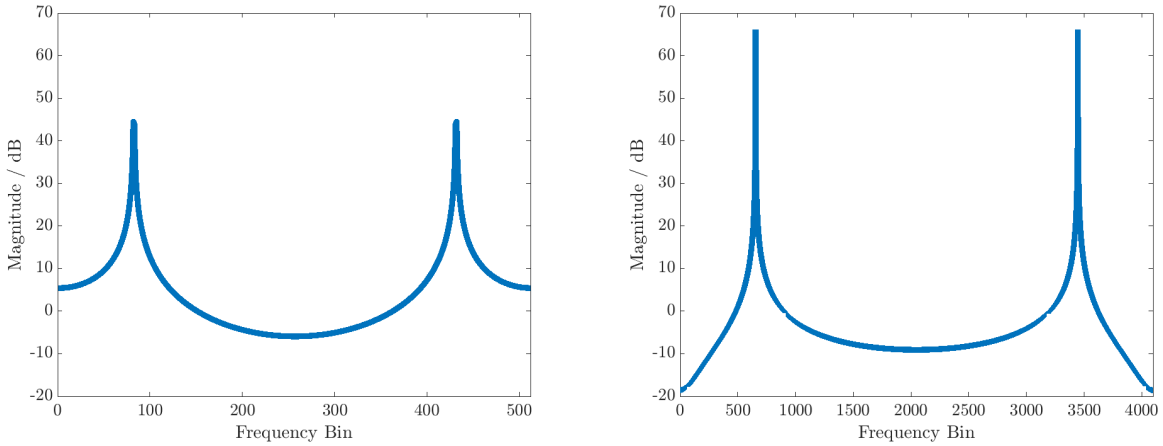


Figure 2.2: A post FFT (no window) real sinusoid. On left, 512 pulses were used and on right, 4096.

It is observed that the peak amplitude of the Fourier signal is greater when the FFT length is longer. The spectral width of the Fourier peaks grows narrower and their magnitude grows larger with a longer input sequence. For the detection of small targets in radar, the more pulses coherently integrated together means a higher SNR of peaks along the Doppler spectrum. The (absolute) increase in SNR is linearly proportional to the number of pulses coherently integrated, and this can be reflected in the RRE by multiplying equation 2.7 by the number of pulses. However, increasing the Coherent Processing Interval (CPI) via capturing more and more pulses is not a total solution, as this analysis assumes that a target has a constant velocity. If the target accelerates during a CPI, the returns will be spread over several Doppler bins and the gain in SNR will be less than the theoretical maximum. In practice, a compromise must be made between FFT length and CPI duration, which is made easier if the radar employs a high PRF. In Figure 2.2, where a real sinusoid has been Fourier transformed, two peaks are observed, demonstrating the importance of creating a complex I/Q signal to determine target direction. Similarly as for range, there is a maximum unambiguous Doppler frequency that if exceeded, will appear as a target moving in the opposite direction. This value is $\Delta f_D = \frac{PRF}{2}$ and is dictated by the Nyquist sampling that a frequency can only be totally recreated if the sampling rate is greater than two times the maximum (baseband) frequency of the interrogated signal. This means the maximum velocity unambiguously measured is $v_{max} = \frac{\lambda\, PRF}{4}$, beyond which peaks

will begin to appear in the negative velocity direction. In high PRF surveillance radar systems, this value is greater than the expected velocity of any target.

2.2.3 Radar Signal Processing Choices for Drone Detection

There are a few pulse-Doppler, staring radar research works such as [74], [75], that offer kilometer order detection capabilities of birds and drones. These radars are industrial products and are used and implemented in real scenarios. There is not as much literature present of pulsed, L-band staring radar research for drone applications [76], so this subsection will begin by sharing some relevant key differences between pulse-Doppler and FMCW systems. In literature, much work of the analysis of bird and drone spectra is using FMCW radar, and often at much higher operational frequencies at, X- [77], [78] and K- [79], [80] and V- band [79], [81] in part due to their comparatively smaller SWAP , and reduced difficulty of setup and experimentation.

Both FMCW and pulse-Doppler radars permit a long dwell of interrogation of targets, that allows a fine resolution of Doppler measurements to be taken. The maximum unambiguous range for FMCW is fundamentally dictated by the chirp duration (in the simplest configurations [78]), similarly to the PRI in a pulsed radar. Generally for pulsed radar, the setting of the PRI is not the limiting factor for range, so it will be chosen to slightly exceed the maximum instrumented range of detection so as not to overload the processor with unambiguous tracks. Conversely, in (FM)CW radar there is no blanking of the receiver so as not to interrupt the phase continuity of returns, and so there is a technical challenge to co-locate a FMCW transmitter and receiver so that there is minimal leakage from the former to the latter. For this reason, creating a high powered, long range FMCW radar is difficult, and so drone and bird detection of FMCW is in the literature limited to ranges not much greater than $200\,m$, which is in fact *less* than the minimum detection range of the L-band staring radar featured in this work, which is $263\,m$. This highlights the specialist build purpose of the low frequency staring radar as a cutting edge technology for drone detection at a long range. Aside from these additional range limitations of FMCW radar, the Doppler acquisition and processing options are common between them which

is the focus of the remainder of this section.

Targets with rotary blades can lead to very distinct Doppler signatures termed Micro-Doppler (μDoppler). Much information about a drone target can be extracted from the μDoppler returns alone. The μDoppler characteristics of rotor blades was first studied in the context of helicopter's large rotating blades [82], [83], from which much of the same modelling can be applied to the assessment of μDoppler returns from rotary wing drones [84], [85]. Doppler returns can be represented via a Short-Time Fourier Transform (STFT) to produce a 2D, time-Doppler spectrogram. This common approach is used for radar-target visualisation [17], [86], and as a method to feed radar-collected data for subsequent classification for drones detection [6], [87]–[97], and indeed many other domains[98]–[103]. The creation of spectrograms is not the only approach for producing and 2D representation for a set interval of radar data. A *cepstrogram* involves applying an inverse digital Fourier transform along the temporal axis of the time-Doppler spectrogram, after applying a logarithm all spectra. This method highlights periodic occurrences of frequency components, and has been used to represent targets in the context of drone detection [88], [104], [105]. Additionally, the *wavelet transform* offers another alternative to the Fourier transform with potential advantages. A Fourier transform, decomposes timeseries information into a succession of *sin* waves for a spectrum of frequencies, while a wavelet transform does the same but with more abstract, decaying oscillations which potentially allow for a better fit to the specific features of the timeseries data. Discrete and continuous wavelet transform were used in the analysis of drone returns for mm wave radar in [106], and [107] which performed classification on synthetic data drone.

For the remainder of this work, spectrograms will be formed using a digital Fourier transform implemented with FFT. There is a trade off in the FFT length to use; if small then the spectrogram will have a low Doppler/frequency resolution but a fine temporal resolution, otherwise it will have a course temporal axis but finer Doppler resolution. A longer FFT also decreases the likelihood of constant target velocity, which will observe increased Doppler spread for highly-manoeuvrable targets. By performing FFTs not consecutively but allowing overlapping durations of time,

maintains the Doppler resolution with an increased temporal resolution [108]. Windowing is used to lower the effect of spectral artefacts [109] such as frequency sidelobes that result from the first and last point of the FFT not necessarily being continuous. The Doppler response in the spectrogram is dictated by the duration of the FFT compared to the duration of a blade rotation; where high update rates allow the measurement of the propeller motion as it rotates, while low update rates slower than multiple rotations of the propeller will manifest sidebands, often called HElicopter Rotor Modulation (HERM) lines. Doppler-time plots, known as a spectrograms, are created with a STFT of timeseries baseband returns. A spectrogram can be of great duration if a moving target is tracked in 3D space by concatenating fast-time samples along its flight path. A detailed analysis of a drone's μDoppler HERM line spectrum was had in [87], which used a C-band CW and a W-band FMCW radar to investigate the spectral properties of an active drone. Figure 2.3 shows a STFT of the raw timeseries data to get a high update rate image of the propeller returns from a single-motor, two-blade model helicopter.

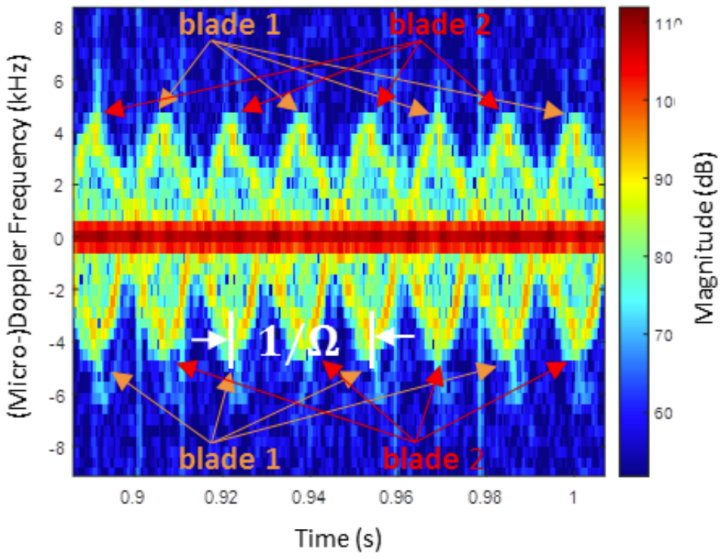
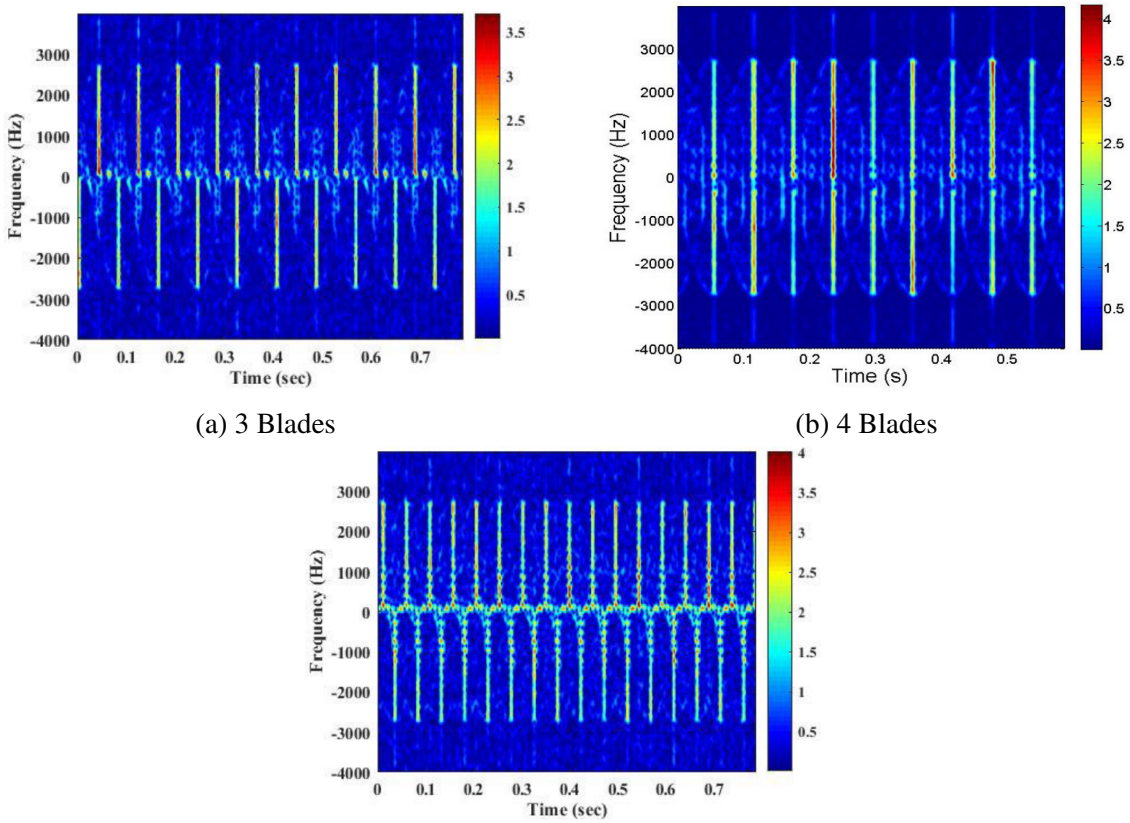


Figure 2.3: STFT of (drone sized) helicopter with 2 blades returns gathered by CW C-band radar. Ω is the drone's motor rate; $\frac{1}{\Omega}$ is the time period for a full rotation. From [87].

Each μDoppler component in the spectrum can be attributed to a target motion. When the rotor blades of a two bladed helicopter line up with the radar's radial axis, there is no μDoppler detected as the blades are moving tangential to the axis Doppler is collected. The blades then

accelerate / rotate until they are positioned tangential to the radial axis where they impart the maximum Doppler shift on the EM radiation, which is seen as the maxima on either side of the 0^{th} Doppler bin, known as blade flashes. If the helicopter had one blade, this would be observed in the returns like a sine wave with μDoppler present only in one direction at a time. It is possible to observe the (anti-)symmetry of the STFT lines to give information of the number of blades present on a target. Figure 2.4 from [110] shows the STFT spectra of helicopters with different number of blades on its single motor. With this information, it may be possible to count the number of blades given sufficient SNR. However, as these examples shown are for a rotor system with only a single motor; in practice using multi-rotor systems, with each motor spinning at a different radar, the described analysis may not be suitable to extract the relevant information from a more complex or chaotic spectrogram.



(a) 3 Blades

(b) 4 Blades

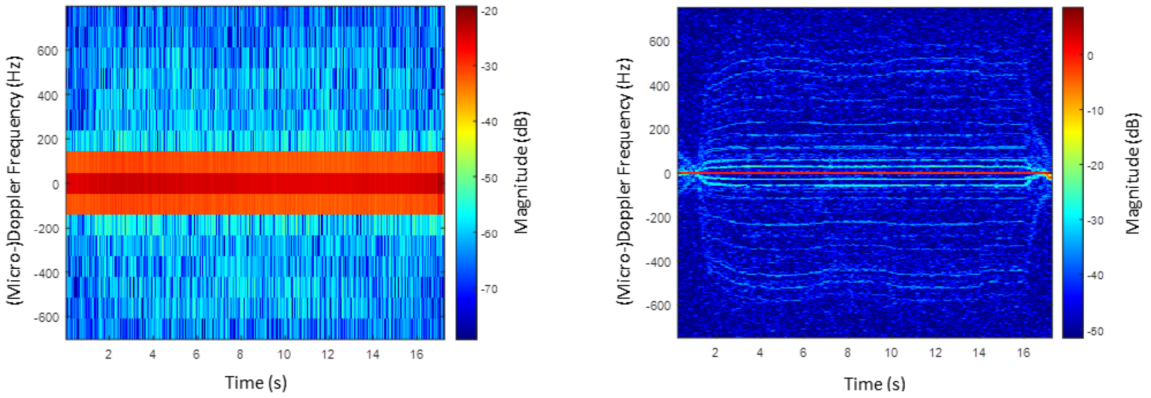
(c) 5 Blades

Figure 2.4: The effect of number of rotor blades on a short-window STFT spectrogram. From [110].

It is possible to estimate the drone blade rotation rate by observing the periodicity, although there would be some uncertainty to which peaks in Doppler velocity correspond to the flashes of full rotation, so an understanding the characteristics of differing blade numbers on drones is important to get a reliable measurement. Then, with the speed of the blades determined by the temporal periodicity of returns, the maximum extent to which a blade flash extends out in Doppler is an indication of how long the blades are [91], [97]. A circularly rotating point of radius, r, and angular speed, Ω, (Hz) will have a constant speed of $2\pi\Omega r$. If the size of the rotor blades is much greater than the radar beam wavelength, then it can be assumed that when the blade is perpendicular to the radar radial axis, the spectrum will feature a maximum instantaneous Doppler, the maxima of which will induced by the very edges of the blade allowing a measurement.

The STFT view of a drone's rotor blade flashes is one that is only viable in short range scenarios, as a series of short time FFTs will not construct a signal with higher SNR than a long CPI over many pulses (except possibly in the scenario of rapidly accelerating targets). The comparative size of the radar's sampling rate and the drone's motor rate greatly affects the spectra observable. For a pulsed radar with a PRF of 7000 Hz and the number of samples used in processing nominally 2048, the ultimate sampling rate, S, of this system is 3.6 Hz. A typical drone's motor speed, Ω is $>$ 100 Hz, meaning the Doppler modulations caused by the drone are undersampled. Figure 2.5 shows a comparison of the short window, $S < \Omega$, and long window, $S > \Omega$ FFT Doppler processing on the same drone target. The more interpretable drone flashes seen in Fig. 2.3 and 2.4 are not of sufficient SNR to be observed by eye or detected by a detector.

Thus, in pursuit of an increased range of operation, a radar can sacrifice the ability to see the blade flashes of a drone return to instead observe the HERM lines seen in Fig. 2.5b to maximise SNR and thus increase detection range. This nature of returns are less less immediately interpretable than in the case of oversampling $S < \Omega$ for a single motor, but is possibly better for interpreting the returns of multiple motors. In the case where the motors speeds are different but at least fairly constant over the duration of a CPI, the Fourier transform finds the periodic signals



(a) Short window FFT leading to blade flashes ($S > \Omega$)

(b) Long window FFT leading to HERM lines ($S < \Omega$)

Figure 2.5: Comparison of long and short window FFT lengths in a low SNR scenario. From [87].

within the returns and provides a gain at these frequencies. Observing Fig. 2.5b, the spectral lines occurring either side of the main body return are from the additional oscillations in the timeseries signal caused by the motor contributions, that are brought out as multiple harmonic peaks at the measured frequency offset.

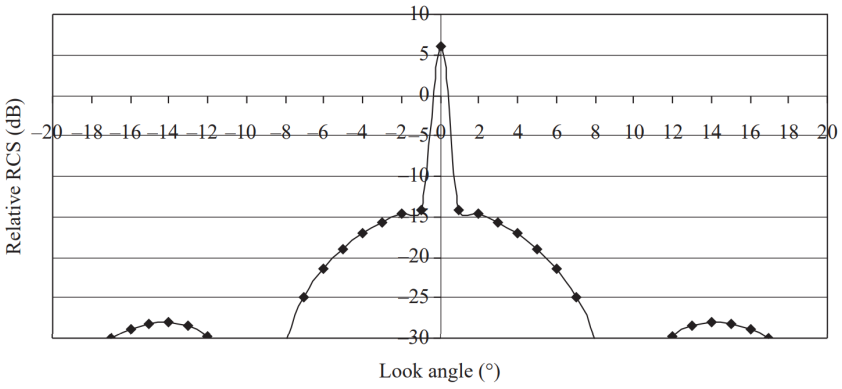


Figure 2.6: RCS of a planar propeller blade at different view angles. From [111].

The intensity of HERM lines has been studied for helicopters in [111] and for drones in [87]. Figure 2.6 shows a numerical simulation of the RCS of a propeller blade of a helicopter at different radial view angles, with the plane of rotation along the radial axis. This pattern will vary as the angle between the plane of rotation and radial axis are adjusted. As drone blade's spin in a pre-determined direction, variation of scattering depending on the angle of attack of

each of the propellers, as well possible increased levels of shielding by the drone body. The flashes also correspond to the point of maximum Doppler, which are the existence of the peaks in Figure 2.4. This brightest peak will be seen by the radar N_b times per full rotation, where N_b is denoted as the number of blades on a single motor. At any other time when the blades are not aligned for maximum RCS, there will be a different and lower contribution to μDoppler much less intense, but yet repeating at Ω. The high intensity lines are called *chopping* lines, and the others are called *spool* lines. With this basic intuition, we can label the HERM lines produced in a high SNR environment to match the origin. In Figure 2.7 the chopping lines are labelled as, generally, the highest intensity peaks, and spool frequency peaks occur in the gaps.

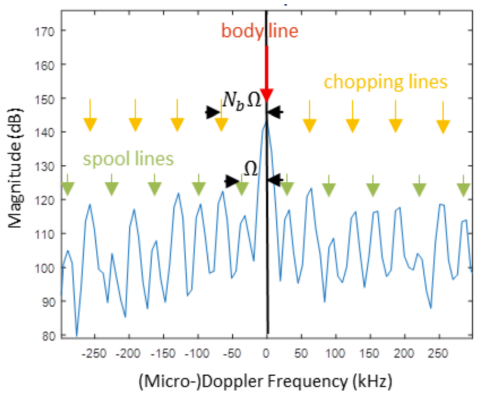


Figure 2.7: Identified chopping and spool lines from a high SNR small helicopter Doppler axis with $N_b = 2$. From [87].

There exists increasingly higher orders of this pattern each decreasing in intensity. In addition, the increase in SNR is very negligible from the chopping to the spooling frequency, only a few *dBs* increase near the centre frequency of the target gaining more as the Doppler increases. This example, again using a single rotating motor structure, is extremely likely not to be as profitable in a multi-rotor case due to again the overlapping spectral lines fluctuating in time will make it very difficult to extract this information under manoeuvring conditions, especially when outside the anechoic chamber. Further to this concern, is the requirement for a drone surveillance system to handle the appearance of a drone swarm, where the same spectral features will be observed but in a much greater quantity and a less predictable manner. Radar studies considering the detection and characterisation of drone swarms [112]–[114] are emerging, and is becoming an

area of significant interest.

For a single drone with a single motor, a measurement of Ω however is possible, which can certainly be used to assist in intra-drone classification if smartly combined with tracker information and drone prior. This phenomena will be revisited for the L-Band staring radar and compared with simulation in Chapters 4 and 6. This project will use the L-Band staring radar data and simulation together to model the frequency of a drone's returns in pursuit of extracting as much useful information out of the HERM lines as possible in the non-ideal case of low SNR, which is essential for a long ranged operation. An important side of drone target detection that this project will also consider are bird radar returns, so that the main challenge of distinguishing birds and drones is an attack on two fronts. Birds have moving wings, which do indeed impart some μDoppler modulations into the signal. Studies on their SNR and RCS fluctuations [115], [116] and μDoppler with FMCW [79], [94], [115], [117], have been done. Section 4.2.2 will present a range of bird signatures that have been collected over the course of this project. The signal processing choices used in the creation of spectrograms for drones in this work allowed excellent clarity when observing bird flocking, a sub-set of which was also presented in [5].

Fixed wing drones' can have have micro-Doppler patterns, and trajectories, that are considerably different to that of standard rotary winged drones. The detection of these targets is of interest for security, but was chosen not to be considered for the experiments of this thesis, as their flight in a densely urban area is more significantly more hazardous than for a rural environment. Many types of fixed winged drones have been shown to produce rotary winged μDoppler so in this case their classification is slightly more straightforward. In [76] using an L-band staring radar system showed a large drone (DJI Inspire 2) producing strong μDoppler, a small drone (DJI Mavic Air 2) producing no μDoppler, and a fixed-wing drone producing weaker but detectable μDoppler.[1]. It has otherwise been observed in [104] at X-band and [80] at Ku-band, that fixed winged drones contain a rotatory propeller as a source of thrust on the platform may exhibit a detectable measure of μDoppler that will depend on the drones pitch and yaw angle and

[1]This independent work used a different radar system to that used in this work described in Sec. 2.3 following, but the DJI Inspire 2 was the same *Inspire* as in this work introduced in Section 4.1.

self-shielding of the rotor from the radar by its body. However, the *RoCrow* fixed wing drone used later in Section 4.2 that possessed a very small and obscured blade showed little-to-no detectable micro-Doppler, similar to smaller quadcopters.

With these considerations, the reliable classification of drones based off distinctive micro-Doppler returns is itself dependant on sufficient radiation reflecting off these smaller, moving components of these platforms. This allows higher-frequency radar systems to detect highly detailed spectral features of smaller drones and birds. Lower frequency systems will have the advantage to perform detection over very long ranges, while higher frequency systems can possibly perform better beamforming to enable better classification at the extent of its operational range. The later work of this thesis will consider the classification of drone targets with and without μDoppler, via the use of drones with small and large blades compared to the operational wavelength. The following section will present the radars used for the experimentation throughout the remainder of this thesis.

2.3 Radar Employed in this Project

This PhD project uses two commercial off-the-shelf radar systems that were engineered for and marketed towards protecting airports from drone incursions. The installation of the first radar was installed on Gisbert-Kapp roof on 22^{nd} November 2020 - 7 weeks after the start of the project - and a second was installed on European Research Institute the 7^{th} of November 2021. The data collected by these radar systems formed the datasets from which research into data driven classification approaches was based. The operating principle, some details and example target data will be presented as applicable to the remainder of the thesis.

2.3.1 Staring Radar Principle

The Staring radar is a powerful, emerging radar trend. This radar also known as holographic, omnidirectional, ubiquitous or single-input multiple-output radar, looks everywhere in a search space continuously. As there is no scan of any kind, this method offers many immediate advantages to surveillance applications over narrow beam approaches - such as unbroken line of sight measurement and continuous integration of returns. The concept as applied to pulsed monostatic radar is as follows; a transmit antenna with a small total amount of individual radiators transmits a broad null-to-null beamwidth pulse over a field of regard. This interacts with the environment and some energy will scatter back towards the antenna. The receive antenna has more radiators than the transmit, so the beamwidth of the receive mode antenna is less. Figure 2.8 presents a pictorial representation. The 2D receive array digitises the I/Q data for each element, and all beamforming is done post receive and provides a means for simultaneously generating multiple receive beams. Thus the multiple receive channels coupled with a high PRF results in this kind of radar generating a large amount of data.

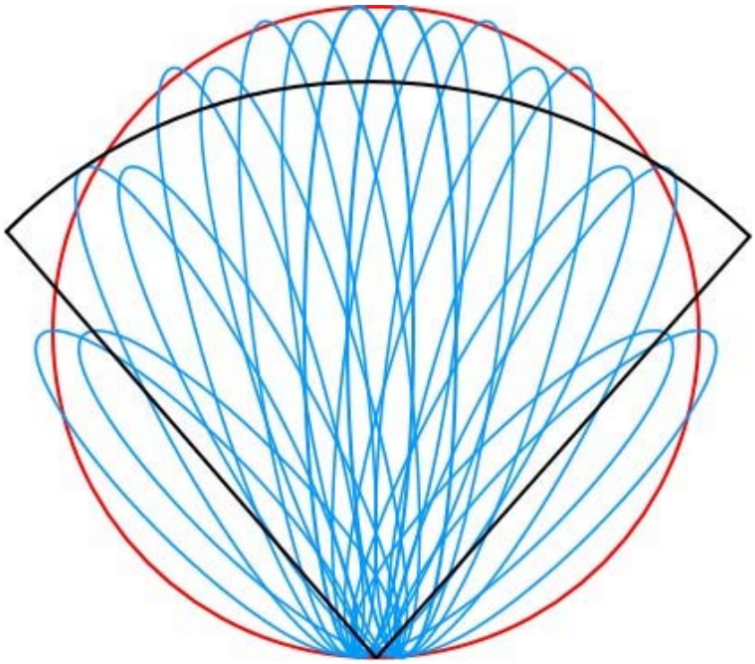


Figure 2.8: The pictorial principle of staring radar. One wide transmit beam (red) digitally cut off at the edges (black), and many narrow receive beams 'created' in post processing (blue).

The radar manufacturer provides a real-time processor that is able to detect targets, locate these in 3-D and produce tracks. It possess also a target classifier that is based on the measured trajectory and Doppler profile, and uses a Bayesian classifier to classify targets amongst eight classes. Generally for this work, the radar's classification output was ignored due to the algorithm being covered under the manufactures IP. Operators are able to extract files containing the location and time of the targets - however validating the tracks without a physical observation of the target means that further steps are required for using this gained information for research. Section 4.3 will describe filtering, verification and labelling of collected tracks that were used in later research. The L-band staring radar used also held the ability to record the raw staring radar I/Q data to be used in offline processing. The user interface allows the recording post-matched filter, base-band I/Q data to disk. A large repository of collected drone and bird data was created, and efforts to organise and process the raw data for extracting relevant or interesting target signatures was done as part of the research process and is fully described in Chapter 4.



(a) Radar#1 (European Research Institute)



(b) Radar#2 (Gisbert-Kapp)

Figure 2.9: The Gamekeeper Staring Radars as installed on campus roofs.

Table 2.1: List of specifications for the L-band staring radar

Specification	Value
Transmit Array	8 row × 1 columns
Receive Array	16 row × 4 columns
Azimuth Range	90^o
Elevation Range	30^o
Receive Azimuth Beam Peak-to-null	30^o
Receive Elevation Beam Peak-to-null	7.2^o
Operating Frequency	L-band
PRF	~7 MHz
Update Rate	0.28 s
Range Gate Separation	76.75 m
Operating Range	5 km
Transmit Power	1 kW

The L-Band staring radar installed at UoB is designed for the accurate tracking and classification of small targets, especially slow moving low altitude objects. This subsection will describe how its design parameters give it the best chance for its purpose. The radars as installed on the UoB campus are pictured in Figure 2.9 and a table of its specifications are in Table 2.1. L-band is a general good choice for a surveillance radar, as it suffers little atmospheric attenuation and it is highly robust against strong precipitation. It is also suitable for slow and fast manoeuvring targets as the constant interrogation allowed by the staring principle with fast update rate allows long CPIs of many pulses over short duration's of time to achieve target measurements with fine Doppler and temporal resolution. L-band also being low frequency allows high-power transmitters to be used, altogether ensuring targets can be measured with high SNR to enable

robust detection and classification. The operational settings in Table 2.1 were set by the manufacturer, optimised for drone detection in real operation settings, and were not changed for the collection of urban data for this work, aside the maximum operating range in some experiments.

2.3.2 Spectrogram Formation

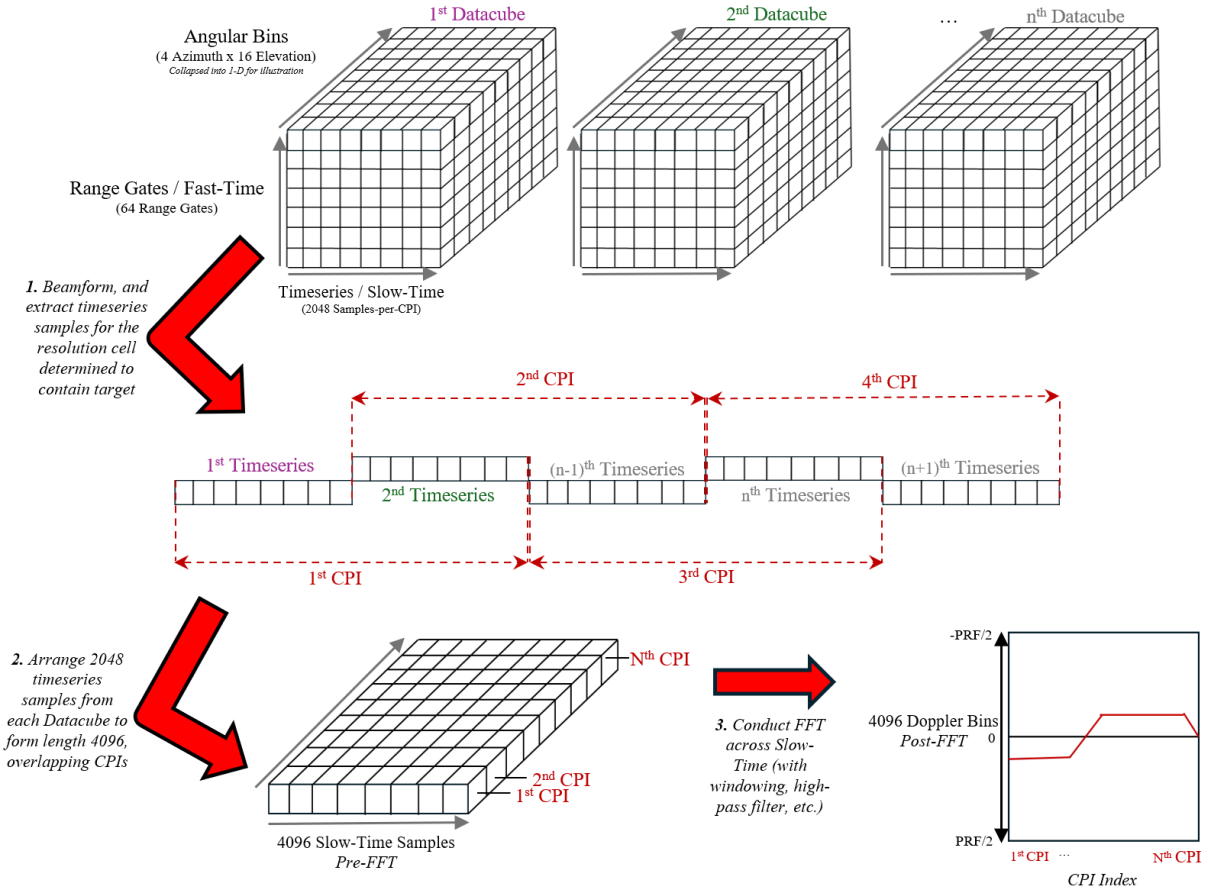


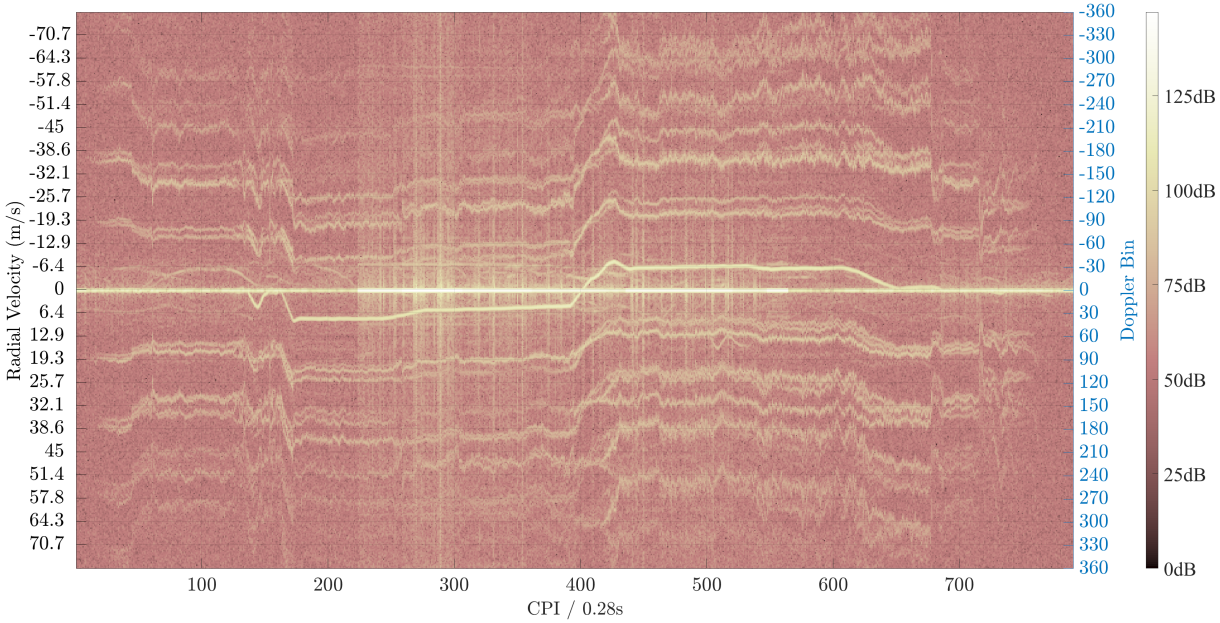
Figure 2.10: Illustration of spectrogram construction from a series of datacubes organised per CPI.

Figure 2.10 illustrates how a time-Doppler spectrograms are constructed from a series of recorded CPIs for all receiver channels and range gates for all of the original spectograms of this work. The optional parameters for spectogram formation were selected by inspecting their effect on the appearance of targets across a sample of target types and local clutter, and was fixed for all spectrograms used in the remainder of the thesis. The choices were similar to other literature

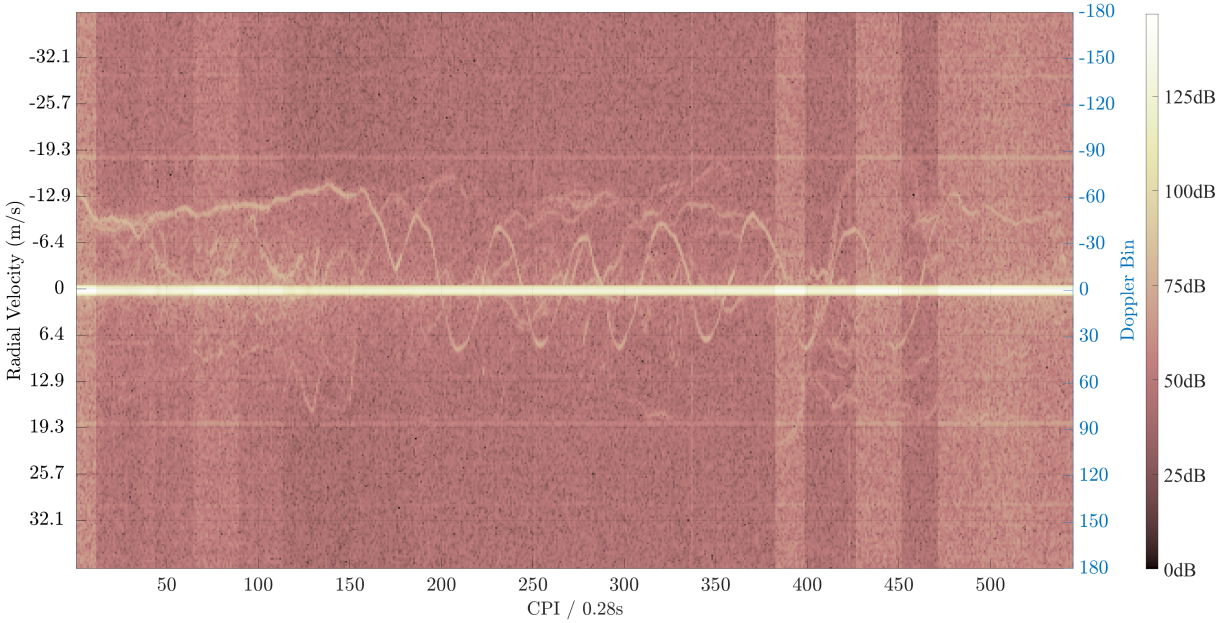
using drone returns [80], [88], [110], [118]–[129], and so presented findings can be used with relevant comparison to these other works. A FFT length of 4096 pulses was used to provide a large gain in SNR of returns out to kilometre order range, that with an effective update rate of 0.56s will still capture dynamic motion of targets. The update rate of the spectrogram was doubled by using a scheme of 50% overlapping CPIs, the effect of which was investigated [123]. The staring radar recorded 4D datacubes in range, azimuth and resolution with 2048 timeseries samples, from which the overlapping 4096-sample CPIs were generated in post processing. A Blackman-Harris window was found to reduce sidelobes very well and was used throughout. The level of clutter returns collected by the staring radar from the urban environment required the additional use of a high pass filter when considering the input of data into machine learning classifiers. As the level of urban infrastructure varied across locations in the radar field of view, the magnitude of returns in the central few Doppler bins compared to the noise floor varied largely, from 50dB up to 130dB. Therefore, for machine learning where the spectrogram need to be rebound to a fixed interval, such as between 0 and 1, there is a risk that fluctuations in target signature contrast would adversely affects learning. A high-pass filter of 15 Hz cut-off, removed returns with speeds less than $1.65ms^{-1}$ effecting the central 18 Doppler bins.

Figure 2.11 shows spectrograms of a drone and bird recorded by UoB's L-Band staring radar. These spectrograms shown were produced using the approach described, but without the high-pass filter. The drone (Fig. 2.11a) has a strong main body return with a series of sidebands decreasing in strength on either side. The bird spectrogram (Fig. 2.11b) only features the main body. There are no repeated Doppler sidebands as with the drone spectra, however, there are multiple Doppler peaks which can be attributed to multiple targets (likely birds) being within the same resolution cell. The stark difference in returns of these two target types gives a specifically designed classifier a great chance towards detecting rotary wing drones and avoid confusion with birds. However, as the physical size of a drone's blade is much less than the main chassis, so is the RCS of each component and so a drone's μDoppler sidebands are always, and sometimes a great deal, of lower SNR and so there is a range limit on the sideband detection. A classifier over-reliant on the sidebands will be susceptible to a lower probability of detection of drones

at long range, and cases where a malicious user has purposefully obscured vision of the drone blades. The specific μDoppler characteristics of drones' rotor returns was discussed in Section 2.2.3 and, later a brief discussion and results of proposed classification approaches towards a reliable drone detection without the presence of μDoppler sidebands will be presented in Section 5.2.4.



(a) Drone Spectrogram from DJI Inspire 2 using Radar#1.



(b) Unknown Bird Spectrogram

Figure 2.11: A drone and bird as acquired by the UoB L-Band staring Radars.

2.4 Chapter Summary

This chapter has offered a review of the key principles of modern radars that enable them to perform detection of moving targets, while capturing information of their range, radial velocity and angular. The maturity of radar technology enable quality detection capabilities that are fairly cost efficient. Additionally, within the spectra collected by a radar, there exists returns, or features, specific to targets that may provide some semantic assistance to target classification, such as drones possessing μDoppler. Capitalising on the details of returns captured by the radar to allow a robust classification is a strong trend of modern radar research. It is imperative however, that any classification algorithm deployed is able to provide a tempered and well-measured semantic justification for its classification, so that it is not easily fooled by strange or lesser-seen targets. It was shown in this chapter that returns may only be detectable by a radar if there is sufficient SNR to enable algorithmic detection. As there is a great variety to drone and bird targets that may be seen by the radar sensor at any time, a classifier employed must additionally be robust to a wide variety of returns within the selected target classes.

This will be the focus of the following chapter. The impulse of data science has been touted to hold great potential for radar systems. Machine and deep learning algorithms learn to produce useful outputs for operators for a given dataset or sensor, by allowing the algorithm to learn from a dataset, and use found features to enable classification of unseen targets. There is simultaneously great promise for these algorithms, but also some reservation to how they can be robustly trained, and verified. Performing deep learning of radar data for bird and drone distinguishment, as the main topic of this thesis, will now begin in the following chapter with an extensive review of machine and deep learning at large and within radar. This review will demonstrate the potential of deep learning within radar, and justify the experimentation of the later work of the thesis. It will be seen that a massive scope of algorithms and approaches exist within machine and deep learning. This work in the later chapters aim to provide some novel suggestions to improve the state of classification of drone targets, but we begin with a review of existing principles and recent work in the domain.

Chapter 3

Machine and Deep Learning Essentials

As the world is going more digital in every regard as we progress into the 21^{st} century, the amount of data generated in many regards of modern life is now vast - too vast for human to glance at and draw any insights or conclusions. Organisations and projects now employ data-scientists to turn massive quantities of images, audio and many other modalities of raw data into useful plots, views or models to assess the characteristics of the data, which allows for informed decisions to be made. Frameworks for performing classification of a new data point in context of a large amount of already labelled data, are rich and are growing in maturity and variety. To apply a machine or deep learning approach to a new area of research requires a combination of domain knowledge, and most critically, a large amount of (labelled) training data. It is up to the developer to use the correct tools well to achieve a robust real world classification that can be trustfully relied on to label unseen data.

This chapter will lay the essential machine and deep learning concepts used in the rest of the work, and is divided into three sections. Section 3.1 introduces the basic concepts of classifier training and evaluation, along with a selection of standard approaches for classification under the umbrella term of *machine learning*. Afterwards, Section 3.2 will review the modern standard of *deep learning* for classification applications, with many of the extensions described. Finally in

Section 3.3, we will highlight where the techniques reviewed in Sec. 3.1 and 3.2 have been used and published for radar classification applications, within the bird-drone challenge and beyond in other radar applications.

3.1 Review of Machine Learning Techniques

Machine learning refers to a set of algorithms that learns from input data to enable a desired processing, such as for classification. In a simple machine learning system, a human can be in the loop to highlight to the algorithm important features that are known to be particularly important in the domain, such as μDoppler in a Drone-Bird radar classifier. This current section will list some of the most common machine learning techniques that can be used to make, or assist, a classification attempt.

3.1.1 Evaluating Classifiers

A classifier predicts the semantic label of data input, x. A simple mathematical expression for the operation of a classifier is

$$p = \text{classifier}(x) \tag{3.1}$$

where p is the semantic classifier *prediction*. During training (†), a classifier factors in what the true semantic label, t, is as it takes the data input,

$$p = \text{classifier}^\dagger(x, t). \tag{3.2}$$

With this nomenclature in place, we can define the figures of merit that can be used to assess the efficacy of a classifier. A perfect classifier will choose a label that matches the true class every time. A random classifier would be expected to predict correctly at a rate of $\frac{1}{\#Classes}$. For a

fixed evaluation dataset, the most direct way to assess its performance is to observe the classifier accuracy. Accuracy is mathematically defined as

$$A = \frac{1}{N} \sum_{n=1}^{N} \delta_{tp}^{n} \tag{3.3}$$

where A is the accuracy value (usually reported in %), N is the total number of samples in the evaluation set, and δ is the Kronecker delta (δ_{ij}) which is defined as

$$\delta_{ij} = \begin{cases} 1, & \text{for } i = j \\ 0, & \text{for } i \neq j \end{cases} \tag{3.4}$$

where for the case in equation 3.3, for each n in the summation over the evaluation set the classifier can be either correct ($p = t$), or not ($p \neq t$). Accuracy however is not the most suitable metric if the number of samples for each class are significantly imbalanced in the evaluation set. For imbalanced test sets, other metrics such as the F1-score or more general F*-score factors will be useful [130]. This study will use accuracy as the main metric as balanced classes in all test sets was ensured. Having enough data for balanced test and train classes is not easy to achieve in multi-class problems with large numbers of classes, and particularly if a rare occurrence is an important detection target to the operator. Existing literature of drone and bird classification with various configurations of class numbers and target compositions have been reported, of which Sec. 5.1.1 will be dedicated. This project's essential aim is to differentiate detected radar signatures as containing drone presence or not, and so treats the classification challenge as a two class problem: Drone vs Bird, or Drone vs Non-Drone. Another approach taken is to also use an unknown class [94] that may represent false detections. The datasets constructed later in Chapter 4 attempted to ensure target presence at all times in the collected data to maximise the differentiation of targets across a diverse and varied real dataset.

Accuracy as a metric does not consider how the specific classes are treated by the classifier. For

instance, a bird-drone classifier at 90% accuracy could be caused by both classes performing with 90% accuracy, or all drones classified correctly with bird accuracy at 80% (assuming perfectly balanced test classes), which in this case has a significant impact on the operational effectiveness of the classifier giving no birds false alarms but a 20% rate of false drones classifications. This work will employ recall per class to assess how a classifier performs for that specific class. The recall is the rate of true predictions of a class to total predictions of that class. A 90% drone recall means 90% of predicted drones were truthfully drones, which provides a measure of how trustworthy a classifier is for a specific class. Classifier balance refers to difference of misclassifications across the labels, which ideally will have equal probability of false alarm for each class, but may possibly be biased towards a specific class selecting it more often than others. A confusion matrix allows the performance of a classifier to be displayed for a single classifier and fixed dataset. A schematic of a confusion matrix is shown below in Figure 3.1 that explains each of the sectors and offshoot metrics.

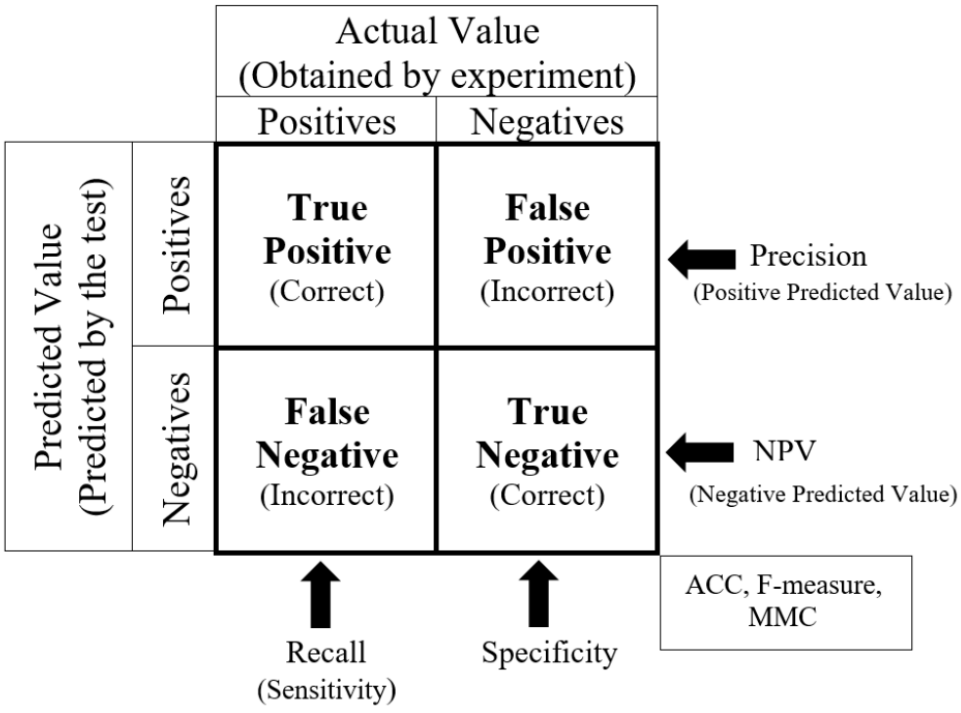


Figure 3.1: A confusion matrix, from [131].

For individual classifiers, it is very useful to observe the confusion matrix. For displaying an

assessment of an ensemble of classifiers, metrics derived from the confusion matrix can be plotted as against another investigative parameter which will be the primary visualisation later in this work. For significant results, the best performing confusion matrix will also be shown.

3.1.2 k-Nearest Neighbours

A common and conceptually simple classifier that does allow some sense of feature dependency is k-Nearest Neighbours (kNN) classification. A two dimensional data set, such as (for radar) target height and radar cross section, σ_{RCS}, could be plotted on an axis with the points labelled as for example cars and aircraft. Entering a new data point onto a graph and voting the classes of the k (a user defined hyper-parameter) neighbouring points that are 'nearest' to it allows a classification results that will scale well with more data. Figure 3.2 illustrates this principle.

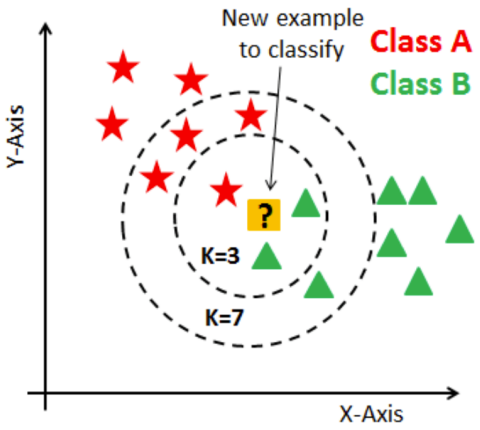


Figure 3.2: A 2-dimensional graphical example of kNN.

There are many options of how to define nearest, with the most intuitive being the Euclidean distance of the two axis given their units [132]. Of course the principle here extends seamlessly beyond 2D, and a high dimensionality of inputs will allow for a higher ceiling of potential for an accurate classifier [133], [134] if classes are balanced. In highly non-linear data, kNN can still be used if a more creative measure of 'nearest' is used or the data is further preprocessed with a dimensionality reduction or feature based representation. A similar classifier to this is a Support Vector Machine (SVM) [135], [136], which draws a hyperplane that separates the labelled data in a multidimensional space. The level of curvature of this hyperplane is user defined, which

if too low, will not separate non-linear data well, and if too high, will lead to overfitting (see Section 3.2.4).

3.1.3 Dimensionality Reduction

If a classification is attempted on a high dimensionality data point, such as the pixel values of a 2D Doppler spectrum, it can be difficult to get a robust set of features for classification due to the sparsity of relevant data in the computation space. Data as it appears in the real world is highly dimensional, so it is an efficient practice to compress or reduce its dimensionality while preserving important features[137]. In simple data, a dimensionality reduction could reveal enough separation of classes to perform a classification in a supervised learning case. More likely, is that a simple or highly complex classifier will use the compressed data to make a final decision. Two common examples are the unsupervised Principle Component Analysis (PCA) and supervised Linear Discriminant Analysis (LDA). PCA finds dimensions in the data that offer little variance to the others in the data, and removes them, until a number of target dimensions is reached [138]. LDA finds linear combinations or transformations of input data dimensions to a lower number that maximises (minimises) a measure of the inter (intra)-class variances [139].

For a large, high-dimensional variable sets where linear combinations of values do not efficiently solve for a global minimum, more recent non-linear approaches such as t-Distributed Stochastic Neighbour Embedding (tSNE) [140]–[142], or Uniform Manifold Approximation and Projection (UMAP) [143]–[145], can be used as a single tool to decompose an entire dataset to a visualisable number of dimensions. This allows for an analysis of a sample with respect to all the other samples in the dataset, which provided the initial dataset that encompasses the diversity of the operational environment and could be used to perform sub-category estimations and provide a measure of risk of classification. They are advantageous to neural network based methods due to their very fast training time with comparatively fewer hyperparameters, however their post-training runtime is greater and deployment applications are fewer. UMAP with pre-processing steps has been used to categorise weather radar data, showing performance advantages over

PCA [146]. tSNE has been used in a selection of radar studies [147]–[152]. Sec. 8.1.2 and 8.2.2 will contain further expansion and show decompositions of radar data with these mentioned techniques.

3.1.4 Other common methods

Is is possible to use the above methods with tracker, RCS and extracted μDoppler information, and plausible for the classification based on raw data with enough prepossessing. An often used candidate for a classification using the human-interpretable physical features of a target based of tracker information is a Decision Tree (DT). A human crafted DT could be made for a simple classification challenge such as for differentiating ground and air vehicles, via questions about the height above ground and velocity of the targets. A machine learned DT asks simple questions to the training data recursively, adjusting the questions to have maximum differentiating impact at each node through the tracking of Gini Index and Information Gain [153], [154]. The advantage of DTs is their superior interpretability. A misclassification can be tracked through the decision system, a potential source of error found, and a subsequent improvement or adjustment can be made to the processing or dataset. An extension of DTs that are less interpretable but have a much greater performance on large, high dimensional data are Random Forests, in which a large amount of DTs are each trained on a small, different random subset of the training data, called bootstrapping. Each tree in the forest is uncorrelated with the others, so a voting of the final results to produce a classification is going to be more robust to data that have a single feature outlier or two, which is an example of ensemble learning [155], [156]. A Support Vector Machine (SVM) [157] draws a dividing line, plane or hyperplane that best separates classes as they are found in an embedded space. They optimise an algorithm to maximise the distance from the nearest points to the boundary - thus requiring points to be well separated in the first instance. These can be hard boundaries with no outlier points (hard-margin SVM), or with a tolerance for misclassifications but producing a smoother, less over-fitted boundary (soft-margin SVM), with many further extensions that suit it to a particular application [158]. Demonstrations of radar target classification with this approach have been conducted in other radar domains [159].

In a field where there is a lot of domain expertise, the previously listed classification methods can be applied to human-monitored, *Hand-Crafted*, physical features extracted by an operator who is knowledgeable of their data. This is only possible where physical phenomena in the real world is mapped intuitively to the sensor output. To illustrate, some hand-written digits could be classified using physical features - for instance, 8, would be identified as being the only number with two closed topological loops which is an unambiguous and true claim for 8's in general. However, with the addition of salt and pepper noise suddenly the two loops may not exist mathematically, and the feature although intuitive will lose its effectiveness. Alternatively, if in a rush an 8 may be written with either loop not intact which will flaunt the effectiveness of the feature created. Effective extraction of physical features for classification requires that (a) the domain will not likely conjure physical manifestations that are unexpected by the operator, and (b) that the feature extracted depreciates sensibly with increased noise. Guaranteeing both these requirements is not simple and requires a large amount of data for testing to ensure robustness. Conversely, machine (and deep) learning provides a method of classification that is based only on the training data. Hand crafted features are used in radar applications where known scattering phenomena allows targeted extracting of information, but their ability to provide reliable information to a classifier will be suspect at the edge of the radar's operating specifications. In the radar detection of bird and drone targets, many handcrafted features could be used to assist classification, such as the observation of the symmetrical peaks in the Doppler spectrum has been shown in [160], [161], as well as detection of drones carrying payloads detection [162] and measurement of drone motor speeds [6] that this work part-took in. These methods acting upon physical features generally allow for more interpretable processing than using a machine or deep learning approach. However, for dynamic operational environments and challenging targets, deep learning holds better potential for maintaining its performance with a wider variety of spectra from unseen scenarios. This thesis mainly concerns deep learning approaches to further explore their potential to help enable reliable classification of drones and birds. Section 3.2 following will review deep learning approaches as well as used across radar ,

3.2 Review of Deep Learning Techniques

Deep learning, is a contemporary sub-set of machine learning where an algorithm learns from a digitised dataset and uses the gathered information to make predictions of other unseen samples. The difference is how the information is processed and how the learning is reflected in the model. Deep learning is a neural network structure with one or more hidden layers of perceptrons, that learns from a dataset recursively by updating perceptron parameters via gradient decent of a loss function measuring how wrong the model was at that stage. Unlike the approaches in the previous section that all were unique in their implementation, a deep learning model describes a general architecture that has a very large inner dimensionality and contains standard building blocks. However, the construction of any specific deep learning classifier may be totally customised. This section will at first introduce and motivate the use of common Neural Networks for classification in Sec. 3.2.1. Sec. 3.2.2 will describe the Convolutional Neural Network (CNN) used exclusively in Chapters 5 and 7. Sec.3.2.3 will show other common NN archetypes that have been used in radar literature. Sec. 3.2.4 will describe the process to optimise the training of a deep learning classifier. Finally, Sec. 3.2.5 will briefly introduce the Convolutional Autoencoder (CAE) which is another common neural network capable of performing unsupervised learning, to which Chapter 8 is dedicated.

3.2.1 Introduction to Neural Networks and Deep Learning

A monumental paradigm shift occurred in the classification and sequence prediction fields with the popularity increase of artificial Neural Networks (NNs). With a large amount of labelled training data alone, it is possible to get good performance in highly non-linear classification tasks [163], without any prior knowledge of the contents of the data, only the nature i.e. timeseries, image, etc. which dictates the configuration of the model. In radar, there is a mapping of the search environment to the digitised radar signal, so drone features such as the μDoppler sidebands have a direct relation to the classifier objective. But as classifiers seek to become more accurate in challenging environments, a robustness to non-linear features in the radar

returns is required, especially in the pursuit to decrease the probability of false detections.

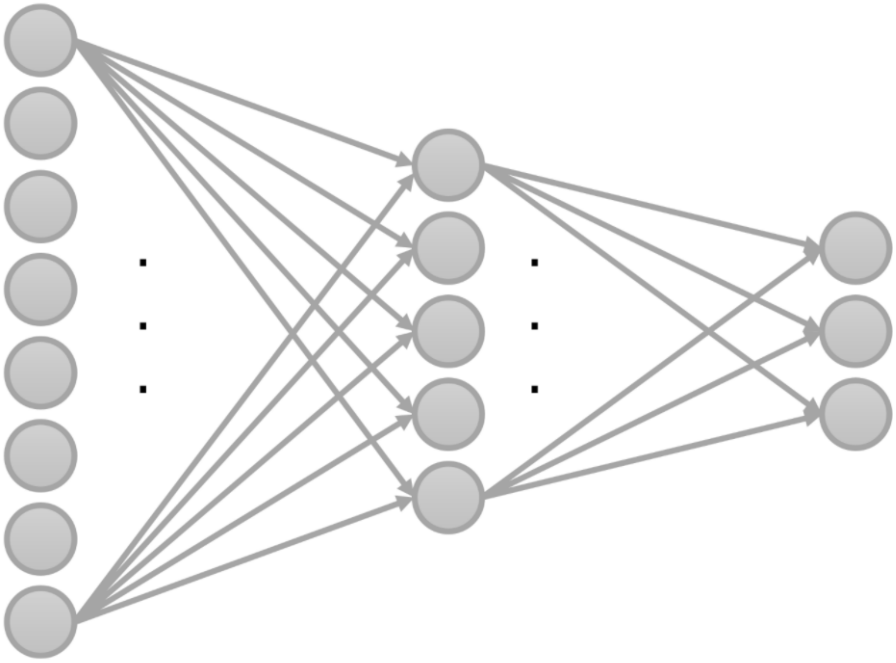


Figure 3.3: Node schematic of a Neural Network.

A NN defines a space of a huge dimensionality that could represent any combination of functions and expressions using an input selection of numerical data. Figure 3.3 is a schematic for a elementary NN classifier. The number of nodes that appear in each layer, and indeed the number of layers, are hyperparameters. This neural network has 8 inputs, and 3 outputs. This means an 8-Dimensional *numerical* vector is fed into the NN, which is the sample to be classified. 3 output layers would correspond to a 3-class problem. The output of the neural network classifier is also a number, and each of the 3 outputs will contain a number that represents the confidence of the prediction. For convenience, the output is normalised that the sum of the outputs is one, so it may interpreted as a probability distribution. A *forward-pass* of the network is to pass a sample point through the input layer and to observe the classification result, which is the largest value of the output layer.

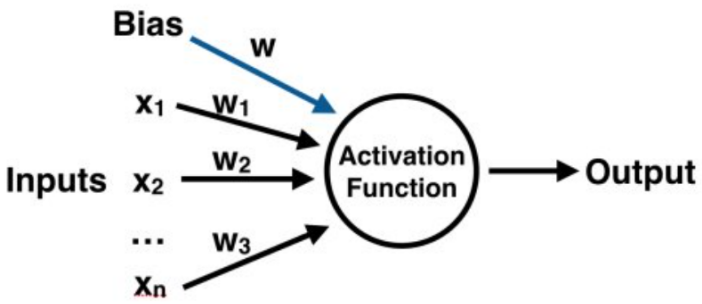


Figure 3.4: The inputs and output of a generic perceptron.

The parameters, are the values that make up the neural network. The mathematical structure of each node in the neural network follows the format in Figure 3.4, and are called perceptrons. Each node in the network has as bias, and each connection in the network has a weight. The weights and the biases are the learnable parameters of a neural network. A neuron will take as many inputs as the size of the previous layer, and output one single value to all the inputs in the next layer. The output of perception is defined as follows,

$$x_{l+1} = \Gamma(\, \mathbf{x}_l \cdot \mathbf{w}_l + b \,) \tag{3.5}$$

where x_{l+1} is the output of a perceptron, $\mathbf{x}_l$ is the vector input from all the perceptron outputs of the previous layer, $\mathbf{w}_l$ are the weights of all the connections to the previous layer, and b is the scalar bias of the neuron. These components are vector summed, and the result is input into a differentiable *non-linearity* or *activation* function, Γ, such as a sigmoid or *tanh* function which re-maps the real number line to between 0 and 1, or an alternative such as the Rectified Linear Unit (ReLU) and its many versions. An untrained artificial neural network, is a large ensemble of these perceptions (mathematically) arranged as in Figure 3.3. A loss function is designed so that if the classification is very wrong, there is a larger cost associated with the forward-pass.

With a cost function defined that scales with the severity of the error, the backpropagation algorithm is the final piece to training the network to be a classifier [164]. The loss function is ultimately a function of the weights, biases, and the input layer values. Theoretically, if this

function could be differentiated, the global minima of the loss function would the absolute best possible classifier performance of the NN with the hyperparameters it has for that given dataset. Practically however, the number of variables is so large due to the fully connected layers that this differentiation is not possible. After a forward-pass, a backpropagation algorithm [165], such as Stochastic Gradient Decent with Momentum (SGDm) or Adaptive Moment Estimation (AdaM) estimates the gradient of the loss function with respect to the model parameters. Their action is to give a slight perturbation to all the weights and biases in the each layer of the network moving backward, and the effect on the loss function is calculated. The effect of adding a small value to a weight, could have a positive or negative effect on the loss at the output, and a small increment is added or taken from to the weights and biases of the network such that the loss is minimised. If perturbing a weight or bias changes the value of the cost more than others, then the weight or bias will be adjusted to that neuron more, so that the calculated loss is reduced as fast as possible. This process is repeated for all the training data, until the weights and biases of the network become fully adjusted so that the classification of a forward-pass is totally accurate. To speed up training, and also to prevent the network cost getting stuck in a local minimum, NN training will usually feature stochastic gradient decent, which is the term for using sub-*batches* of training data at a time, and the adjustment to the NNs parameters will only be applied after averaging over the batch. There are a variety of cost functions and optimising algorithms that are used in NNs, and there exists many libraries and extensions for most high level programming languages that have the various costs, optimisers and computational layers available for a convenient use. This project uses MatLab 2021a-2023b and PyTorch 3.7+ [166] for later original investigations of deep learning with bird and drone radar data.

3.2.2 Convolutional Neural Networks

There are a few extensions of the basic NN that are especially good at solving a specific task, and Sec. 3.3 will show where notable works have used deep learning for a radar classification tasks. For classification of 2D pixel images, it is possible to flatten the image to fit a 1D input

layer, and this will provide a very respectable classification score[1]. However, much better is to use *convolutional layers* [167], [168] which are backpropagation-trainable square or rectangular numerical filters that are slid over the pixel image and output many intermediate images that are smaller than the input image. Each of the pixels in the filter are learnable parameters, and so they are used to extract extremely general features via backpropagation directly from the dataset. At their output, a normalisation of features usually occurs to keep the gradient from exploding or vanishing, with typical options being layer [169], batch [170] or channel [171] normalisation. Frequently, *max-pooling* layers are used to select the largest value of a square region of the image to quickly shrink the dimensionality and scale of the problem, and also to make the model more robust to local variability [172]. Both the max-pooling and convolutional layers are defined by kernel size, stride, and perhaps some dilation or padding, but only the convolutional layers feature learnable parameters. After a series of convolutional layers, the final output is flattened and a fully-connected NN uses this as input to perform a non-linear classification upon the features received.

This model is known as a Convolutional Neural Network (CNN), and performs extremely well on images compared to flattening the input as it preserves local connectivity of the data, and produces high level features extracted from the input samples by encoding information about the dataset into the learnable convolutional parameters. In this work, we will only consider 2D inputs to the CNN using spectrogram images. However, in radar, the use of 1D convolutional layers applied to single spectra has and continues to be of research interest. It requires a lot of data and more careful optimisation to train a 2D CNN from a random initialisation. For radar applications where data is not as widely available as optical images for example, transfer learning from models extensively trained on optical images have been used in radar. This work will explore transfer learning in Chapters 5 and 7. Chapter 8 alternatively will use a radar-data exclusive pre-training approach for CNN classifiers using a relatively small radar bird and drone dataset based on unsupervised learning to be briefly introduced in Sec 3.2.5.

[1]This is performed with the collected bird and drone data using tSNE and UMAP in Sec. 8.2.2

3.2.3 Other Common Neural Networks

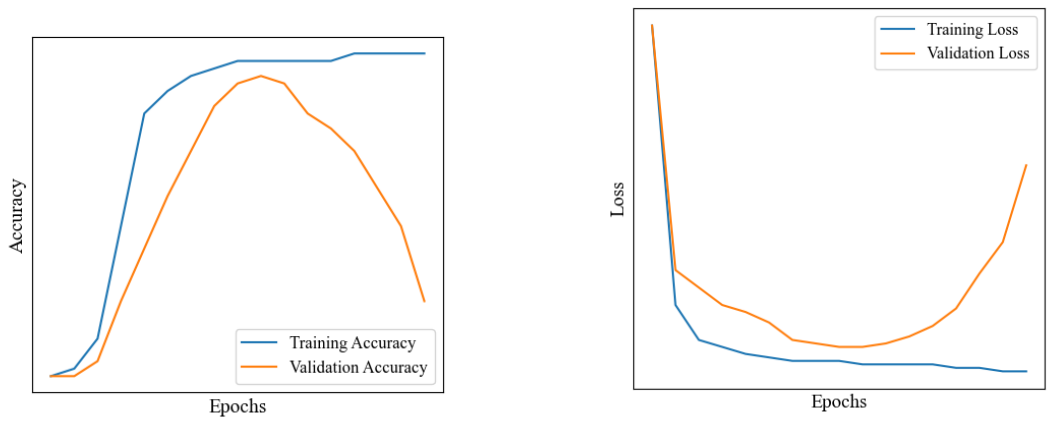
While the original work in this this thesis will be performed with CNNs, many other neural network architectures exist and have been used in radar-based classification studies. Residual networks [173] are deep networks that have connections that skip layers. This allows feature combinations at low and high level, which increases the complexity of the model without a massive computational burden [76]. Another common flavour of classifier is the Recurrent Neural Networks (RNN) that can be used both to classify and predict time series data, which uses an inner loop amongst the layers that feeds numeric information from previous time steps. These types of classifiers have been used in literature towards detecting drones, using FMCW radar on spectrogram images [168], [174], with Long-Short Term Memory (LSTM) [175] gates used that with an increased computational load has better long term memory. Very recent advances with attention mechanisms [176] has lead to transformer architectures [177], with large language models [178] such as ChatGPT (Generative Pre-trained Transformer) receiving huge attention [179]. Transformer and CNN architectures were compared in recent radar classification studies with birds and drones in [180] and [76] and will be more expansively reviewed later in Sec. 3.3.4. Another emerging type of neural networks are Spiking Neural Networks [181] that have a closer resemblance to biological brain operations, have been used in a small collection of radar studies such as person identification [182] or automotive applications [183].

The NNs in these works have a semantic label output that is a 1-D vector the length of the number of classes. Neural networks generally can be trained with any output and many levels of structure. The Autoencoder is a neural network that with backpropagation recreates its input, so learns from the data in an unsupervised manner without any semantic labelling [184], [185]. A detailed review of autoencoder architectures is covered in Sec. 3.2.5, and this particular approach for bird-drone classification will be reported in Chapter 8. NN based learning and input-output are modular, so fused networks can easily be constructed and trained either separately or end to end. Each module of a fused or hierarchical network is said to have learned its own representation that connects to each of the others, and may require only a small amount of training end-to-end

to fuse the representations for an application. Portions of networks or even layers when trained individually is referred to as greedy learning. Such models feature in radar literature, and is a huge area of interest to enable low bandwidth networking of multiple radar systems [186], and also across different sensors [187].

3.2.4 Optimising Neural Networks

Training of a neural network for the purposes of deployment in a real system requires an analysis of its performance on unseen data samples not included in the training dataset. The common practice is to divide the entire dataset into train, test and validation subsets, with no shared samples between each. While the classifier learns generic features about each class from the training set, a validation set is passed through the classifier and its performance, through measuring the loss or accuracy, is assessed. The performance of the training data for that epoch is also assessed, and they are compared. Figure 3.5 shows two views of overfitting that will be observed while training that also illustrate the importance of evaluation sub-sets. Fig. 3.5a shows generic curves that could represent loss or accuracy of a train and validation set. A longer training duration gives the classifier more opportunity to learn features from the train set. At some point, the extracted features will be so tuned to the specific samples in the training set that generalisability of features will no longer support sensible processing of unseen examples.

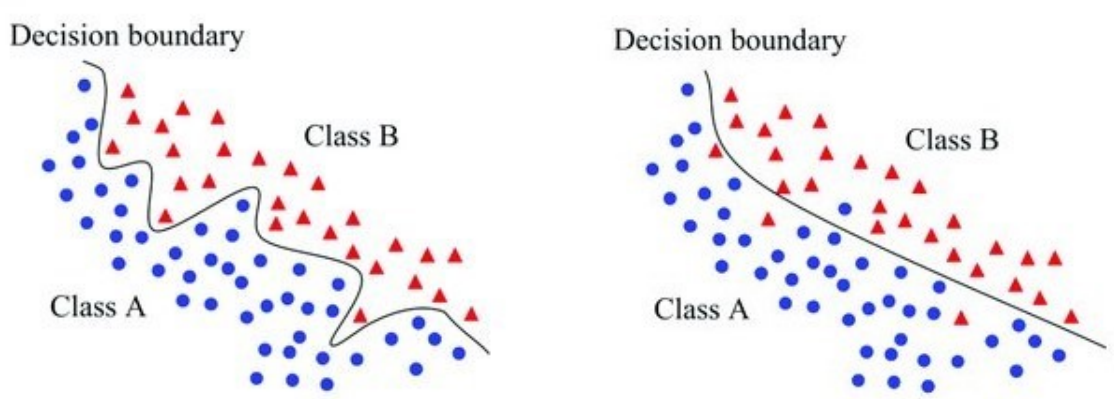


(a) Classifier accuracy with training duration

(b) Classifier loss with training duration

Figure 3.5: Representations of overfitting in classifier training.

Figure 3.6 illustrates a representation of overfitting in a SVM context of drawing a separation line between two classes. Creating a model with optimal balance between under and overfitting is a key task for machine learning engineers. The approach taken in this work was to use the maximum sum of train and validation accuracy scores, which is highly reproducible and ensures a well trained model that stops when as validation diminishes.



(a) Overfitting (b) Reasonable fit

Figure 3.6: Illustration of overfitting in SVM. From [108].

Once a model is determined to be appropriately trained, it will be evaluated on the test set that will again contain only unseen data. This is where the final metrics will be reported for benchmarks or experimental studies. The choice of set balance requires consideration of the availability and quality of data. 60:30:10 train-test-val set balance is a choice for massive collections of image data. As there is twice the training data as test data, the extent of unseen generalisability is tested at the possible expense of lack of variety within training data. For smaller datasets, 70:20:10 or similar has been used. The results of this work will begin in Section 5.2.2 with further details of the training approach taken.

In the construction of a neural network, there are both arbitrary functions and many scalar hyperparameters that may dictate specific global and local effects that optimise the performance and robustness of an operational classifier. Data scientists need to finely control these to optimise performance for specific details of the data challenge they are presented with. The domains of optimisation could be decided into three groups; training parameters, model structure and

data processing. In Appendix A: Table A.1 - A.3 a set of significant hyperparameters for these groups that feature in this work are shown, with some commentary on their effect, significance or role within a convolutional neural network. Table A.3 concerns digital signal processing and input choices as was discussed in Section 2.2.2 related to radar target classification. A challenge of deep learning is that there are a great number of hyperparameters to consider. Running a number of experiments in a grid search of parameters is one way to iterate towards an optimised classifier, but other strategies such as Bayesian optimisation conduct training experiments based on the distribution of previous results, which will be used later in Section 5.2.3. Without Bayesian optimisation, good neural network performance can be achieved - but is imperative to scan the learning rate to experimentally determine optimal performance. The other parameters define the model while the learning rate, optimiser choice and regularisation parameters enable learning with that model.

3.2.5 Unsupervised Learning with Autoencoder Neural Networks

An unsupervised CNN based approach is the Convolutional Autoencoder (CAE) [188], which is foundational or deeply analogous to many now standard optical image deep learning methods [189], [190]. CAE has seen an increase of use in recent work in fundamentally data-limited radar challenges such as human activity recognition [99], low-bandwidth communication [186] and autonomous navigation [191]. Autoencoders have been shown to be effective for pre-training deep neural networks before fine tuning towards downstream tasks such as classification in machine learning [192], and in a growing number of radar applications in studies with real and augmented data for human activity classification [99], [193], drone detection [194] and even drone-bird classification [152]. Supervised classification, such as that in the training of AlexNet, uses a loss function that captures the difference between current classifier state and true label. The CAE is a method of performing label-agnostic unsupervised learning. The loss function in this case measures the difference between the input and the output, thus forcing the model to learn how to recreate its own input. Autoencoders can be under- or over-complete. An AE is undercomplete if the hidden layer size is less than the input dimensionality, which upon training causes the network to learn, in the ideal case, a compressed representation of the data [195] – one that extracts key, pertinent features of the entire training set that may be utilised to interpret arbitrary, unseen data drawn from the same set. Chapter 8 has further details of its anatomy and implementation of a symmetric CAE with the collected bird and drone data to investigate its use for classification and downstream applications.

3.3 Review of Deep and Machine Learning in Radar

This section reviews the use of machine and deep learning in radar applications inspirational to or directly concerning the radar drone-bird challenge. Applications of AI towards radar are increasing greatly as data processing hardware continues to evolve. As a result, many of the fundamental radar data processing tasks, such as detection [196] and classification have been considered as a deep learning, computer vision problem [197]. Utilising processed tracks for classification is an area of substantial research, but is not the main the subject of this thesis. It is however a major interest for radar engineers as another layer of classification for further robustness, and will be overviewed first in Sec. 3.3.1. Then, Sec. 3.3.2 will review where hand-crafted feature extraction approaches on raw radar data has been shown sufficient for accurate classification, with a particular focus on birds and drones. Finally, Sec. 3.3.3 will review where automatic feature extraction approaches of neural networks on spectrograms across radar applications, and Sec. 3.3.4 will focus on spectrogram based drone and bird classification with machine and deep learning. Sec.3.4 will briefly summarise the takeaways from the chapter, and preview where and how later chapters be will expanding on the methods reviewed.

3.3.1 Classification Using Tracker Output

Tracking of targets is an essential for a surveillance radar. The challenge of tracking becomes increasingly difficult with the number of targets, variety of flight and closeness of targets compared to the length scales of sensor resolution. Much required development has seen a great number of Kalman filter [198] based track-from-detection algorithms [52], that may possess multi-target functionality [53], [199] within their design for many applications. Radar tracks of birds and drones have been demonstrated useful for feature extraction and classification with machine and deep learning. Simulated tracks of urban, civilian drones were reported with 99% accuracy using hand-crafted extracted trajectory features as input to a simple NN classifier in [200]. More recently, [201] conducted an analysis on both simulated and real targets, and [160] and [202] applied DT and RF to hand crafted features from real tracks and both track and

spectral features of a field trial of birds and drones with a floor of 85%. Deep learning has be demonstrated to support classification using handcrafted kinematic and morphology features from the tracker, with bistastic σ_{RCS} features, in [203] on birds and light aircraft with passive radar.

3.3.2 Handcrafted Feature Extraction of Spectrograms

Radar spectrograms of drones and birds can provide a basis for classification. Bespoke frequency domain algorithms that highlight micro-Doppler returns have been used for the identification of drones from confuser targets, such as for L-band spectrograms similar to this work via a symmetrical component extraction feature in [161], and log-harmonic summation borrowing a technique from audio processing in [87]. Various schemes have demonstrated the use of handcrafted features for drone classification, including feeding such features into a deep learning classifier which was shown to be an effective approach in [89].

The difficulty of this approach for surveillance radar is that features will be subject to significant target variations, such as motor speeds and rotor blade size, and environmental fluctuations such as detection range and reduced SNR, clutter variability, or interference/jamming. However for cases where the identification of the target is positive and SNR is sufficient, handcrafted approaches can reliably measure aspects of returns with a minor overhead. As a result of work shown in Sec. 4.1.3, hand crafted extraction of (four) individual motor speeds of 2 drones from time-Doppler spectrograms in [6] was demonstrated at a range of 1 km distance at L-band with 0.81% error, along with estimations of blade length which was less reliable with 29% error. Ref. [97] at W-band estimated blade length and rotation frequency at a range of 1.5 m with average errors of <2% on a fixed drone with different numbers of rotors without reporting on any variability of speeds within the motors. [128] performed payload mass estimation of flying drone targets using both hand-crafted and singlar value decomposition. features using diagonal-linear and Naïve Bayes classifiers.

3.3.3 Automatic Feature Extraction in Radar classification

The main study of this thesis is to analyse the prospect of target classification using modern AI techniques when applied to radar spectrograms. The rest of this subsection shares approaches of deep and machine learning classification as demonstrated with radar systems on non-bird and non-drone targets. A great range of radar targets possess classifiable features, and deep learning approaches, in particular the CNN has in the literature been shown to effectively be able to classify simulated and real target datasets. This subsection will review key work demonstrating deep learning as a suitable tool for target classification in both civilian and military contexts. SAR target recognition with CNNs has been widely studied, due to both the fact that substantial open datasets have existed for nearly 20 years [204], as their process returns produce a 2D image, and also due to their large scale implementation and importance for international security. Works include transfer learning from the optical domain [205] and without [206], [207] to more advanced processing utilising unsupervised [208], [209] or semi-supervised [210] learning, and also with explainability featuring as part of some implementations [211]–[213].

The automotive area has seen much investigation of deep learning driven by the commercial demand for radar as a cost-effect solution for tasks. Again, numerous open datasets [214]–[217] have been shared that allow researchers to develop and test algorithms at a rapid pace. Object detection in range-Doppler [218], [219] and segmentation of range-azimuth maps [183], [220] have shown to be viable in a variety of experimental conditions. A large area in this domain is the information fusion of multiple sensors - both many sensors around the vehicle, but also fusion of say camera and radar information - as detailed in the review [221] and demonstrated in other recent work [222], [223]. Deep learning challenges such as fusion and semantic segmentation require an excellent level of representation learning for a system to firstly work and then more importantly to be robust in a dynamic and safety critical operation such as that for cars operating on on real roads. The theme of representation learning will be explored at length in Section 8.1 where we will investigate how unsupervised learning for the drone and bird challenge could lead to better approaches for classification. The work in this thesis using the CAE will also

take inspiration from human monitoring applications that is another radar area that has seen much research and experimental demonstration of unsupervised deep learning and strongly shares the issue of lack of available training data. Work showcasing viability with real data have been done in the contexts of fall detection and activity classification [99], [224]–[230], vital-sign monitoring [231]–[233], person identification [182], [234]–[237], and sign language interpretation [103], [238]–[241].

3.3.4 Automatic Feature Extraction in Bird-Drone Radar classification

The final section of this chapter will now review key results of deep learning with bird-drone radar signatures. Section 5.1.1 will also review in detail a selection of experiments that are similar and inspirational to the studies conducted in this work of classification of bird and drone spectrograms. Before discussing classification in this section, deep learning has been applied to small target detection [194], [242] with drones as the test subject. Deep learning based detection like these, or alternatively detection-with-classification [243] are interesting prospects that may become more normalised in such surveillance systems. The differentiation of detected drone and bird targets is the subject of this work, and numerous publications have demonstrated the capability of various types of CNN classifiers to effectively perform this task in a great variety of target, radar and environmental conditions. Literature featuring birds and drones will be reviewed after the following based on drone targets. Bird radar data is considerably more challenging to collect as they are a non-cooperative target, but their consideration is critical.

Unlike research work of deep learning on SAR images where many publications are born from a single open dataset, open radar drone datasets are not as ubiquitous in the space, in part due to the range of returns across different radar bands and appearance with different sampling rates and digital signal processing options among other reasons [244], although this is beginning to change. Recently published in 2022, [121] is an X-band openly available dataset of 4849 spectrogram images of 6-class across 5 UAVs including a 'bionic bird' target. The authors using the pre-trained optical deep CNN classifier, VGG-19 [245] to achieve 97% overall classification

accuracy. [246] from May, 2021 describes an open-contribution dataset of an available to buy radar platform at W-band, that was initially released with 187,932 individual processed spectra values from vehicle, person, bicycle and UAV classes.

Demonstrations of CNNs with *simulated* drone returns (see also Chapter 6) were shown to classify multiplicities of blades and rotors from drone returns with sampling rates greater [92], and lower [247] than the motor speeds, both with simple mathematical point target simulations. Ref. [110] performs a Gaussian independent component analysis approach on an ideal, high fidelity X-band analytical Fourier simulation to estimate all tested 7 physical parameters to within $< 1.6\%$. [125] simulated 6 real similar drone models with RCS mesh Finite Element Analysis (FEA) simulations at X-band, but classified only 2 of the models reliably using an unoptimised SVM classifier. [129] performed a long window FFT to a detailed analytical simulation at X and W-band to produce HERM lines for measured parameters for 5 different drone models with a random sampling extent of variation, and successfully used a CNN to investigate classification performance of the models as a function of SNR without any retraining. [126] simulated the μDoppler of drones and a flapping wing with body geometric models and analytical simulations with X-band FMCW processing, with Doppler-time and CVDs to construct schemes of multi bird and drone combinations including simulated buildings via physical optics, and used a CNN to separate class with performance near or above 90% fairly consistently.

Classification of real drone targets has been done and will be listed in order of increasing target to receiver distance, which ranges from indoor bench conditions to ~km order classification. These studies do not feature birds, but do demonstrate the CNN to operate well on real data, able to learn features from minor differences to discriminate between drone types. [248] measured anaechoic monostatic RCS and Doppler data with a Ka-band radar and classified 9 similar drone models above 99% accuracy at various operating frequencies using a LSTM network with adaptive learning rate optimisation. [88] used drone data collected with a pulsed multistatic system of drones with and without payload, and the optically pretrained AlexNet classifer [249] was able to classify 6 different classes with spectrogram, cepstrogram and CVD images with 95.1%

accuracy using ~400 training samples. [124] used the same classifier on 6 real drone models on Ku-band FMCW data to achieve a 96% F1 score with targets divided by size, and 68% when attempting drone model classification on 6,600 radar spectra. [123] used an Ancortek X-band FMCW radar to collect 10,000 training samples of a Mavic Air 2 drone, two bird-mimicking fixed winged drones, and humans walking and sit walking, achieving accuracies between 79 and 84% with modified ResNet CNN architectures.

CNN classification of *real* bird and drone Doppler data has previously been done in many notable publications that are of great relevance to this work. Section 5.1.1 will further expand on these publications as the details of their implementations and results will be compared to the main research contribution of this thesis. University of St. Andrew's have two prototype FMCW radar systems at 25 and 94 GHz that in [94] were used to collect a total of 2,400 images of 0.4 second dwell Doppler-time spectrograms of 3 drones and 4 birds of prey. They used both a GoogLeNet optically pre-trained CNN and a smaller, simpler un-pretrained CNN, which necessitated creating RGB and mono-color images for the classifiers with 3 and 1 input channels respectively. Two-class bird and drone, and four class bird, drone, clutter and noise labelling was used, with the two class problem giving better bird and drone accuracies by 0.7% and 3% respectively which was on average in the high 90s. Elsewhere, [250] used a low cost SAAB W-band FMCW sensor to produce 74,000 Doppler spectra for 6 real drones and 2 species of opportune birds baited by food with 90% accuracy. Then, classification attempts of unseen drone models was done, dropping performance to 78% that was restored to 86% with the incorporation of simulated drone model returns. [251] showed that a CNN was challenged to observe unseen drone models in a bird-drone context, that was resolved with an extension of the training set with real data. The use of simulated data for assisting classification of real drone and bird targets is a major focus of this work shared in Chapter 6 which contains detail of the synthetic data approach and validation alongside real data with CNN analysis. The use of synthetic data for real classification has been done before with drone bird classes [119], [250], and this work builds on top of the foundations laid with considerations of motor speeds providing a novel and powerful contribution to the field.

The publications of H. Dale [120], [124], [251]–[253] are uniquely relevant as these works were conducted with the same radar systems as in this study, and was supervised by some of the same University of Birmingham staff as this project. The results provide excellent background and initial exploration of CNNs for bird and drone classification with L-band radar. These studies along with other notable similar works will have the detail of their bird and drone CNN findings shared in 5.1.1 alongside and with comparison to the results and methods of this work.

Also and very recently, [76] used rural, L-band staring radar spectrograms of entire tracks of single birds, bird flocks, rotary and fixed wing drone data to train a residual, split-attention network. Their 5 class problem used 4843 samples for supervised training and 1574 test and validation samples captured between 0.3-2 km range, showed an average classification performance of 98%, as well as advantage over other common CNN architectures without optical pre-training, at high SNR, and similarly although not the best when tested at artificially reduced SNR. This shows that CNN architectures can be built and trained to a high standard without transfer learning, however making these robust to short dwell times is another question. Elsewhere [180] compared a sweep of transformer and CNN architectures in a recent classification study on single spectra at X-band on a 8-class problem of: drones, birds, helicopters, ground vehicles, a windmill, and other aircraft, and clutter; achieving $\sim$ 74% with the CNN that was marginally better than for the transformer.

Last to mention is the application of unsupervised learning using Autoencoders to radar drone and bird data, which has been done [152]. The approach in [76] performed end to end training in one supervised stage while [152] used an end-to-end but two stage training, first in an unsupervised manner then fine-tuning in the described supervised fashion to achieve classification. Denoising autoencoderss have also been used for drone detection on a 1-D range profile to overcome the presence of strong clutter in [194] (and for aircraft in passive radar [57] as well as for range-Doppler maps in [254] for indoor human activity recognition), and a hierarchical fusion network explored improving classification of birds and drone targets with an unsupervised algorithm [152]. Otherwise, more significant volumes of literature using the autoencoder for data limited

Table 3.1: Selection of recent radar literature utilising deep learning. Att., using attention mechanism; XAI, explainable artificial intelligence; BNN, Bayesian Neural Network; CRNN, Convolutional-Recurrent Neural Network

Application	Approach	Algorithm	Citation
Drone Detection	NN	AE	[194] [152]
		CNN	[89] [194] [92] [255] [247] [126] [248] [88]
		RNN	[89]
		Att.	[76]
Automotive	NN	CRNN	[218] [219]
	MM Fusion	CAE	[222]
		CNN	[223]
SAR	NN	CNN	[205] [206] [207]
	XAI	BDD	[211]
		BNN	[212]
		CNN	[213]
Human	NN	CNN	[234] [235]
		CAE	[99] [254]
		BNN	[224]

human activity classification tasks and other radar applications are in existence, and Chapter 8 will contain a dedicated review of unsupervised approaches and will present experimental radar results to enable data clustering and improved classification over supervised methods alone using our collected dataset.

3.4 Chapter Summary

This chapter, in Section 3.1 reviewed the essentials to performing classification with machine learning, and Section 3.2 briefly presented the fundamentals of deep learning with neural networks. Section 3.3 finally presented uses of deep and machine learning in radar literature at large which was progressively narrowed down to classification of bird and drone spectrograms that is the primary objective of this work. Table 3.1 presents a summary of radar literature cited

in this chapter, to showcase the variety of approaches that have been shared[2] Overall, it has been shown that deep and machine learning offer an almost overwhelming wealth of approaches and techniques to data driven classification of targets of interest. Deep learning's superior performance over machine learning techniques come at the cost of reduced interpretability and/or a higher barrier of entry or the requirement to outsource expertise. The unique circumstances of an environmental condition and operational tasks will require an assessment of how a radar can have its particular advantages exploited for the most robust classification. Thus based on the review shown in this section, for this thesis to have a meaningful positive impact on data-driven detection for all drones and all radars, we will propose the following advances in the following chapters:

- Chapter 5 will produce a benchmarking of real, urban bird and drone staring radar datasets of various target configurations using an established CNN method based on optical transfer learning from AlexNet [249]

- Chapter 6 will present an approach for creating high fidelity μDoppler synthetic data, based on a simple and highly flexible simulation using meta-data extracted from a drone after a real drone flight

- Chapter 7 will rigorously validate the previous chapter's simulated data for use in synthetic training of CNNs compared against those trained with real data in Ch. 5, helping to overcome the burden of small data quantities that is in total contrast to reaping the benefits of deep learning

- Chapter 8 will perform a fairly extensive review of the less well established unsupervised learning techniques for bird and drone data, while exploring the quality of the unsupervised learning process and also demonstrating them to achieve excellent classification results trained on real and synthetic data

[2] Later in Sec. 5.1.1 Table 5.5 will display recent literature concerning bird-drone classification alongside the original results of this thesis.

These chapters, and indeed any investigation of machine or deep learning, are fundamentally reliant on the data for performing the training and testing of any classifier or inference model. While the focus of this work is on drones for their detection, the bird targets that are the confuser targets also need to be well understood in order to robustly classify drones correctly. The following Chapter 4 is dedicated to showing and explaining the processes for data collection of drones and birds, using the radar system installed at the outset of the project detailed and discussed in Sec. 2.3. This data collection process took place within a densely urban and suburban city location, that provides a significant challenge to the application of deep learning classifiers, due to varying noise and clutter levels across the locations of the field of view. Collection of birds and drones from this environment allowed for the building of a substantial database of target signatures that, allow us to verify and extend reported performances of common classifiers for drone targets in a high-clutter environment (Ch. 5), which subsequently will allow the testing of new approaches in classifier training and optimisation. The following chapter will detail the targets used, the trial procedure, the quantities and spectral qualities of the different targets used to create datasets used in the later chapters.

Chapter 4

Experimental Data Collection

In this work investigating classification of bird and drone targets, a campaign to collect a dataset of real radar signatures was necessary. Urban drone and bird data was collected in an opportune and controlled manner that allowed realistic assessments of a range of classification approaches shown in later chapters. For deep learning, there is an extra onus to collect large amount of data to allow robust training not troubled by overfitting of a small, self-similar dataset. This chapter will explain the methodology of each of the data collection processes, and each section will finish with example spectrograms collected that show the nature and variety of targets collected. Section 4.1 explains the procedure and results of gathering drone returns, Section 4.2 mentions the control bird trials that this work helped to undertake with the benefits, challenges and results of the process and finally, Section 4.3 shares the opportune data collection process employed that enabled us to take advantage of the countless natural birds habituated in the Birmingham City area. Section 4.4 briefly summarises the collected data as used in the remaining work on deep learning.

4.1 Control Drone Trials

This section will explain the process and details the efforts undertaken in collecting a dataset of real, multi-rotor drones with an L-band staring radar in an urban environment. The radar data of drone signatures collected formed part of the essential basis of classification investigation in the following chapters based on deep learning and synthetic data.

4.1.1 Motivation

As a first step for the assessment of data-driven classification approaches, systematic drone signature collection using real drones with a real radar system began and was maintained throughout the duration of the project. It is a major goal of this thesis to reduce the burden of radar data collection for the training of deep learning classifiers, but assessments of synthetic data require varied real data ideally from a number of platforms to be fully assessed. The construction of a large, real, varied dataset is required as a basis when comparing findings from any alternative approach in classifier training or design. It is also of great benefit to collect drone signatures in a real, urban environment that will significantly impact the composition of returns. The aim is to collect a dataset that will feature challenging returns from a real environment that is not an idealised simulation of an operational condition. Examples of such environmental factors that were captured are

- weather conditions, such as strength of wind,
- clutter conditions, from various buildings near flight locations, and
- bird presence cohabiting the resolution cells of measured airborne drones.

Data logs from the drone flight were collected as *ground truth*. GPS data enabled rapid production of spectrograms from the 4D staring radar data cube by extracting the timeseries samples from the resolution cell determined to contain the target. Further drone meta-data exists in the drone logs that contain information of potential interest when studying their radar returns,

Table 4.1: Drones used in data collection trials.

Ref.	Drone Make	Mass / kg	Price / $ (2023)	Body Diagonal / mm	Blade Radius / mm
Matrice	DJI Matrice R300	4	7,000	605	190
Inspire	DJI Inspire 2	3	2,000	895	220
Mavic	DJI Mavic 3	1	1,500	375	90
Mini	DJI Mini 2	0.249	300	213	60

such as drone roll, pitch and yaw, measured wind-speed, flight operation (manual or automatic), and *motor speeds*, which this work will explore in great detail as the subject of Sec. 6.3.2 being a key enabler to creating realistic synthetic data.

4.1.2 Drone Inventory

Four DJI drones were routinely used in data collection and Table 4.1 presents information and the quantities of data collected over the course of the project. Section 4.1.4 later will show how these targets appear to the radar via a time-frequency spectrogram.

Images of the four drones are shown in Figure 4.1. The two large drone models, *Inspire* and *Matrice*, have large propeller blades (~20 cm) that produce strong Micro-Doppler (μDoppler) modulations detectable by the radar system used. For the two smaller drones, *Mini* and *Mavic*, that have a blades radius of ~8 cm, these at the L-band operational wavelength used in the project do *not* produce any significant, detectable μDoppler as is shown in Section 4.1.4. The inclusion of these targets in the dataset adds to the challenge of deep learning classifiers to be wise to drones without the characteristic sidebands. Generally, the presence or absence of blade μDoppler will be a function of operating frequency and propeller size.

4.1.3 Trial Procedure

A drone is a very convenient test subject, as most market drones come with a GPS recorder. GPS information is essential for recording the drones' position at each time-step, and the raw and processed radar data allows for a great deal of analysis of the radar's performance in that situation. A drone trial consists of the following steps:



(a) *Matrice*



(b) *Inspire*



(c) *Mavic*



(d) *Mini*

Figure 4.1: Drone Target Inventory. Floor tile length ~ 1 m

1. Securing a trial site (in the urban city area) where drone flights are permitted and safe

2. While radar(s) are recording:

    - Launch drone, fly a predetermined flight pattern, land drone
    - Image the drone on the way up and down for a verifiable timestamp
    - Note the flight pattern, configuration and time occurrences of the drone and radar during launch

3. Archive the various drone logs

4. Convert GPS logs into trajectory files, that will be used to extract the relevant timeseries radar data

5. Convert the raw recorded radar data from native binary format into MatLab 4D datacubes (64 range/slow-time gates, 16 elevation channels, 4 azimuthal channels and 2048 fast time samples) cropped to when the flight was taking place.

6. For the length of a trajectory file, find the range and angular channels where the target appears and extract the fast time samples from this datacube to form a 1-D complex baseband signal

7. Apply a STFT to resultant 1D signal to form 2D time-Doppler spectrograms that are used to build classifiers

All of the (DJI manufactured) drones in the fleet allowed access to GPS flight-records that contained the location information, as well as other metadata of interest such as wind measurement, drone axis state (roll, pitch, yaw), controller input, and much more. An additional record was archived where possible, which were the `.dat` files. For the two larger drones (*Matrice* and *Inspire*), the `.dat` files were decryptable by the openly available software *CSVview* at datfile.net, which contain more flight data that is less adulterated than the usual, more universally accessed 'flight record csv'. Four fields of this csv are the four motor speeds of the quadcopter in Rotations Per Minute (RPM). The dat files of the smaller *Mini* and *Mavic* were not decryptable by this software due to their Software Development Kit (SDK) not yet being available. Motor speeds are used in the rest of this project as a valuable tool for the simulation of drone signatures resulting in the project's first contribution in [1], and have since been used in further published work to validate radar-based motor-speed measurements in [6]. This is the subject of Chapters 6 and 7. The inability to extract the smaller drones' motor speeds was not a limitation of this project as these targets did not produce any considerable μDoppler, however it would be a consideration for higher-frequency radar systems where the μDoppler of smaller targets have greater presence. Some suggestions to enable future work in this regard are presented in the final discussion in Sec. 9.2.

The L-band staring radar produces a 4D datacube capturing the returns in 3D space in azimuth,

elevation and range resolution cells, and in time including every pulse in the CPI. To extract the raw I/Q timeseries results for the tracked target, the slow time axis returns are gathered for only the resolution cell that is determined to contain the target using the recorded GPS information. From this 1D vector of timeseries I/Q data, spectrograms could be formed at ease with various DSP implementations. Previously, Figure 2.10 illustrated this creation of a spectogram from a datacube. Research of deep learning based classification using I/Q data alone for classification has been performed in drone detection contexts [256], but using a STFT has three significant advantages:

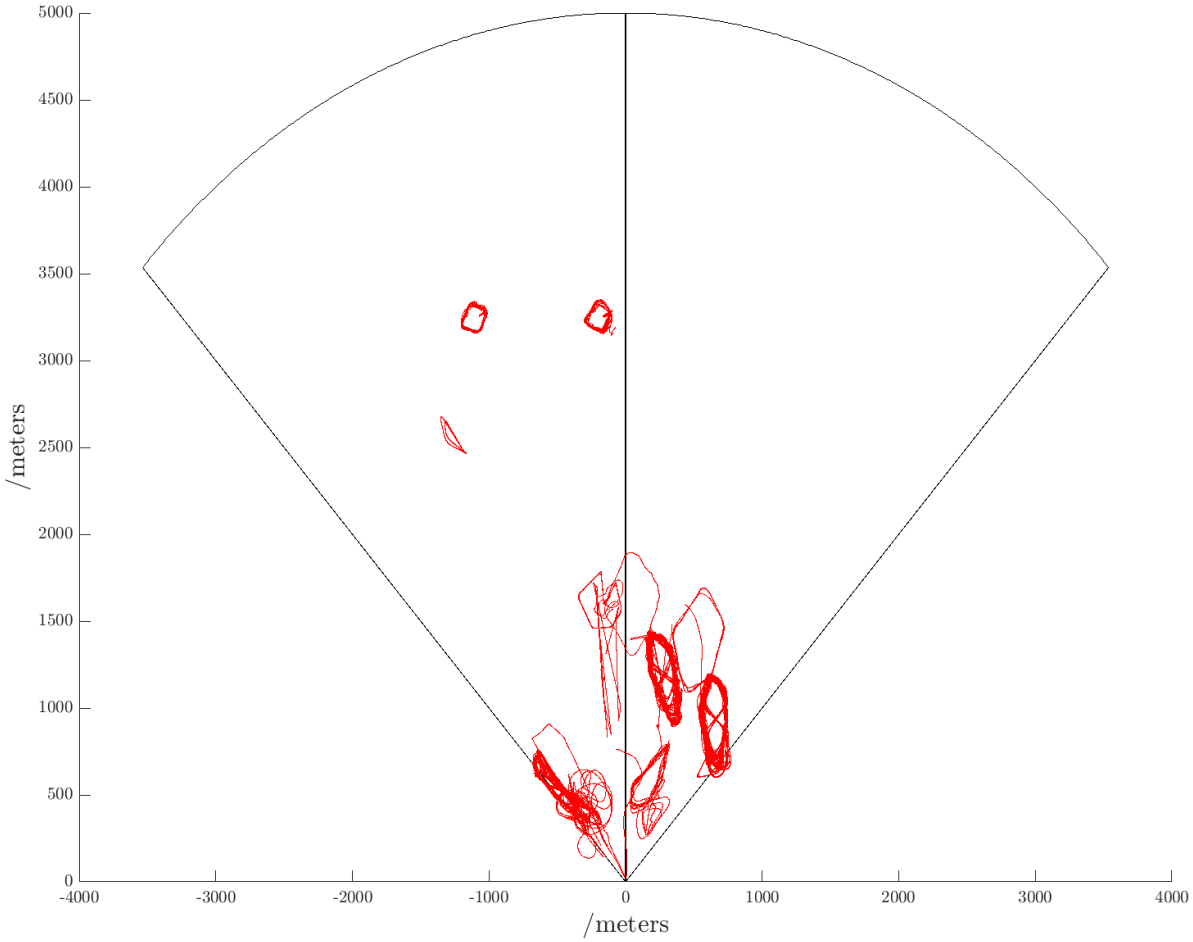
1. classifiable real world features of drone targets are inherently tied to real world velocities and the FFT separates the frequencies that a deep learning algorithm would need to learn anyway,

2. a time concatenation of spectra of CPI shows both instantaneous and temporal features to produce a rich 2D output that can avail of the mature deep learning classification approaches such as convolutional layers,

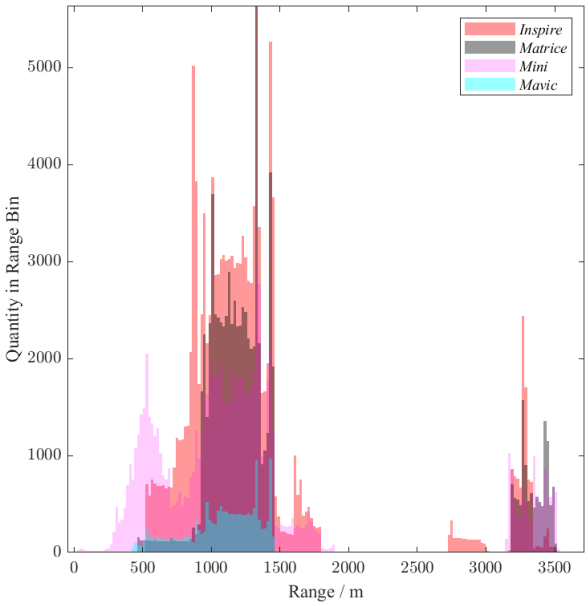
3. the time-frequency representation allows for easy interpretation of data samples, particularly at low SNR that is useful for understanding mis-classifications and in enabling large scale data collection.

As shown in Sec. 4.1.4 following, spectrogram representations of drones, and later for birds in Sec. 4.2.2 and Sec. 4.3.3, display many features that can be tied intuitively to the targets physical construction and could be used for robust classification of expected targets. Thus, part of the challenge is to understand the variety of possible targets which is presented in this chapter. Later sections will use subsets of the collected data for individual experiments for machine learning and synthetic data evaluation, with Figure 4.2 showing the locations, ranges and heights of the drone targets used for creating the datasets in the remaining chapters. In total, for each drone; 6.2 hours of *Matrice* data was collected over 116 flights, 10 hours of *Inspire* data over 234 flights, 1.1 hour of *Mavic* data over 42 flights, and lastly 7 hours of *Mini* data collected across

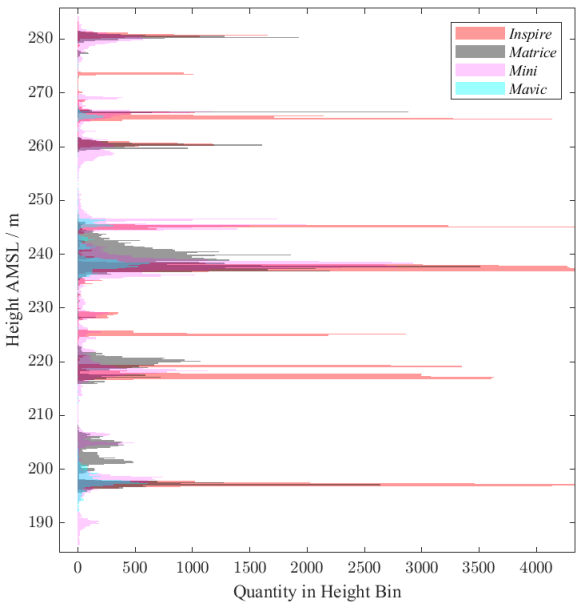
172 flights.



(a) Top down plot of trajectories



(b) Histogram of ranges of collected drone spectra.



(c) Histogram of heights AMSL of collected drone spectra.

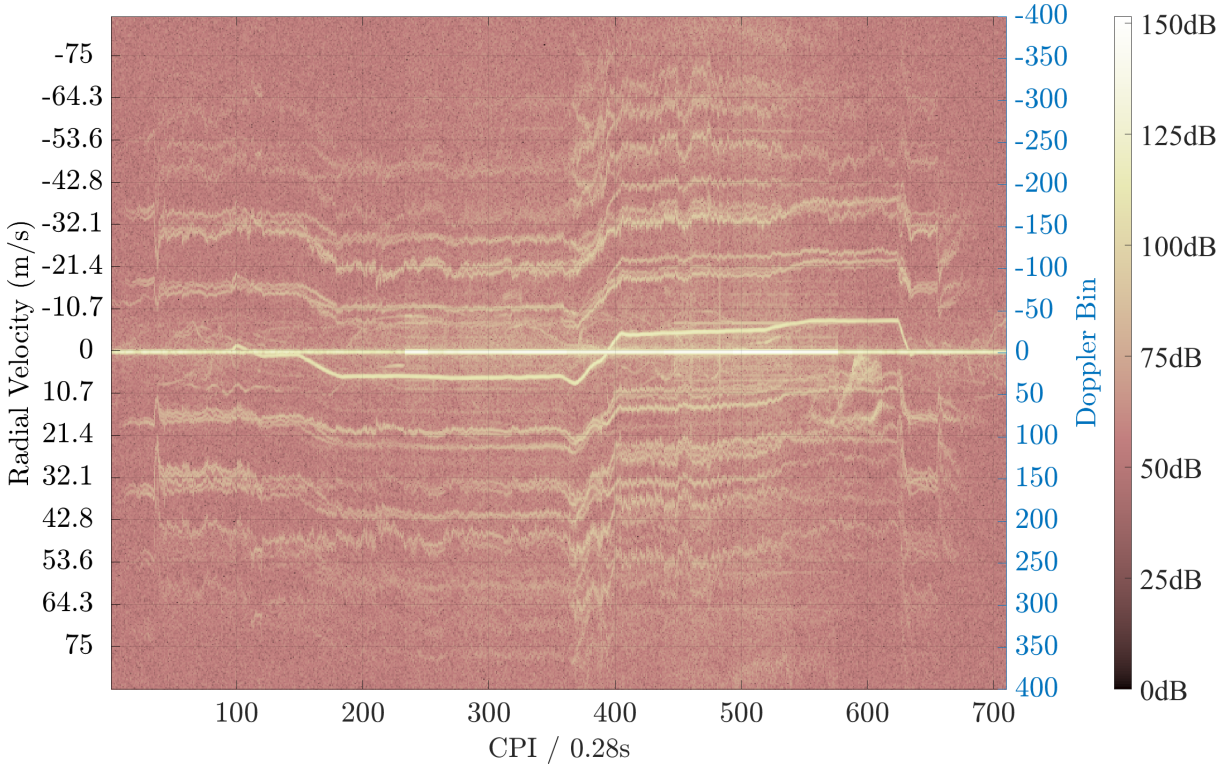
Figure 4.2: Plots summarising locations of drone data collected.

4.1.4 Drone Signatures Collected

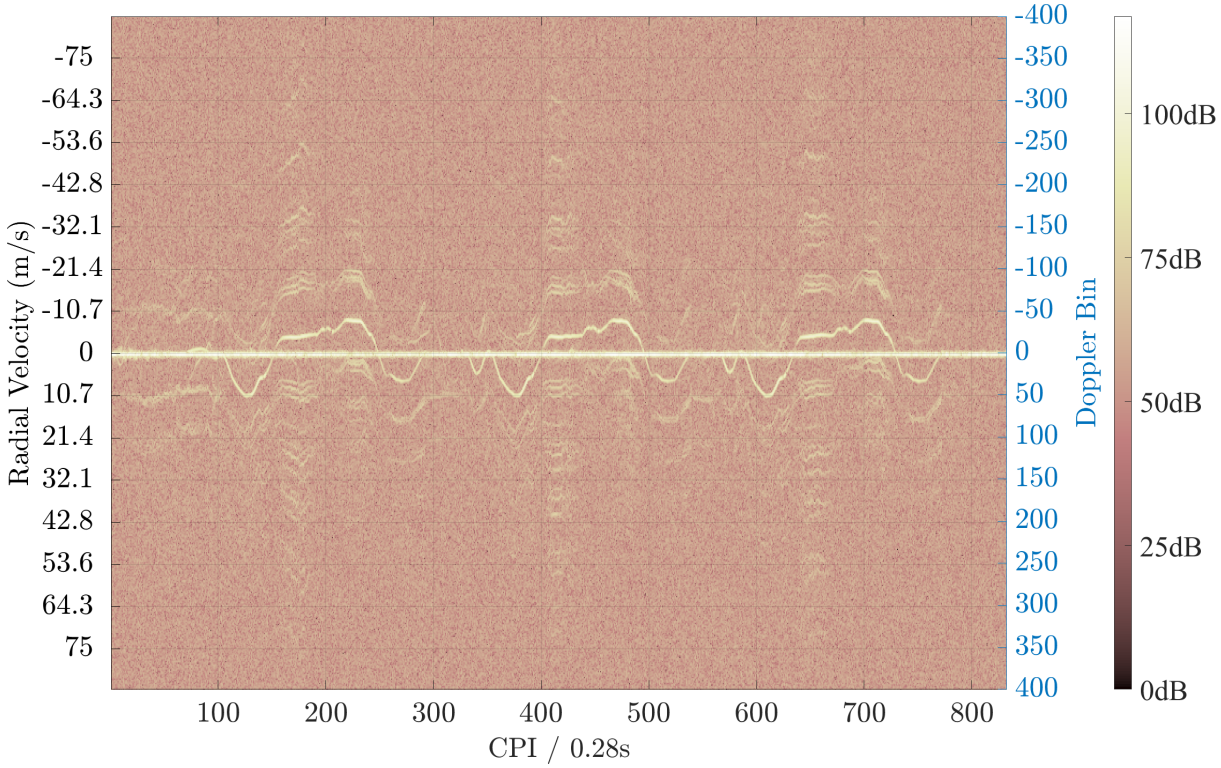
Spectrograms are produced using a Short-Time Fourier Transform (STFT) of CPI duration 0.56 s with a Blackman-Harris Window and 50% overlap, as previously described in Sec. 2.3.2, as were all spectrograms presented in this and later chapters.. Resultant real spectrograms for the *Matrice* target of flights lasting for a duration of 3 minutes and 48 seconds are shown in Figure 4.3. Note that the characteristic μDoppler sidebands induced by the rotor blades are clearly observed around the main body. For the two example spectra, Fig. 4.3a shows the *Matrice* at a relatively close range of 1.5 km with Fig. 4.3b at a further range at 3.4 km. At near range, the propeller μDoppler is very strong, clearly displaying multiple groups of sidebands and features detail of the motor fluctuations within each harmonic group. At the longer range, the presence and count of the sidebands is not constant.

Figure 4.4 shows the *Inspire* at three flight heights with Radar#1 at a similar 1.5 km range but at a location with significantly more urban clutter. There was also a considerable amount of bird presence in the area, of which their returns can be seen within the central 60 Doppler bins of each Fig. 4.4a - 4.4c. At each height the drone presence is obvious via the sidebands, which makes it quite distinct from non-Drone radar echos. The strength of the body and sidebands do not degrade much between the 100 and 80 meter heights, but at 60 meters their fullness and power are more significantly reduced. At this location, tall trees were present between the flight location and the radar transceiver which could have reduced the overall signal of the drone at the lowest height.

The *Mini* and *Mavic* smaller drones did not produce significant μDoppler due to the size of their propeller blades. Four spectrogram representations of these targets are shown in Figure 4.5. Fig. 4.5a and 4.5b, show the smallest *Mini* drone at the same locations as Figure 4.3 for Radar#1. Without μDoppler, small drone presence amongst birds poses a more challenging classification task. Fig. 4.5c and 4.5d show the *Mavic* drone with nearby bird presence. The dominant return of these spectra is the drone body as the spectrogram was formed using a GPS recording.

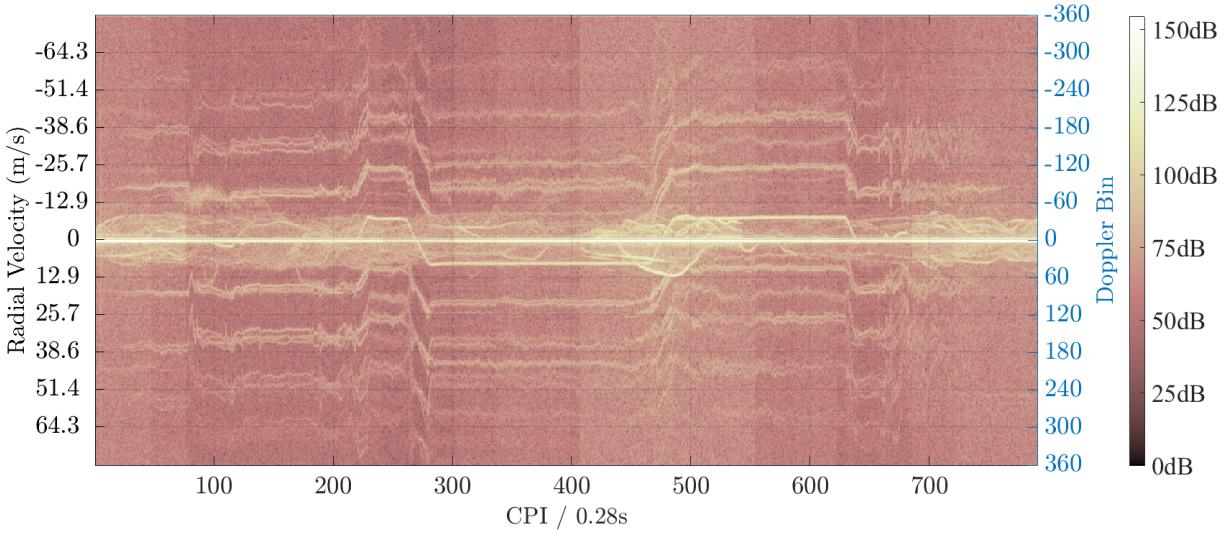


(a) At Near Range, 1.5 km on a calm day

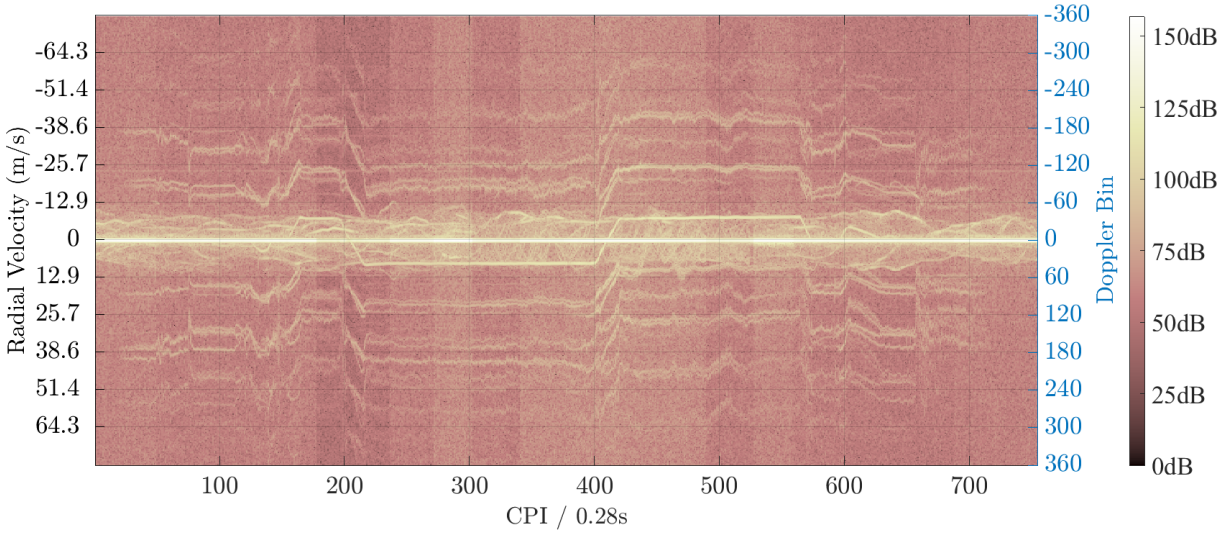


(b) At Long Range, 3.4 km on a windy day

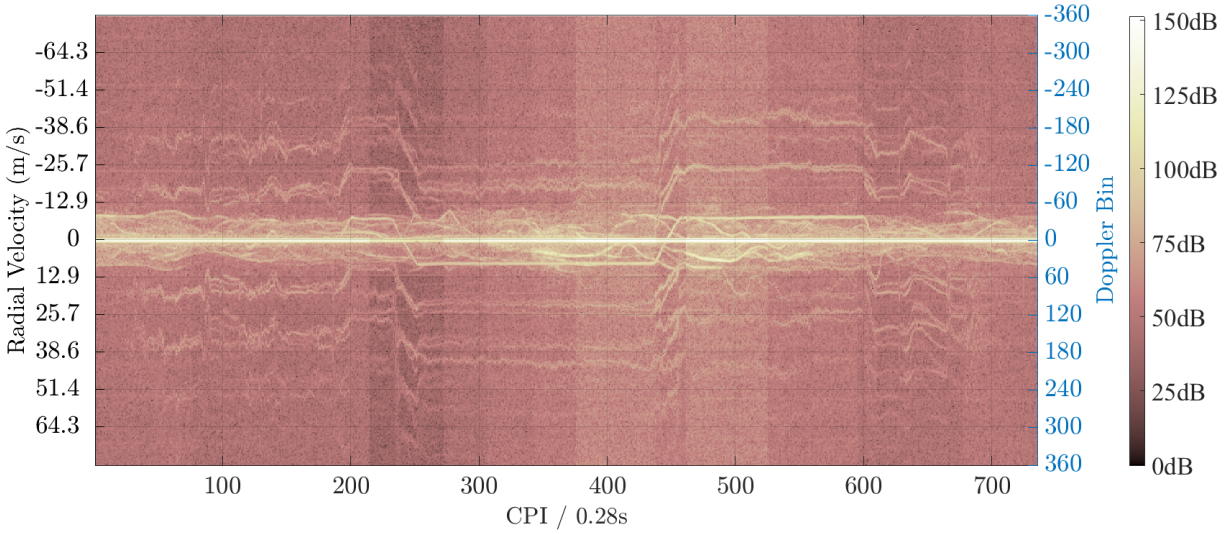
Figure 4.3: Spectrogram of the *Matrice* using Radar#1.



(a) 100m above ground

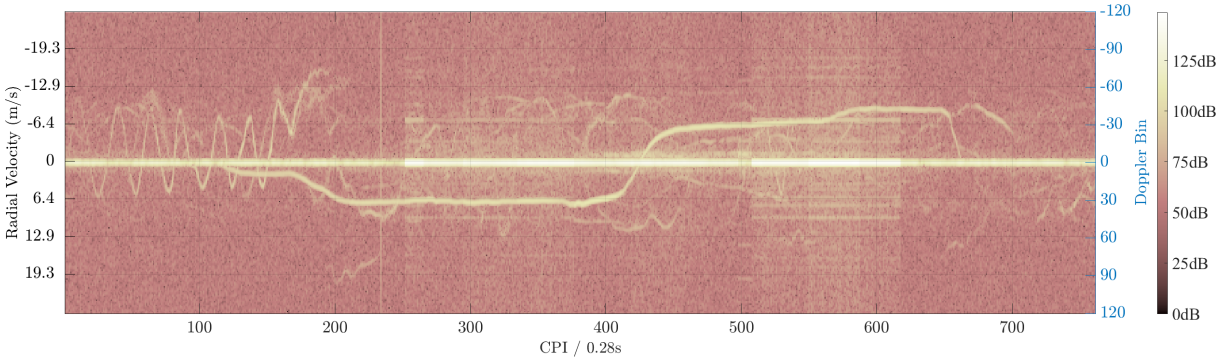


(b) 80m above ground

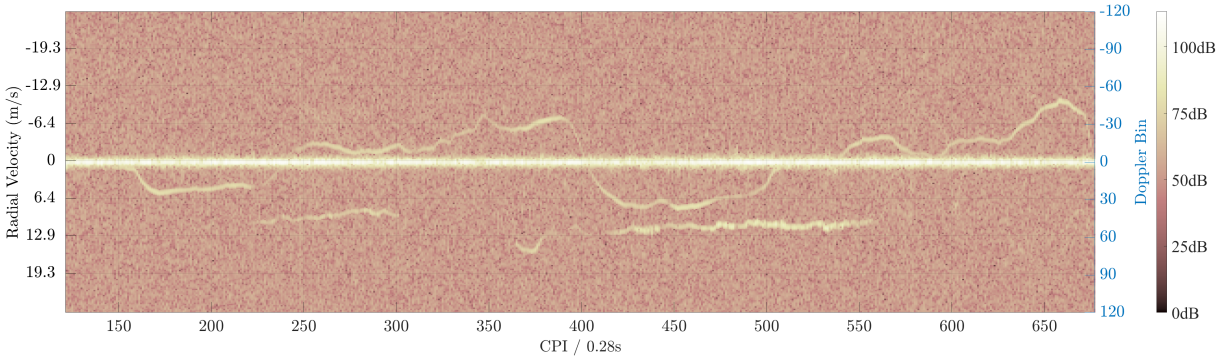


(c) 60m above ground

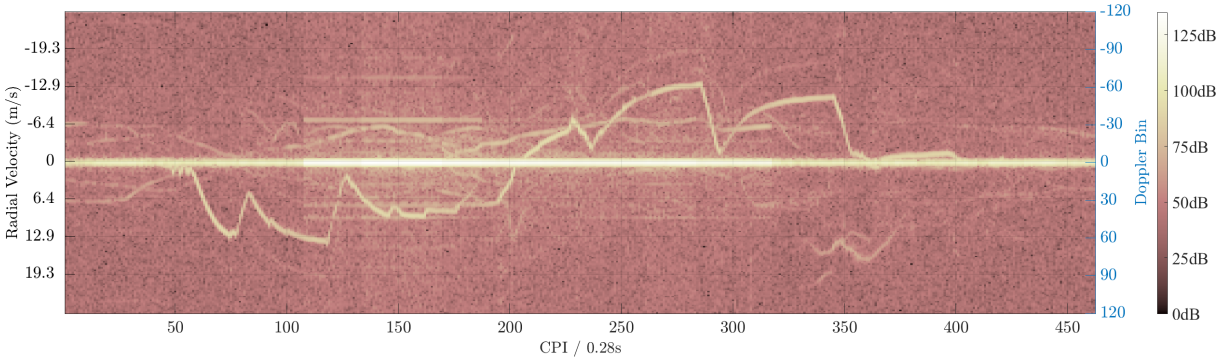
Figure 4.4: Spectrogram of the *Inspire* using Radar#1 at various heights with bird presence at low velocities.



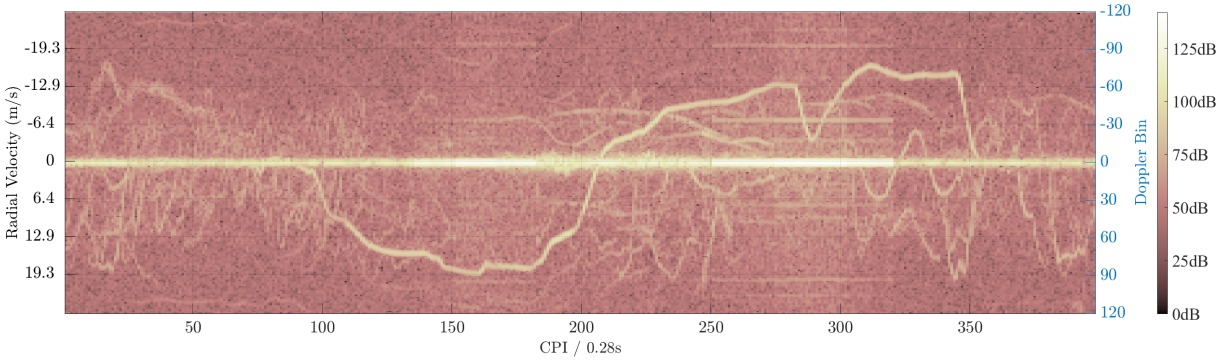
(a) *Mini* at 1.5km on a calm day



(b) *Mini* at 3.4km on a windy day

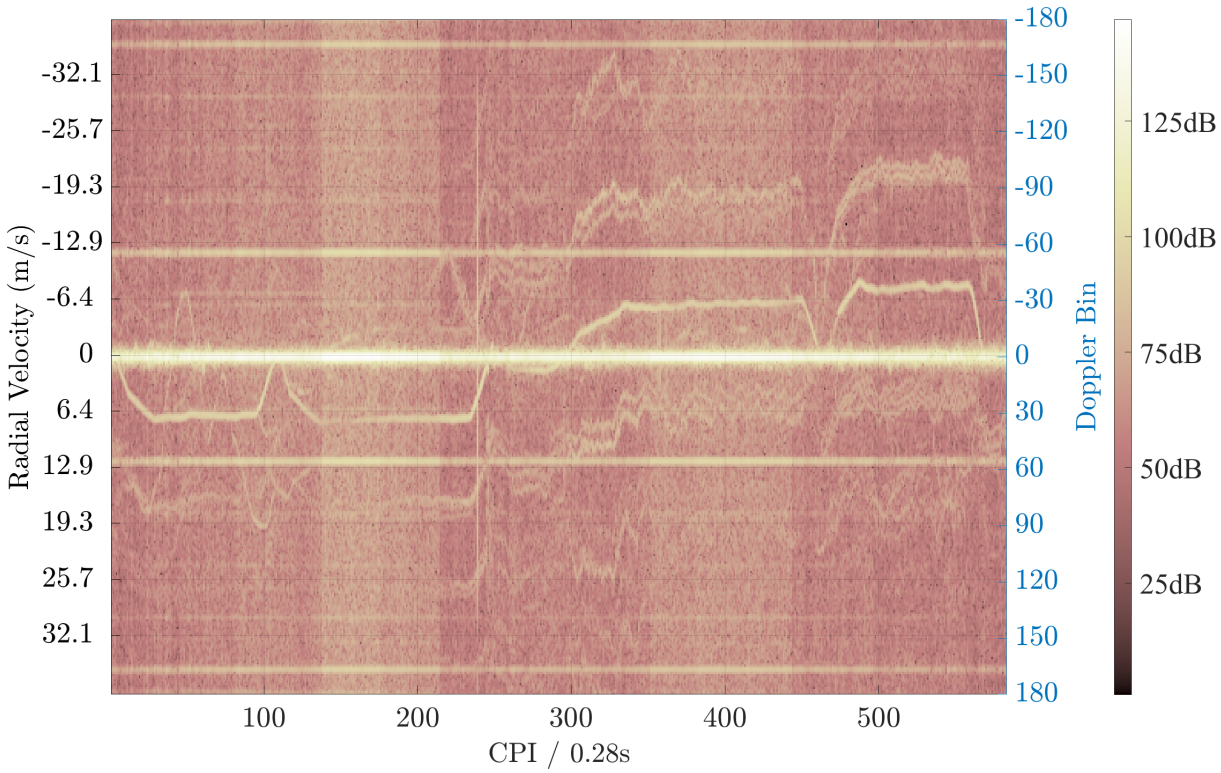


(c) *Mavic* with minor bird presence

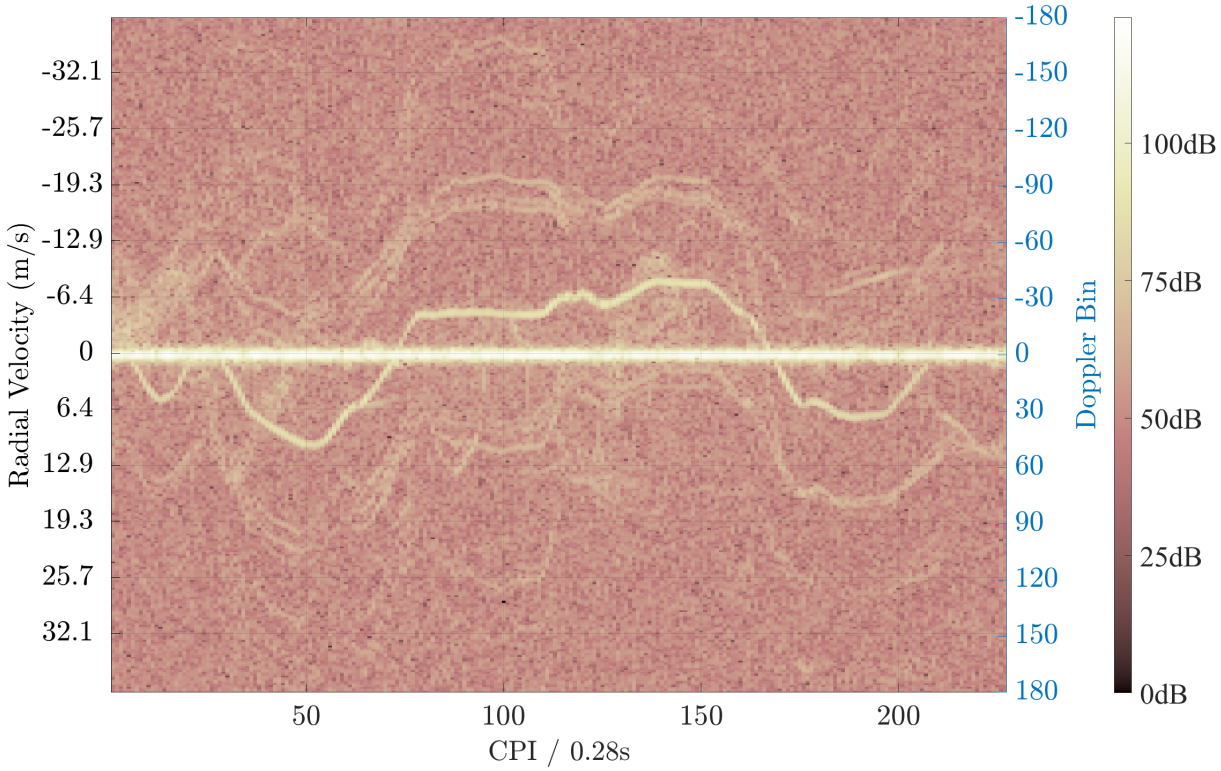


(d) *Mavic* with more bird presence

Figure 4.5: Spectrograms of the small, μDoppler-less drones using Radar#1.



(a) *Matrice* at 1.5km



(b) *Matrice* at 3.4km

Figure 4.6: Spectrogram of the *Matrice* using Radar#2.

Radar #2 was also used for data collection in the same way. Compared to Radar#1, the signatures have lower SNR due to the impact of urban clutter causing location-dependant raised noise backgrounds and other effects. [10] uses these exact radar systems and provides an analysis of the effects of clutter on the measured spectra of the two radars at a variety of locations. Figure 4.6 shows the *Matrice* with Radar#2 at the same locations as Figure 4.3 and Figs. 4.5a and 4.5b. In Fig. 4.6a, the effect of location-dependant raised noise background at closer range is apparent and makes the sidebands difficult to observe in certain resolution cells. As well, the horizontal lines at constant frequency are a symptom of strong clutter and hardware limitations that can make interpretation of data difficult. More information of the causes and effects of the noise induced by the urban environment on these radar systems are described in [7], [10].

Returns from Radar#1 and #2 were both used in datasets for deep learning featuring in test, validation and training subsets. The level of signature variation for the same target across the two systems is much greater than what would be expected from any one radar system, so this poses a significant challenge to the deep learning algorithms later used in Chapters 5 and 7 to generalise the physical returns of the target amongst a wide range of possible radar backgrounds. Greater insight of this difference is shown in the unsupervised clustering work that is reported in Section 8.2[1]. The use of two differing radar systems within the already highly dynamic urban environment brought a greater degree of background variation that ensures the classifiers built as part of this work, are able to generalise against different clutter backgrounds. Synthetic data can be used to create scenarios of theoretical drone presence in totally unforeseen radar circumstances.

[1]The unsupervised dimensionality reduction techniques of PCA, tSNE, UMAP and CAE are applied to the collected data set showing that the radar of origin was a major factor bringing variation to the collected returns.

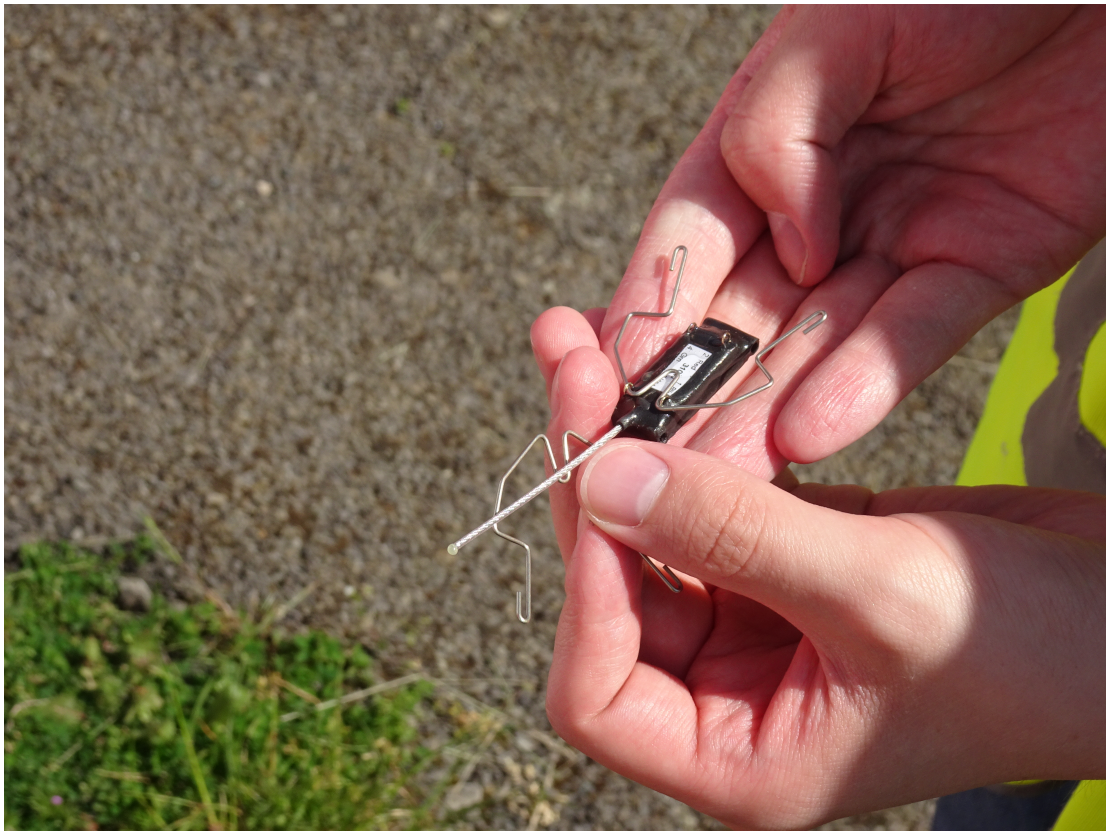
4.2 Control Bird Trials

This subsection shares the process and results of control bird trials that took place over the duration of the project. Collecting bird signatures with corresponding positional and species information is valuable for assessing the radar's performance on these ubiquitous targets. Radar trackers allow an estimation of an isolated bird's position, but information of the species, if it was a single bird or group, and where it took flight and landed may be only estimated from the track and the appearance of the spectrogram. Two control bird trials took place with different species which enabled the analysis of a variety of returns from bird targets. These control bird trials gave confidence to collect bird data at scale for deep learning purposes going forward.

4.2.1 Trials Undertaken

Two control bird trials of different motivation were conducted as part of this work; measurements of Birds of Prey in a rural environment were performed at Cranfield University, and racing Pigeons within the urban environment at University of Birmingham. Both of these trials used *PathTrack NanoFix*[2] GPS tags to record the trajectory of the birds. Their use in the trials are further described in [8], [257], and the tags are shown with their attachment to two birds in Figure 4.7. These tags required experienced handlers to attach to the birds, and their memory and battery limitations required swapping which added to the challenge to collect data rapidly. Their ability to record bird position in a complex urban environment would be essential to validating research of tracker performance that was not within the scope of research attempted in this work.

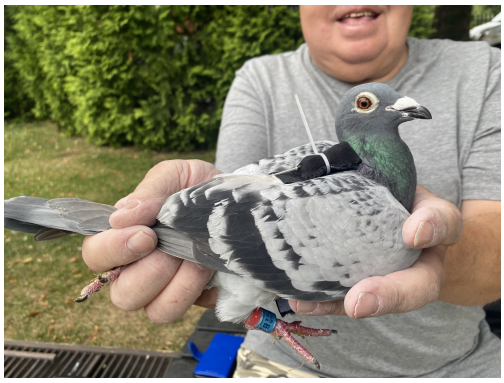
[2] www.pathtrack.co.uk/product/nanofix-geo



(a) Pathtrack NanoFix GPS Tags



(b) Affixed to tail feathers of Grey Buzzard Eagle



(c) Within harness attached to racing Pigeon.

Figure 4.7: GPS Truth from Cranfield Trial.

On 1st July 2021, Cranfield University Airport was closed for 2 hours for a trial lead by UoB with the International Centre for Birds of Prey (ICBP) providing several Birds of Prey, miniature GPS tags to live track and record their position, and a fixed wing drone shaped as bird flown as a lure to encourage their flight. The lure was not always necessary for the birds to fly, and some birds were flown without the lure. The birds species flown on this trial were the Hybrid-Falcon,

the Grey Buzzard Eagle, the Yellow Kite, the Gyrfalcon and also the fixed-wing *Rocrow* drone manufactured by *Wingbeat*[3] which was the lure. The trajectories flown are shown in Figure 4.8, with Fig. 4.8a showing all the GPS tracks collected, Fig. 4.8b showing the track of the lure being chased by the Hybrid-Falcon, and Fig. 4.8c displaying the *Rocrow* with controller.



(a) All GPS truth tracks from Cranfield Trial



(b) Hybrid Falcon (red) chasing a human operated RoCrow fixed winged drone (pink)



(c) The'RoCrow' fixed winged drone flown as lure

Figure 4.8: GPS Truth from Cranfield Trial.

On 20^{th}-21^{st} of July 2022, 10 racing Pigeons provided and released by an experienced handler from inside the field of view of the two radars in Birmingham. The trajectories taken by the birds are shown in Fig. 4.9 and travelled left to right towards their roost in north-east Birmingham, avoiding the city centre. A track of great interest was one that flew directly by the radar, and its spectrogram and some analysis will be had in the section following.

[3] www.rofalconry.com

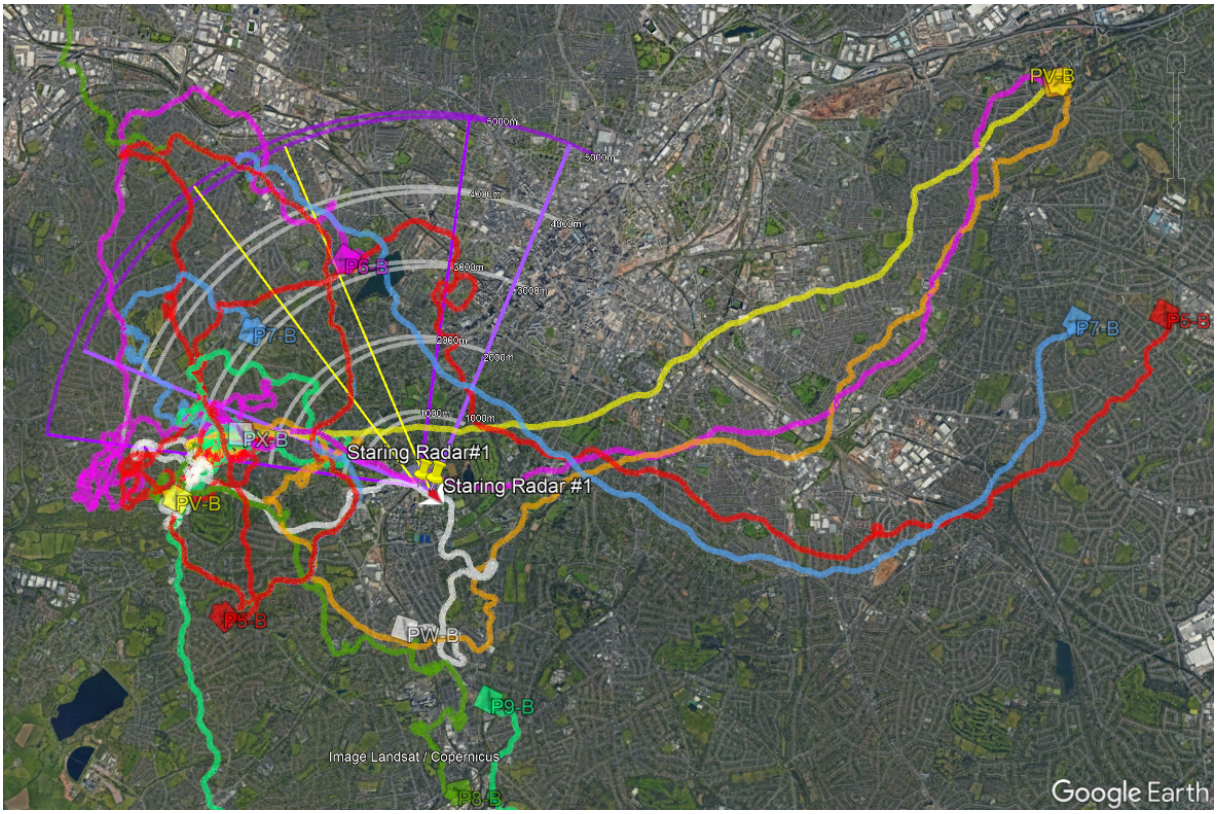


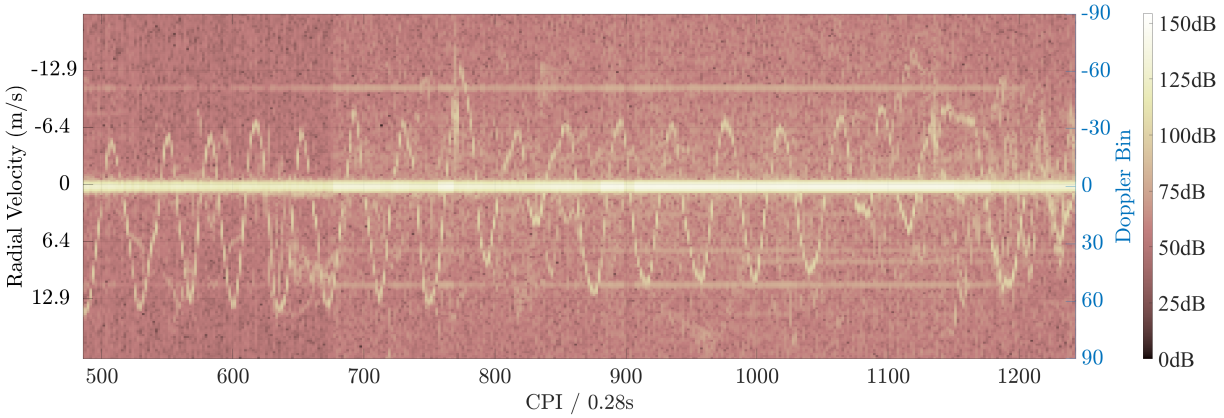
Figure 4.9: GPS data collected of all Pigeons with both radars' fields of view.

4.2.2 Control Bird Signatures Collected

Firstly, an example signature collected at the Cranfield trial in collaboration with ICBP [5] is shown. Figure 4.10 shows the track and spectrogram of a Black-chested Buzzard-eagle that flew to a peak height of 190 m altitude then descended slowly in a spiral pattern, characteristic of raptors and other large wingspan birds to hang in the air and assess their surroundings. The appearance of the spectrogram shows clearly the direction of the bird's travel over time. This pattern of flight where the body Doppler oscillates is also frequently seen in the spectrograms extracted of opportune birds from the radars installed in Birmingham (See Sec. 4.3.3).



(a) GPS Track of Grey Buzzard Eagle flight, and



(b) Spectrogram of part of the spiralling track

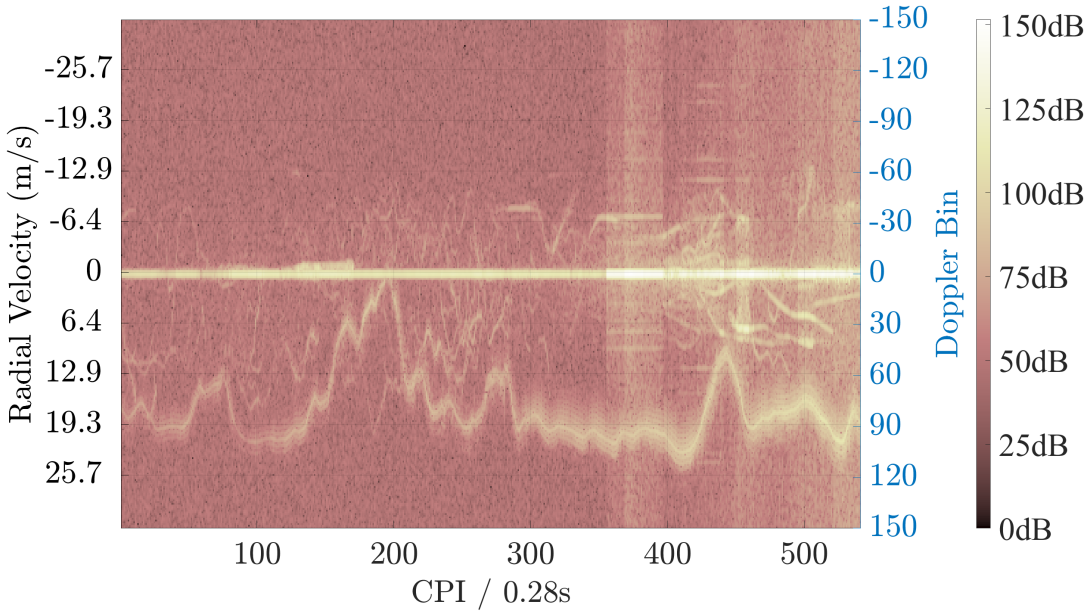
Figure 4.10: GPS and Spectrogram of Grey Buzzard Eagle that flew for an extended period in a spiral pattern at the Cranfield Trial.

Next, Figure 4.11 shows the positional information of a Pigeon that flew from range to close proximity to and past the radar transceiver in Fig. 4.11a with the corresponding tracker spectro-

gram in Fig. 4.11b. The body of the Pigeon is accompanied by clear sidebands, which appears similar to a rotary winged system's μDoppler, in this example using a long window STFT that is greater than the pigeon's wingbeat frequency of approximately 5 Hz (see Section 7.1.1). Interestingly, opportune collected data from the Birmingham radar described in the next section also featured these returns at a variety of ranges up to 4 km. Whilst these limited trials with control birds have provided useful measurements, data from opportune birds form the majority of the spectrograms used in later classification studies.



(a) GPS Track of Pigeon flying close to radar transceiver, and



(b) Spectrogram of Pigeon displaying a form of single sideband μDoppler.

Figure 4.11: GPS and Spectrogram of 'Pigeon-W' that flew close to radar transceiver.

4.3 Opportune Bird Target Collection

This section contains the process taken for at-scale collection of opportune bird targets to support deep learning. Targets were initially selected based on the tracker output of the L-band staring radar employed, then the tracks were further assessed and filtered. The requirement to collect large amounts of data for the training of deep learning algorithms motivated this, but the use of these signatures without any field observed truth necessitated steps to ensure collected tracks were correctly labelled as birds and suitable for addition to the dataset.

4.3.1 Motivation

The control bird trials provided several bird signatures with precise truth information describing their airspace occupation. However, the trained captive birds are expensive to rent and the rate of useful data collected was very slow - too slow to be solely used to collect a dataset large and diverse enough for machine and deep learning. Manual validation of radar output tracks was done in some related work through line sampling transects that allowed the bird species to be associated with the data. This process is labour intensive, slow and inefficient but essential for species label insight for radar-bird analysis, and was reported in [5], [258]. Target category validation using position and direction marking with observation label and photo evidence, using an app such as Theodolite [259], was also attempted and reported in [8], but due to its reduced effectiveness in a highly urban area where line of sight of the radar is severely restricted it was not worth continuing for data collection in this particular project. In the effort to rapidly collect a large amount of bird signatures, harvesting of opportune data from the tracker output was the most suitable approach to allow the collection of both an excessively large dataset, but also one of genuine variety where a large sample of bird target compositions are taken without any artificial intervention by human instigators. The configurations of bird flights observed in nature will far surpass the variety and complexity that can be created or coaxed from a collection of

captive birds brought to a new environment.

4.3.2 Collection and Filtering

The radar produces detections of targets within their zone of operation. From the sequence of detections that are triggered by a moving target, a track is formed that represents the continuous motion of a real moving target. For a track, the tracker positional information and the spectral information that was collected over the course of the track is scrutinised for a semantic class prediction. After a period of recording radar data, the track positional information is read along with the classification information. About 10,000 tracks are saved by the radar every hour, each of which allow for the extraction of the radar spectrogram if the radar was also recording raw data.

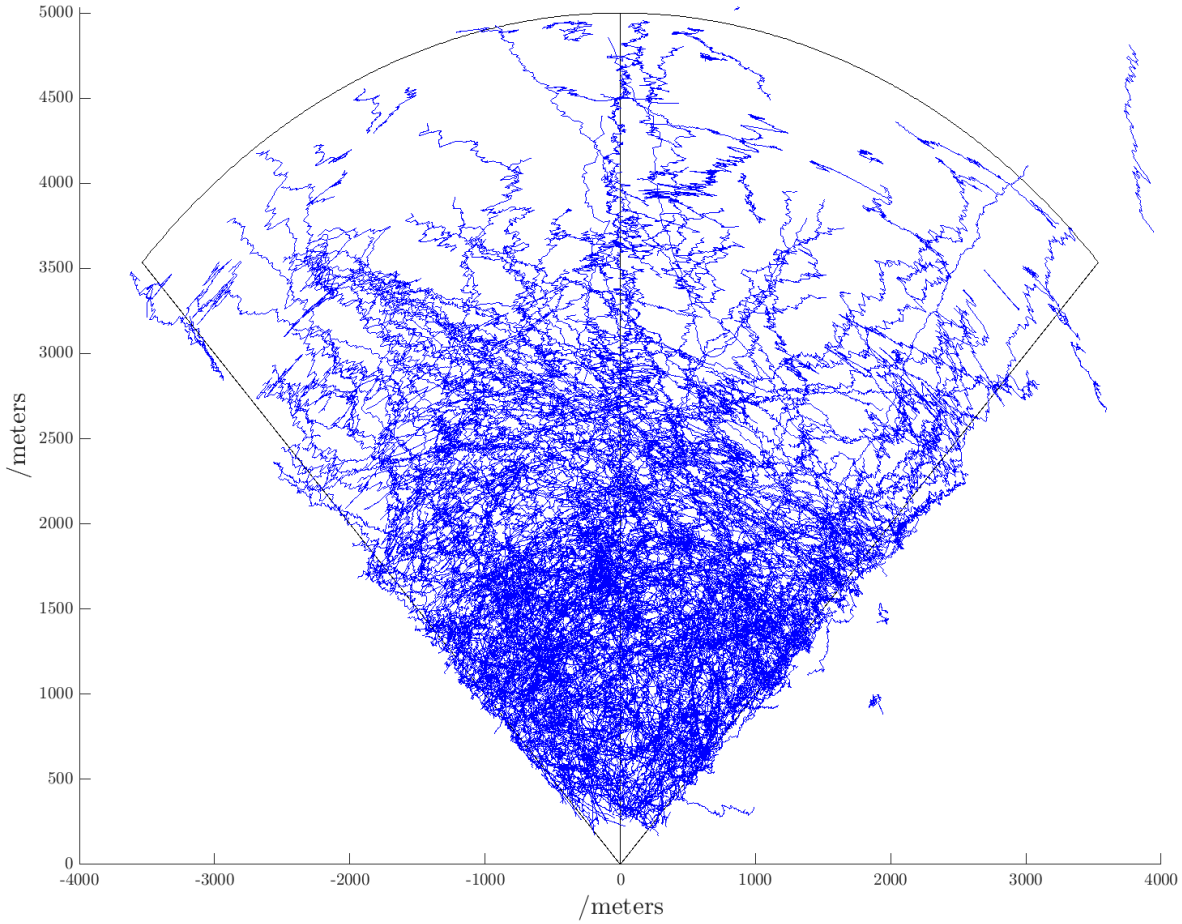
The great quantity of tracks collected by the radar necessitates filtering to ensure real, representative bird targets are included in the datasets. At first, the targets are removed if their duration was too small. Then, targets that were below the height of radar could not be reliably used as at this height ground targets will be detectable as the ground height in the area was greatly varying. Choosing targets above the radar ensures that we are including avian targets in the dataset. As operation was in the urban clutter environment, phantom tracks can be triggered due to the increased level of returns from buildings, along with moving trees and vehicles, that causes detections that are more challenging to convert into a track - thus tracks that are not representative of real targets may be included. The probability of inclusion of such tracks was minimised due to using tracks longer than a set duration, and also removing tracks with too high an average speed. Using a minimum track length of 61 time-steps (17 seconds) seen $\sim$ 80% of all reported tracks discarded. This specific length of track was used to maximise the amount of targets gathered, while reducing the number of small, possibly erroneous tracks that were less reliably verified as bird targets. In the later chapters performing deep learning with this data, samples comprising of 20 CPIs were used. Therefore, choosing a minimum track length of 61 ensured that all verified birds contributed at least 3 samples to the dataset. 2 samples was not

considered worth the additional processing and verification efforts.

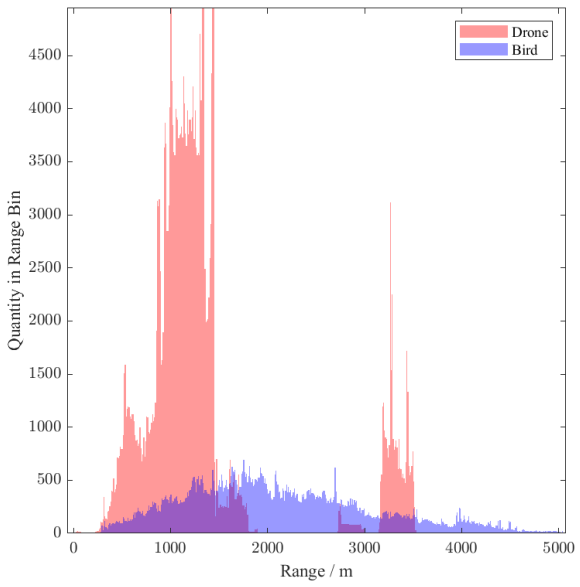
The last stage of opportune filtering was manual, human verification. From the control bird trials, the experience of observing genuine bird returns was applied to the collected datasets before accepting these as labelled data in machine learning studies. Spectrograms were removed from the datasets that did not appear to contain an obvious target, contained an obvious avian non-bird target such as a passenger jet or helicopter, or contained any instances of drone μDoppler anywhere across its spectrogram. All the datasets in this work used later for deep learning were drawn from a total collection of 19.6 hours of bird data from 932 opportune tracks. Figure 4.12 shows the locations, ranges and heights of the opportune birds that have been collected for this work. For comparison, these plots also have overlaid the histogram for drone targets, to allow an appreciation of the relative difference of locations for the two target classes.

4.3.3 Opportune Signatures Collected

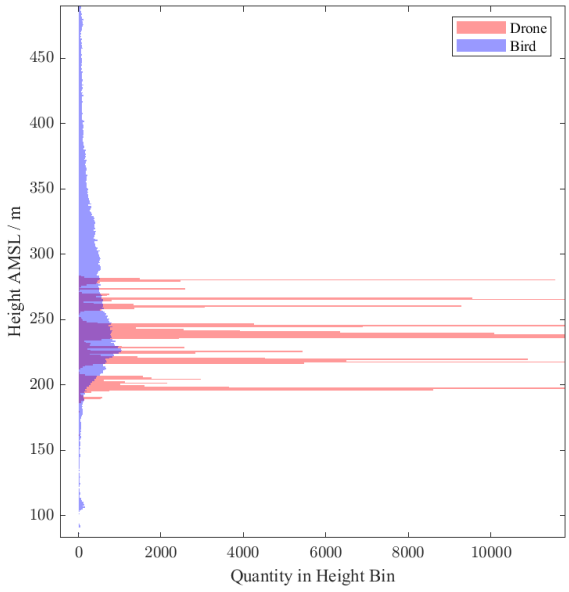
This subsection will briefly share a selection of opportune bird spectrograms that will be later subject to classification assessments by inclusion in the deep learning datasets. The natural, urban bird signatures collected were widely varying such that it would not be unreasonable to describe each track as unique. Figure 4.13 shows a selection of different bird number occurrences that demonstrates the variety of returns naturally observed, that are labelled in the captions by the interpretation of their spectrogram. As well as the great amount of signature variation per track, the length of the opportune tracks were much shorter than the drone GPS recordings. This meant that many more bird tracks were needed to counterbalance the available drone data.



(a) Top down plot of opportune bird trajectories

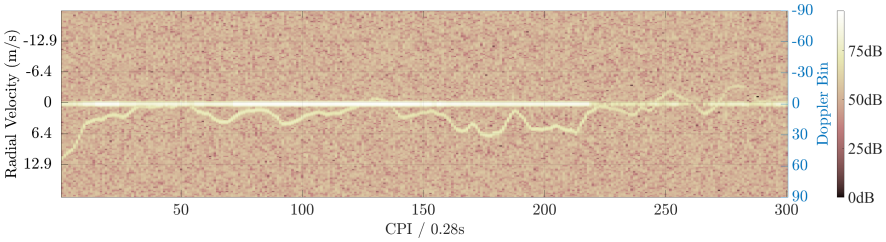


(b) Histogram of ranges of birds with collected drones

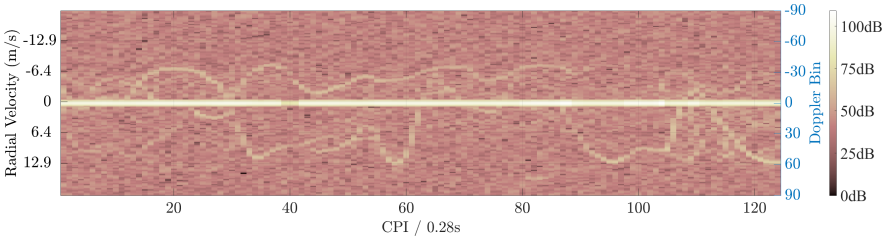


(c) Histogram of heights AMSL of birds with collected drones

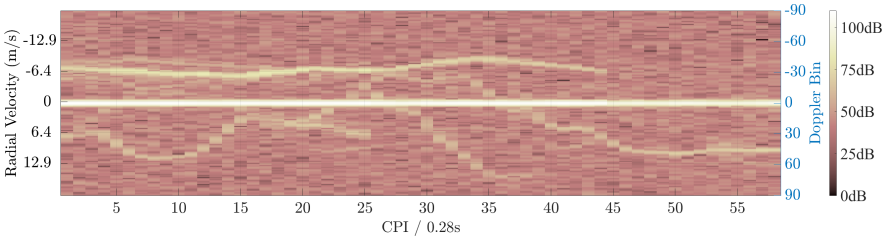
Figure 4.12: Plots summarising locations of opportune bird data collected.



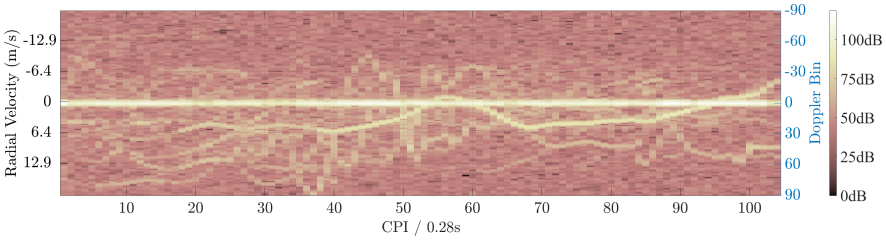
(a) Single Bird.



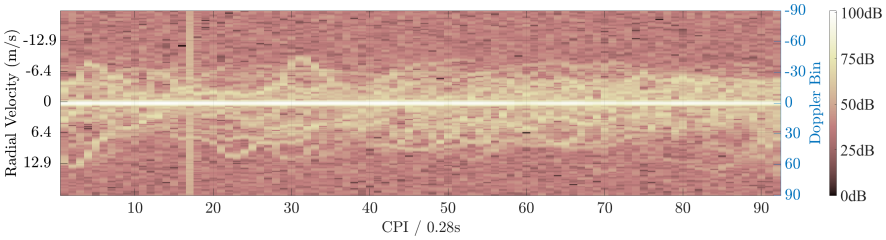
(b) Pair of Birds.



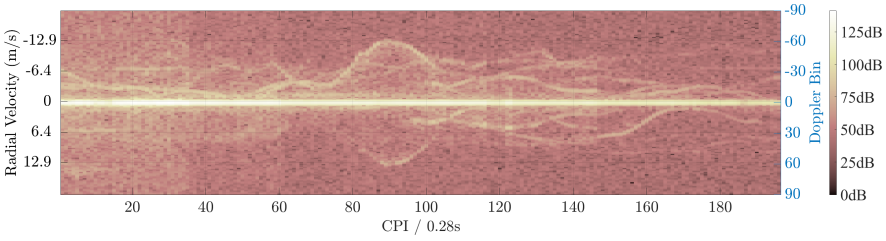
(c) Trio of Birds.



(d) Small Flock of Birds.

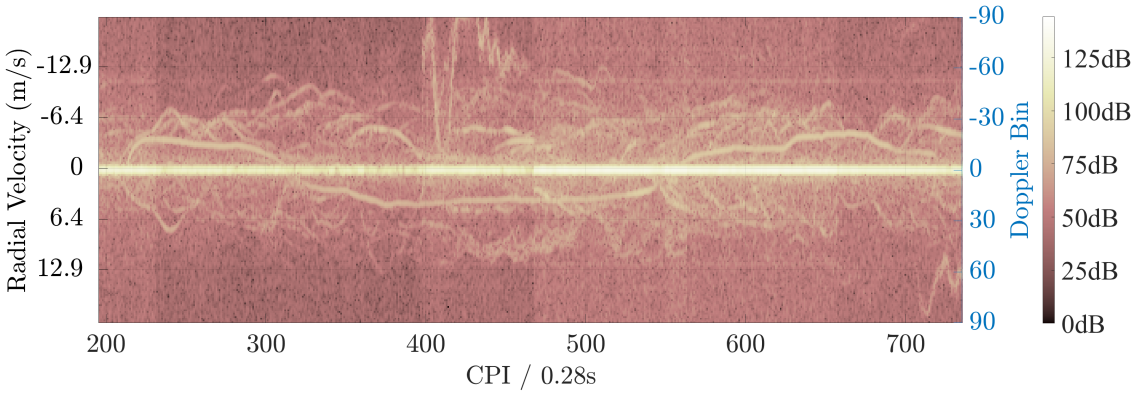


(e) Larger Flock of Birds

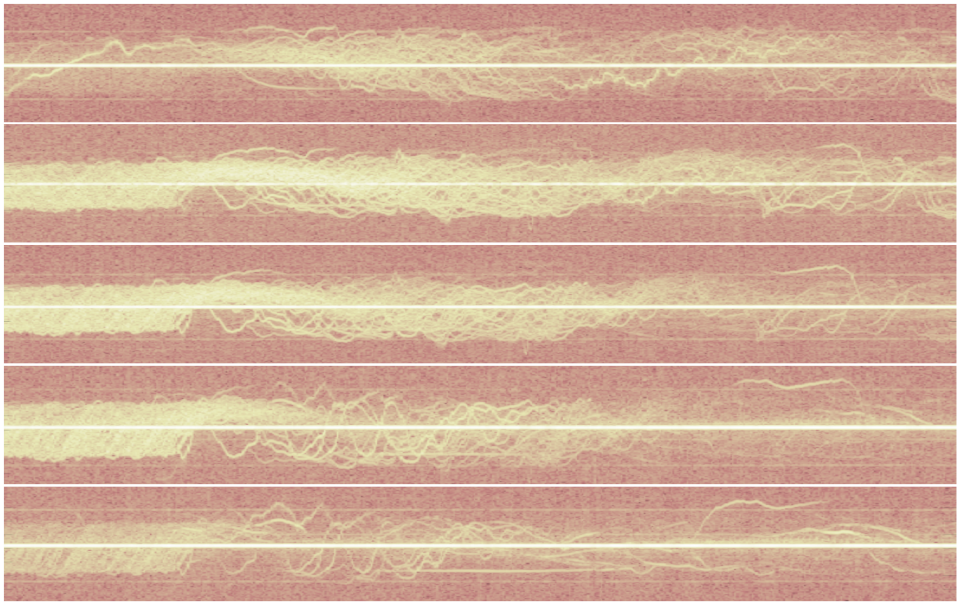


(f) Several birds entering and leaving the tracked resolution cells.

Figure 4.13: A selection of opportune bird spectra observed with Radar#1.



(a) *Mini* drone amongst flock.



(b) Five adjacent range resolution cells equating $\sim 400\,m$ capturing 150 seconds on flock motion. Bottom is closest to the radar.

Figure 4.14: Gull flocking observed with Radar#1.

On a particular drone trial, a great flock of about 50 Lesser Black-backed Gulls with minor numbers of other species engaged in a 'feeding frenzy', encircling an open field that was observed by the members at the trial site. The radar spectral returns collected during this observation are shown in Figure 4.14. The *Mini* drone was flown in the vicinity of this flock of birds, which resulted in Fig. 4.14a that would be quite challenging to recognise as a drone,

especially if only a small snippet of time was observed. These birds after circling the site for a period of time were noted to slowly leave the location en mass. Fig. 4.14b shows time-Doppler spectrograms formed over five range resolution cells from where the event was geographically noted. On the left, the dense mass of birds are seen circling together very closely, before a great dispersal of the targets out of the selected resolution cells.

4.3.4 Opportune Labelling

Manual human labelling of the tracks were done for some specific investigations. A single track can contain potentially a great variety of returns which made it difficult to label entire tracks without a degree of presence of other targets. Labelling for machine learning done in this work is at the track level, which carries semantic uncertainty and so was only used for a few specific studies. [3] used manual sub-labelling to refine birds into single & multiple, and drones into isolated & accompanied by birds. This study investigated the effect of background bird presence on the classificability of small, μDoppler less drone targets in the realistic setting of an urban environment. As observed in the spectrograms of this chapter, large resolution cells causes radar returns to be prone to containing returns of multiple targets. The drone returns collected frequently featured bird presence that presents a challenge to a binary classifier to detect drone presence in isolation and amongst other targets. The deep learning investigation with this data takes place in Sec. 5.2.4.

In a separate study shown later in Chapter 8, a large batch of opportune bird data was labelled as a 'Single', 'Flock' or 'Unspecified' to describe the volume of bird presence to be used to illustrate the performance of unsupervised learning algorithms. The labelling of entire tracks brought a great deal of label noise as the fluctuation of returns in the spectrogram was likely great for a long duration track. The data was cropped between two points to maximise correct labelling of a single track, but this increased the labour required significantly. Section 8.2 will show the potential of unsupervised techniques as a labelling tool that will better allow the occurrence of bird flocking to be featured more precisely, thereby reducing false positives and misclassification

of drone targets, where the manual labelling approach is useful to immediately check the general distribution learned. It will argued as part of the assessment with unsupervised approaches that there is perhaps justification to state that the machine learning methods could be more accurate than human labelling for bird flocks (see Figure 8.13).

4.4 Collected Data Summary

Data from the drone, control and opportune trials described in this chapter were used for training, testing and validation of the deep learning classifiers in the remainder of the thesis. The data from the urban environment produced spectrograms of the target that were not without impact from the strong clutter present. Only the Cranfield Bird of Prey trial used featured returns from a semi-rural environment, and the amount of data collected from these trials was far less than from all other trials. Selected targets from the control bird trials where the target was visible for the duration of the recording, were included in the test set since their value as a genuine target was more in the test set than amongst the opportune birds in the training set. The overall quantities of data collected across all trials that were used in at least one of the deep learning datasets in later work are shown in Table 4.2 below. The table shows the number of flights (GPS recordings or radar tracks), the total number of spectra, the amount of time equivalent to this, as well as how many 20-CPI spectrogram samples were created that was consistently used as the input for all experiments in this work (see Figure 5.2 in Chapter 5). The Pigeon data was present in the table but is hidden within the opportune birds row.

Table 4.2: Overall quantities of collected data

Target	Flights	Spectra	Hours	20-CPI Samples
Drones	572	318,895	24.7	10,063
Matrice	116	80,136	1.1	3,948
Inspire	234	129,045	10.0	6,345
Mavic	42	14,473	1.1	703
Mini	172	90,820	7	4,455
Birds	932	251,919	19.6	23,784
Cranfield Birds of Prey	4	3,955	0.3	393
Opportune Birds	924	247,964	19.3	23,391

Chapter 5

Supervised Spectrogram Classification

This chapter will review and explore supervised deep learning for drone and bird classification, by using a Convolutional Neural Network (CNN) transfer learned from the optical domain, on the urban drone and bird dataset collected as described in the previous chapter. The first section will review selected existing works using deep learning for drone, and drone against bird classification. Earlier in Sec. 3.3.3, the use of deep learning in radar for a wide range of target classification applications was reviewed. The review in this chapter will specify further details of the studies that feature drone and bird targets in a variety of configurations, including details of their radar, environment, targets, data quantities and data processing as relevant to the original work in this chapter. The CNN classifier architecture is based on previous published work by the UoB group [124], [253], [260] and results are obtained on a new dataset collected with the experimental systems of the current work. A focused review of classification of real drone and bird spectrograms where a transfer learning approach has been used is presented in Section 5.1. Section 5.2 following this will explain the model, data, processing used, classification approach taken and finally, results of supervised drone and bird classification with commentary of their interpretation and significance. The classification output with the baseline CNN model are generated with three datasets of different target configurations and are presented; in Section 5.2.3: large drones and birds, Section 5.2.4: small drones and birds and Section 5.2.5: all drone

sizes and birds. The individual results presented in these show the effect of including different combinations of targets in the dataset. The results in this section also serve as a benchmarking of an established CNN for classification for various radar bird and drone targets to be expanded upon in later chapters.

5.1 Deep Learning Classification with Radar Bird-Drone Signatures

It has been previously demonstrated that supervised deep learning with CNNs is an effective classifier choice of radar returns for a host of applications [261]. Studies using the CNN and supervised learning have also been performed on bird and drone targets. This section will review in detail work that uses the CNN or similar approaches for drone target classification and concludes or discusses the general findings from each. This section is only concerning demonstrations with 100% real targets. Each piece of literature takes their own unique approach towards achieving the results in their publications, and any nuances will be highlighted as relevant to the findings of this project to be shared in Section 5.2.

5.1.1 Review of Drone and Bird Radar Classification

In the following, 4 key references using CNNs or other deep learning methods on real bird and drone radar data are scrutinised ahead of this project's original study following in Sec. 5.2.3-5.2.5 benchmarking the performance of a CNN on the collected real, varied urban environment dataset in a variety of target configuration perspectives. These following citations all contain the returns of real bird data and the collection of such data can be hugely challenging as touched on in the data gathering Section 4.2. The different publications go into different levels of detail of the bird collection process which are reported in the discussion.

- [94] - Rahman and Robertson *"Classification of drones and birds using convolutional neural networks applied to radar micro-Doppler spectrogram images"* (2020)

Table 5.1: Radar and RNN classification approach reported in [94]

Variable	***Value***
Radar System(s)	94 GHz FMCW *and* 24 GHz FMCW
Total Samples	2400
Sample / Input	2s STFT dwell
Class Labels	Bird vs Drone Bird vs Drone vs Clutter vs Noise
Drone Targets	DJI Phantom 3, Inspire 1, S900 Hexacopter
Bird Targets	Northern Hawk Owl, Harris Hawk, Indian Eagle Owl, Tawny Eagle
NN Model	Optically Pretrained GoogLeNet CNN *and* Custom 1-Channel (8Conv+4FC) 'Series Network'
Optimiser	SGDm
Reported Accuracies	2-Class GoogLeNet = 99.77% 4-Class, GoogLeNet = 99.01% 2-Class, Series Network = 98.2% 4-Class, Series Network = 96.2%

This paper provides a benchmark for 16 different radar and classifier configurations. For their fixed drone and bird dataset, they repeated the experiment for two radar bands (see Table 5.1), with the optically pre-trained RBG input classifier as well as a single channel, custom network trained from scratch. The convolutional layers in the optically pre-trained model were frozen when fine tuning with radar data to prevent overfitting of extracted features with the limited training set, which was done in this and other work. They considered two-class labelling of bird vs drone, as well as 4-class with the inclusion of confuser classes of clutter and noise backgrounds, which consider ground and airborne returns with no co-located target. The

2-class problems yielded a better overall classification accuracy by 2% for the custom, not pre-trained, single-channel CNN and by 0.76% for the optically transfer learned GoogLeNet with 3 channels. The inclusion of the extra classes reduced the overall accuracy value reported, but the accuracies reported here are only for the bird and drone target accuracies, and does not consider missed-negative classifications, such as a bird classified as clutter. Their key result that optical pre-training allows a very high (+99%) accuracy result, compared to from-scratch training that was behind by 2-4%, shows that optical features are able to classify drone and bird Doppler signatures to a very high degree of performance. Chapter 8 investigates alternative pre-training approaches not based on optical datasets, but this chapter extends the findings of these results with the use of a radar that is operable over much longer ranges, uses more training and test data, and is based in an urban area of operation with increased noise and artefacts from environmental sources.

- [124] - Dale, Baker, Antoniou, *et al.* *"SNR-dependent drone classification using convolutional neural networks"* (2022)

Table 5.2: Radar and CNN classification experiment reported in [124]

Variable	***Value***
Radar System(s)	1.25 GHz Pulse-Doppler Staring Radar
Total Samples	10,551
Sample / Input	20 CPI / 6 second spectrogram + Injected Noise
Class Labels	Bird vs Drone
Drone Targets	DJI Inspire 1
Bird Targets	Opportune Birds from Radar Tracker
NN Model	Six ImageNet pre-trained CNNs: AlexNet, Squeezenet, GoogLeNet, Resnet-18, Resnet-50, Inception-v3
Optimiser	Stochastic Gradient Descent with Momentum
Accuracy	At 44dB = All CNNs ~ 98% At 32dB = 96, 95, 94, 94, 90, 88 % At 20dB = 83, 70, 64, 53, 52, 51 %

The work by Dale *et al.* features datasets that are very similar to the application on this work. They used an L-Band staring radar to gather real returns of a range of drone and bird targets. In [124], a study systematically comparing different common optically pre-trained classifiers with variation of noise injection was conducted. The study showed the effect of SNR degradation individually for the training and test sets, and showed that the recall of birds fell first while the accuracy of drone targets remained high. Their result that compared six different optically pretrained models, shown in Table 5.2, showed that AlexNet performed by far the best with diminishing test and training set SNR, with Resnet-18, -50 and Inception-v3 architectures

falling to 50% accuracy with the greatest amount of noise added. These three architectures were amongst the deepest, so increasing the limited fine-tuning radar data size could possibly yield different results. These three architectures also used either residual (skipped-connection) layers, or for the inception model filter concatenation, which is a somewhat similar process. The work of this thesis, as it used the same radar and similar targets as the study [124], also used AlexNet as it was shown to be the most robust with fluctuations to test and train set SNR.

- [250] - Karlsson, Jansson, and Hamalainen *"Model-Aided Drone Classification Using Convolutional Neural Networks"* (2022)

Table 5.3: Radar and CNN classification experiment reported in [250]

Variable	***Value***
Radar System(s)	77 GHz FMCW
Total Samples	120,000
Sample / Input	Single Spectra, Injected Noise
Class Labels	6 Drone Models vs Humans vs Birds vs Corner Reflectors
Drone Targets	DJI Matrice 200, Mavic 2 Pro, Phantom 3; Syma X23W, Custom FPV drone and another Custom Tarot 680 drone
Bird Targets	Opportune Birds Baited by Operator
NN Model	Custom 1D CNN (3 Conv. + 2 FC)
Optimiser	AdaM
Accuracy	SNR >17.5dB All Targets: ~ 95% SNR >17.5dB Unseen Drones: ~ 82% SNR >17.5dB With Synthetic Drones: ~ 91%

The work summarised in Table 5.3 for [250] conducts an experiment at short range with a low-cost FMCW radar system of a wide range of drone targets, as well as real birds that were lured into the radar field of view via food scattered by the experimenters. Five drone models, including two custom-built drones, were classified against birds, humans and stationary corner-cube reflectors. They also used a synthetic data approach to bolster the dataset against unseen drone targets, that will be touched upon further in Sec. 7.1.1. Their study without synthetic data showed that a small CNN with three 1-D convolutional layers and a single hidden layer on single spectra was able to classify the real birds and drone models to 95% accuracy.

- [127] - Duan, Zhang, Wu, *et al. "Classification of birds and drones by exploiting periodical motions in Doppler spectrum series"* (2023)

Table 5.4: Radar and NN classification approach reported in [127]

Variable	***Value***
Radar System(s)	L-Band 1.36 GHz
Total Samples	Range 507 to 1562
Sample / Input	4-32 Pulses
Class Labels	Drone vs Bird
Drone Targets	DJI Phantom
Bird Targets	Opportune
CNN Model	LSTM Spectrogram Input, and Processed Feature LSTM Input
Optimiser	Not Specified
Reported Accuracies	*LSTM* (4, 8, 12, 16, 20, 24, 28, 32 Frames) = (94.5, 85.2, 81.3, 89.0, 93.7, 88.6, 84.2, 100) % *LSTM+Feature Input* (4, 8, 12, 16, 20, 24, 28, 32 Frames) (93.4, 94.5, 91.7, 94.4, 98.8, 97.8, 97.9, 100) %

Unlike the other classifiers used in the previous listed references, [127] summarised in Table 5.4 uses a bi-directional LSTM recurrent NN to classify a collection of multi-frame bird and drone spectrograms. They compare three approaches: a hand-crafted spectral feature extractor with Naïve Bayes classifier, an end-to-end LSTM classifier, and the hand-crafted features used as input to the LSTM classifier. The authors also used 8 versions of the same dataset with

different dwell lengths, up by multiples of 4 to 32 frames per sample used, which impacted on the amount of data present for each run. It was also found that the classifier was biased towards birds using the LSTM alone, but with the Feature + LSTM process, the bias was oppositely towards drones. This paper also uses L-band radar data with targets collected from a range of 1-4 km, and so serves as a useful reference that demonstrates the capability of deep learning classifiers for bird and drone targets with a recurrent neural network. Another very recent paper from 2023 also using L-band bird and drone data was [76], that achieved 98% accuracy on a test set of 1,611 samples without transfer learning. Their approach used a 'robust coordinate attention synergy residual split-attention' NN that was able to outperform a host of existing CNN architectures without any transfer learning. Their 'RCA-ResNeSt' classifier trained in one stage in a supervised manner gained good performance with diminishing SNR, showing that optical transfer learning is not essential to building a robust classifier. That stated, their work does not compare optical transfer learning with their proposed approach, but it does highlight that the modern AI trends (in this case, self-attention, that is behind the transformer mechanism) can be applied to bird and drone data to perform a reasonable classification, utilising well any target features. In this thesis, we are working on classifier architectures using data collected within a real urban environment which will inform expected performance levels.

Table 5.5 summarises a collection of recent literature each performing classification with deep learning of birds and drones. Other works undertaken looking at the classification problem with *simulated* drone returns are also of interest, and will be detailed in Chapters 6 and 7, focusing on drone simulation, and aiding classification with synthetic data respectively. A question that arises within drone identification is the granularity of labels that they may be assigned to. The classification of specific drones by their make and models using a multi-label CNN classifier has been conducted in several studies with the use of real drone data [80], [121], [250] and with simulated drones in [129]. Previously to these, studies with synthetic data have successfully used deep [92] and machine [255] learning to classify drones by the number of blades and

Table 5.5: Selection of recent radar literature utilising deep learning for classification using real bird and drone targets. *-Indicates work originating from this thesis. TL, Optically Transfer Learned.

Cit.	Classes	Band	Environment	Algorithm	μDoppler	Performance
[4]*	Bird-Drone	L	Urban	Fine-tuned CAE	All Drones	98.69
[4]*					Drones With and Without	92.9
Sec. 5.2.3*				TL AlexNet	All Drones	98.6
Sec. 5.2.4*					Drones With and Without	89.6
Sec. 5.2.5*					Without	90.7
[124]			Urban Rural Mix		All Drones	99.0
[251]					Drones With and Without	93.5
[252]			Semi-Rural		All Drones	98.5
[127]			Rural	Bi-directional LSTM	Minor Drone	100
[76]	3 Drone 2 Bird		Semi-maritime	Split-Att. Res. CNN	Mixed Drones	98.0
[152]	Bird-Drone	Ku		Inter-domain fusion network	Bird-Drone	96.4
[94]		K & W		TL GoogLeNet		99.5
[250]	6 Drones Birds Humans	W		Custom CNN		91.5

motors present, which as shown in Chapter 2 is highly dependent on the radar system's sampling rate and Doppler resolution. Another work [93] used the concept of small, medium and large drones to provide a more reasonable labelling system than arbitrary drone model names. In the author's opinion, the labelling scheme that could be worth pursuing in future work is by their weight. Different jurisdictions have different regulations relating to the drones' weight, and heavier drones bring increased severity of collisions with manned aircraft [262] and better payload carrying capabilities. Labelling drone targets by model name is a useful check for the classifiabilty of similar drone targets via a CNN. This approach is not particularly flexible to unseen drone models, but it is conceivable to have a classification label of 'unknown drone' alongside other recognisable models. Intra-drone model classification would be useful where known drone models are employed regularly, such as if one day routinely used at airports as workers. Similarly in this regard, estimations of payload mass have been done using deep learning approaches which is a very useful property for drone security applications. Patterns in drone μDoppler as a result of payloads (due to weight or momentum) were outlined in [88], [94], [118], [128] and classifiable features were derivable that supported accurate classification of single drone models. Again, different classification regimes in these consider if a payload is present as a binary choice, as a weight to be estimated, or as 'empty', 'light' or 'heavy'.

5.1.2 Introduction to AlexNet

The quantities of data generally used in radar are insufficient to totally train deep learning classifiers in a supervised manner, when they have been shown in the image and natural language domain to be required for excellent performance. Smaller classifier models have been used for discrimination of real avian targets from drones on both single spectra [119], [250] and multi-frame spectrograms [94] training only on drone and bird data. However, many works have used transfer learning from much larger classifiers trained on a huge amount of optical images, referred to as optical pre-training, before *post-training* or fine-tuning the model parameters with the radar datasets. Of the many available optical models for transfer learning, this work primarily uses AlexNet [249] which has 5 convolutional layers and 2 fully connected layers, followed by

usually 1000 categorical outputs to fit training on the ImageNet dataset. This model is trained in a supervised fashion with cross-entropy loss. This particular model was shown previously in [120], (using the same radar system, signal processing and similar targets as used in this report) that it performs the best under diminishing SNR compared to other similar, large, ImageNet pretrained models.

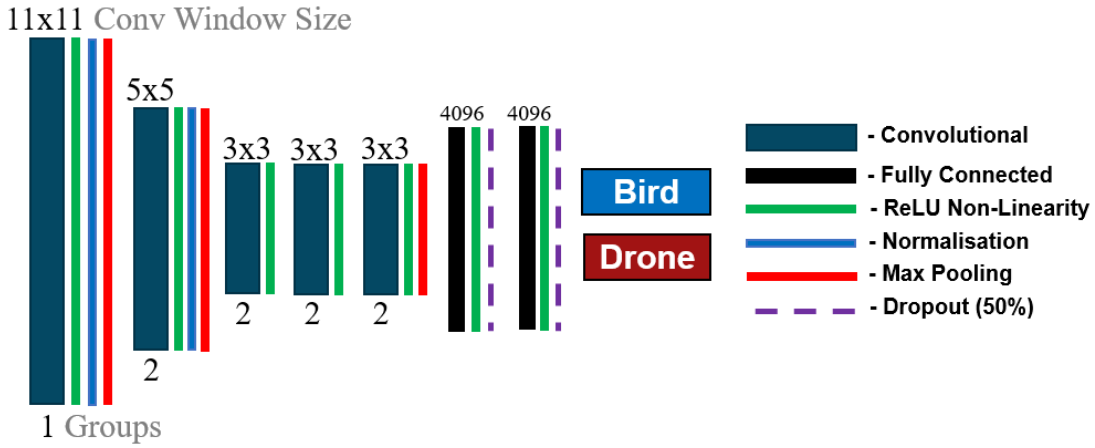


Figure 5.1: Schematic of AlexNet Layers.

The layer specification of AlexNet is shown in Figure 5.1, where its adaptation for Bird-Drone classification is implemented at the output of the fully connected layers. At the output of the convolutional layers, 9216 extracted features are weighted as evidence of bird or drone presence feeding into the prediction output at the class layer. Due to the dense optical features extracted from the convolutional layers, the weights of these layers are frozen. Then, when post-training on radar data, learning is restricted to the fully-connected layers, forcing the classifier to map the fixed, generic features to the bird and drone labels. AlexNet's square convolutional window sizes decrease after each layer, from 11→5→3→3→3, and the number of groups of convolutions (the number of convolution processes to occur in parallel) is 1→2→1→2→2. The single group in the 3rd layer recombines the features in the middle of the extraction process. Merging of the intermediate features consolidates the abstraction learned in the optical pre-training with improved performance, which is justified in the case of a huge amount of classes, where feature generality and efficiency is imperative. The number of channels outputted from each convolutional layer are 96→128→192→192→128. AlexNet in Section 5.2 following was

constructed and trained using MatLab 2022b.

5.2 Supervised Learning with CNNs

This section performs a benchmarking of L-band staring radar data with supervised training of the optically pre-trained AlexNet CNN for drone classification. The resultant classifier will be referred to as the AlexNet (pre-trained) Convolutional Neural Network Classifier ($\mathcal{A}$CNN). Details of the dataset and the training process will be shared in Sec. 5.2.1 and Sec. 5.2.2. Then, Sec. 5.2.3 - Sec. 5.2.5 presents benchmark performances of real data for different target configurations. Finally, Sec. 5.3 will briefly summarise the efforts of the chapter leading into what is to follow.

5.2.1 Dataset Curation

Chapter 4 explained the trial procedure and drone targets used, as well as the process for collecting opportune bird targets. The processing of raw recorded radar data into target spectrograms was explained and justified previously in Sec. 2.3.2. This chapter uses a fixed target dwell time across all experiments and datasets. GPS or tracker based collected spectrograms were divided into samples of 20 consecutive, overlapping CPIs, in total lasting 5.9 seconds, which allows the targets' dynamics over time to be processed. Figure 5.2 shows this with five bird and five drone samples as used as input to the classifier. This data processing approach was also used in [120], [124], [251] as well as in each of the publications to come out of this work concerning deep learning [2]–[5]. The training subsets of data were then partitioned such that data from a given flight was not split across train, test or validation subsets. To achieve this, the number of frames and images per flight was calculated, and balanced train, test and validation subsets with an approximate 65:35:10 ratio were formed at random using this information. The quantities of data for each set for each target configuration are shown in the following sections alongside the results achieved.

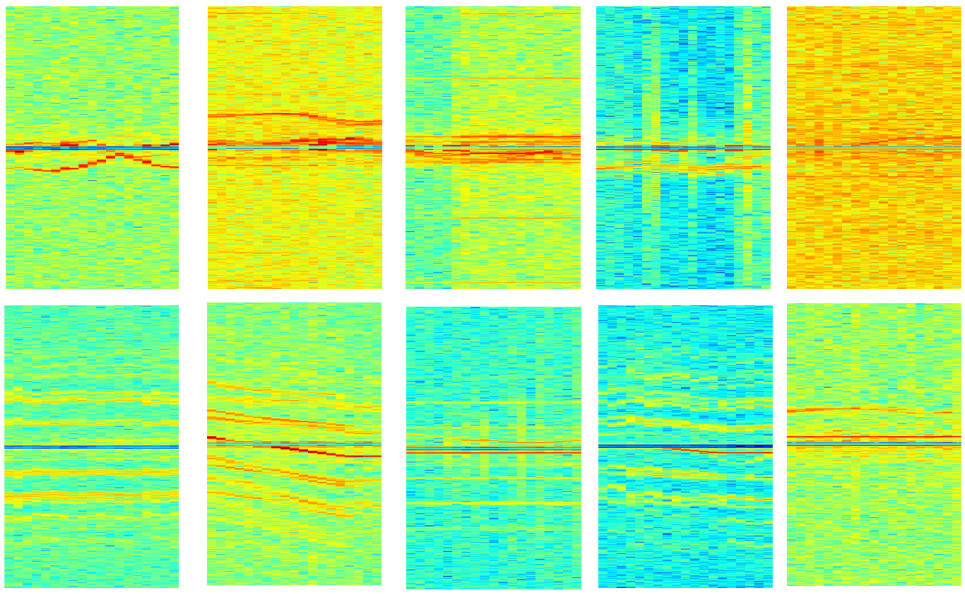


Figure 5.2: False colour drone and bird samples as input to the classifiers of this chapter. Birds are in the top row and drones are in bottom row which are all from large drones, except for the bottom right which is of the *Mini* drone.

5.2.2 Classifier Training Approach

To report results achieved by a CNN, its training must be optimal and free of severe overfitting. Training was stopped throughout this work when the summed training and validation accuracy was maximal, after either a fixed number of epochs, or after the training accuracy had exceeded the validation accuracy for four epochs. The fine-tuning with radar data varied slightly from study to study in the remainder of this section. The topic of hyper-parameter selection is challenging and their optimisation is an extensive open topic [263]. Iteration and testing hyperparameters repeatedly as an optimisation challenge may be aided by open libraries [264]. Bayesian optimisation [265] of the hyper-parameters: batch size, learning rate, L2-regularisation factor, and optimiser (SGDm) momentum, as used in Sec. 5.2.5 was also utilised in [124]. Otherwise in Sec. 5.2.3 and 5.2.4, parameters were used as had been previously used in [251] with the learning rate scanned for the best performance, first doing a coarse and then a fine scan (see further Appendix A).

The general approach of work this thesis is to find methods that can contribute to classifications

of drones across all radar types. The drone HElicopter Rotor Modulation (HERM) lines are a common representation, that can be observed in all coherent radar systems depending somewhat on the drone's range and blade size. With the L-band staring radar system used in this work, small drones (DJI *Mini* 2 and *Mavic* 3) did not produce any significant μDoppler as their blades were about a third of the radiation wavelength. At higher frequencies, practically all drones will produce μDoppler. The following section shares preliminary benchmark results for optically pretrained AlexNet. The AlexNet CNN classifier after transfer learning to the radar domain, $\mathcal{A}$CNN, here is trained and tested with three different datasets of target configurations: large drones with μDoppler (*Inspire* and *Matrice*), small drones without μDoppler (*Mini* and *Mavic*), and a mixture of large and small drones, all compared to a random sample of opportune collected bird signatures. The nomenclature used to refer to these configurations will be

- Large Drones and Birds (LDB)
- Small Drones and Birds (SDB)
- All Drones and Birds (ADB)

5.2.3 Bench-marking Large Drones against Birds

The μDoppler from large drones are an important feature to enable reliable classification of drones in the majority of cases. Even for drones with a protective guard surrounding the motor blades, a flying drone's orientation or tilt will vary significantly over a flight as wind is compensated and momentum is transferred from the motors to the drone, which will present an opportunity for μDoppler to be measured. Thus, the simple but significant classification test of drones with μDoppler against birds will be assessed with the $\mathcal{A}$CNN in the first instance. Table 5.6 shows the targets and their quantities used in each subset. The AdaM optimiser using 10^{-6} L2-Regularisation with epsilon value set to 10^{-8} was used with a batch size of 32. Then, after a course sweep of learning rate order of magnitude, 8 learning rates between 6×10^{-5} and 6×10^{-3} were used and each was repeated twice with different initialisation seeds. Of these,

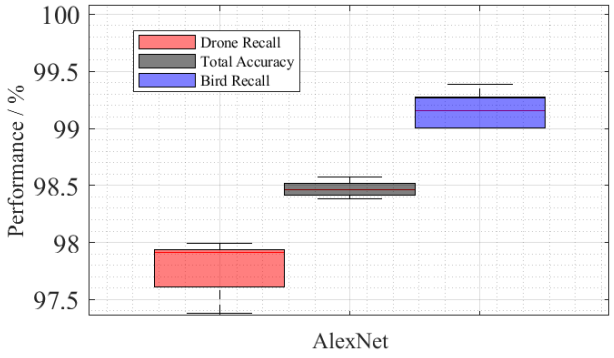
Table 5.6: Composition of Large Drones and Birds (LDB) data subsets

Target	Train Samples	Test Samples	Validation Samples
Bird	2572	1303	421
Inspire	1558	618	200
Matrice	1012	676	219
Total	5142 (2570 + 2572)	2597 (1294 + 1303)	840 (419 + 421)

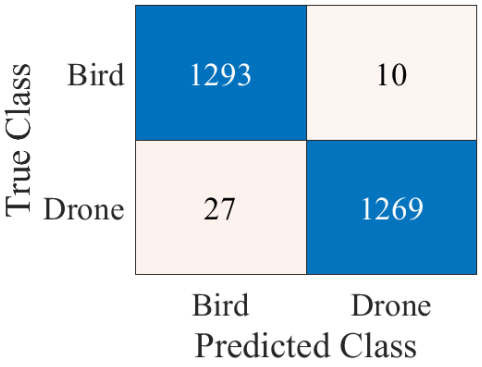
Fig. 5.3a shows in the form of a standard boxplot the top 5 models' accuracy and class recall displayed either side with drone on the left in red and bird on the right in blue. Fig 5.3b displays the confusion matrix for the classifier with the highest accuracy.

Figure 5.3: Results for Large Drones and Birds (LDB) dataset with $\mathcal{A}$CNN.

(a) Recall Boxplots for best 5 accuracies.



(b) Confusion matrix for best accuracy.



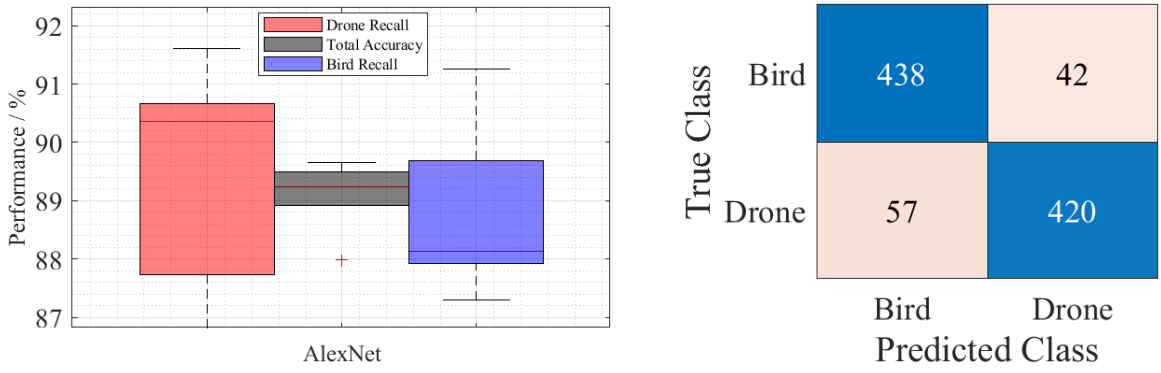
Accuracies at 98.5% were achieved, which agrees with others that also achieved near 99% with targets possessing μDoppler [94], [252] using optical transfer learning. For the best classifier and out of the 2597 test samples, 10 birds were misclassified and 27 drones were misclassified. The drone's probability of detection therefore was 0.98, and the probability of false alarm was 0.008. The recall, that represents reliability of classification for that label, shows that drones are more frequently classified as birds, than birds classified as drones. Here, every classifier was biased towards birds by ~0.5% at minimum. This means the resultant classifier is going to have a very small number of drone false alarms, but a slightly increased number of missed detections. The recall is a key property of classifiers that will be closely examined in subsequent experiments.

Table 5.7: Composition of Small Drones and Birds (SDB) data subsets

Target	Train Samples	Test Samples	Validation Samples
Bird	814	480	29
Mini	654	317	21
Mavic	153	160	14
Total	1621 (807 + 814)	957 (477 + 480)	63 (35 + 29)

5.2.4 Bench-marking Small Drones against Birds

In contrast, drone targets that do not possess μDoppler are of more challenge to spectral classifiers, as a major contrasting feature is not present. As is the case for two drones in our fleet, all radars will have ranges and targets where the μDoppler is absent. Similar to Section 5.2.3, a dataset was constructed that this time contained only small, μDoppler-less drone targets, the details of which are shown in Table 5.7. The AdaM optimizer was again used with (squared) gradient decay factors of (0.999) 0.9 with 10^{-5} L2-regularization, and suitable training parameters were found with a grid search yielding an initial learn rate of 10^{-4} and mini-batch size of 32. Training was repeated nine times with different initialization seeds with the results presented in Figure 5.4 with the boxplot summary of the best 5 accuracies in Fig. 5.4a and the best confusion matrix in Fig. 5.4b.



(a) Recall Boxplots for best 5 accuracies.

(b) Confusion matrix for best accuracy.

Figure 5.4: Results for Small Drones and Birds (SDB) dataset with $\mathcal{A}$CNN.

The accuracy in this case with a much more similar nature of returns of many samples of the

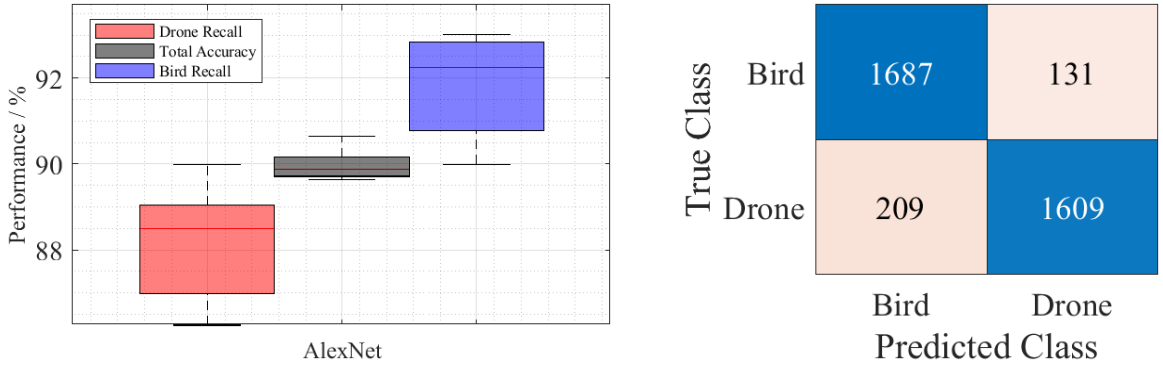
Table 5.8: Composition of All Drones and Birds (ADB) data subsets

Target	Train Samples	Test Samples	Validation Samples
Bird	3625	1818	371
Inspire	1444	612	127
Matrice	1058	671	170
Mini	1129	535	45
Total	7256 (3631 + 3625)	3636 (1818 + 1818)	713 (342 + 371)

two classes, was around 89.5%. The convolutional features of $\mathcal{A}$CNN are able to detect the presence of a drone, in many cases amongst bird presence. The ~6 second dwell of samples enabled distinguishing features to be gathered beyond μDoppler, perhaps thanks to differences in target velocity pattern and spectral width. The classifier balance was less consistently biased towards birds, and depended in this case more on the initialisation seed rather than training process. The drone's probability of detection therefore in this case was 0.88, and the probability of false alarm was 0.09. The results in this small drone dedicated section are a subset of that which was reported in [2].

5.2.5 Bench-marking All Drones against Birds

The same classifier was used to classify a dataset containing both large and small drone targets, further challenging the model to learn features with, and in excess of μDoppler. The use of all drone targets allowed a fairly large dataset to be created with ~3,600 training samples per class, using the same ~6 second long dwell per sample, with target quantities shown in Table 5.8. The training parameters that were used to fine-tune the model for radar data are as follows. Bayesian optimisation [266] was used to optimise the parameters of the initial learning rate (1.5×10^{-4}), mini-batch size (16), SGD-momentum term (0.98) and L2 Regularisation term (2×10^{-4}) for the training of real data. A 0.1 factor decay on the learning rate every 10 epochs was used for the experiment, and also within the Bayesian optimisation stage. Training was repeated 9 times with different random initialisation schemes, and the results are shown in the same fashion as before in Figure 5.3.



(a) Recall Boxplots for best 5 accuracies.

(b) Confusion matrix for best accuracy.

Figure 5.5: Results for Large Drones and Birds (LDB) dataset with $\mathcal{A}$CNN.

The accuracy gathered for this dataset was of the order of 90%, which is significantly worse than the 98.5% gained for the LDB dataset, and ~1% better than the (much smaller in quantity) small drones only dataset. With about $\frac{1}{3}$ of the dataset being a small, μDoppler-less target in the training and test sets, the drone discriminating features would be not solely reliant on μDoppler, and this is shown true as the accuracy of the small drone class in isolation was 73.3%, and the large drones accuracy in isolation was 94.0% (see Table 7.3 in Sec. 7.2.2 and also [3]). The reliability of classification of small drone models decreased significantly by 14.7% when classified in this context compared to the dedicated small drone classifier in the previous section. It was again found that the bird recall was usually greater than the drone recall, which it is tempting to attribute to the greater amount of label variation from large and small drones, but as also seen for the LDB dataset in Sec 5.2.3 that featured no small drone targets, the same similar bias towards birds was observed. The overall trend could be due to the great variety of bird targets that were collected in an opportune fashion by the L-band staring radar, that was shown in Chapter 4 and [5]. For this ADB dataset, the drone's probability of detection this time was 0.89, and the probability of false alarm was 0.075.

Table 5.9: Results for the three datasets using $\mathcal{A}$CNN in Section 5.2

Training Dataset	Best Accuracy	Mean Accuracy of All Repeats (%)
Large Drones and Birds (LDB)	98.6	98.5
Small Drones and Birds (SDB)	89.6	89.2
All Drones and Birds (ADB)	90.7	89.8

5.3 Chapter Summary

The results of this chapter are summarised in Table 5.9, and were generated on an extremely difficult radar classification challenge. The urban environment, along with the extent and variety of bird and drone targets featuring returns from up to 4km range, ensured that results gained are representative of a real system where hand-crafted features would not be able to fit as well the variation of targets and returns that were gathered. The task was made even more difficult by the use of two different radar systems that collected data to a joint dataset as was shown and discussed in Sec. 4.1.4. The same DSP being used for the returns from the two radar systems ensured a great variation in target SNR, spectral artifacts and clutter profile within the same dataset, that is unarguably much greater than would be expected of any one single radar system operating with proper maintenance. With this in mind, the use of deep learning for security critical applications requires that the approach is robust to changes in expected returns, and degrades gracefully with unseen or unexpected targets and conditions.

In this chapter, classification performance was obtained using $\mathcal{A}$CNN using LDB, SDB and ADB datasets representing varying degrees of separability for drone and bird targets. This work has provided corroboration for other work that used a transfer learned CNN to achieve accuracies at above 98% where μDoppler producing drone targets are exclusively present for the LDB dataset. Without any μDoppler, targets were classified to just under 90% which reflects the difficulty of the challenge without this feature, while just over 90% was achieved for the ADB, that had drones with and without μDoppler. The LDB and ADB sets were consistently biased to select edge cases as birds, while the SDB set was less consistent but the majority were biased towards drones. The SDB set was trained on only 1,600 samples compared to the others

that were over 5000, so greater quantities of data may revert this trend. The SDB classifier had the difficult challenge of classifying single birds, single drones and flocks of birds *and* drones amongst other bird presence *without* the feature of rotor μDoppler to aid the classification. The bias therefore could come from the lack of drone features for classification and hence a much greater reactiveness to possible drone classification. This is contrasting to the ADB and LDB cases where the defining drone feature is more obvious, and so its absence made the classifier estimate difficult targets to be birds.

These results have shown that transfer learned CNNs are able to classify various targets for long range detection in a challenging urban environment. Results as presented in this section could be marginally improved with more rigorous optimisation, simple data augmentation techniques such as rotating and flipping to increase quantities, or perhaps other classifier architectures aside from AlexNet. This research work aims to tackle the fundamental issue of the lack of available data that is required to fully optimise a classifier model for real world deployment. In the following Chapter 6, the creation of simulated drone data using new techniques that may support this pursuit is presented, and in Chapter 7 we will compare this against the $\mathcal{A}$CNN LDB and ADB benchmarks from this chapter. Increased quantities of data may allow classifiers to be built to a very high standard without transfer learning, but the quantities of optical images in the tens of millions makes the pre-training option attractive for fully training deep classifier models. Chapter 8 will explore a radar data only pre-training approach with comparison to the results acheived with transfer learning from this chapter.

Chapter 6

Creating Synthetic Drone Data

This chapter presents in detail a simulation approach for producing realistic drone Micro-Doppler (μDoppler). A simulation is useful to better understand the observations of drones in real radar systems, and if it can be made to closely match real observations then the possibility of creating synthetic data is available which can be a powerful tool used to fine-tune feature extraction algorithms or train deep learning classifiers with a reduced data collection burden. A mathematical framework for producing radar simulations allows an extremely fine control of created signals, which is an enabler of novel innovations, algorithms and ensuring robustness. This work expands on others' by introducing the motor speeds collected during the trials in Chapter 4 into the simulation procedure. The production of drones' distinct μDoppler is shown to be highly dependant on the choice of motor speeds simulated, that this work uses to create highly realistic drone μDoppler data to a fidelity not seen before in literature encountered. This chapter will define a basic simulation approach for a multi-rotor drone target, and the proposed method is used afterwards in Chapter 7 and Sec. 8.2.1 to create synthetic drone data to train CNNs that are tested on real data. We begin with Section 6.1 that derives an equation that approximates the returns of a rotating motor structure. Afterwards, Section 6.2 will convert the simulation into realistic synthetic data. Finally, Section 6.3 will further modulate the motor speed and return signal to transform the idealised, simulated data into realistic, synthetic spectrograms.

6.1 Simulation Approach

The radar signal processing and approximations of the signal-target EM interaction can be treated as a fully mathematical process, which is greatly simplified in an analytical fashion by employing some assumptions and approximations. Sec. 6.1.1 briefly reminds of the equation for a single point target originally introduced in Chapter 2. Sec. 6.1.2 derives a rotating blade structure from this as was originally pioneered in [267], and presents some options of scatter modelling that could be considered for use. Sec. 6.1.3 briefly extends this model to be applicable to multi-rotor drone targets, and explores the appearance of the scattering choices introduced in a drone spectrogram with real motor speeds.

6.1.1 Point Target Simulation

Starting with the demodulated I/Q signal of a point target that was derived in equation 2.2 in Sec. 2.2.1, a greater structure can be built by considering a network of moving point targets, which has been employed in many radar domains [268], [269]. This approach works generally with all coherent radar systems, however the details of radiation scattering from different targets will be related to the radars' operational wavelength and polarisation. It is possible to model the transmitted radiation as a complex sinusoid and process returns with In-phase/Quadrature (I/Q) demodulation. The return signal is a timeseries complex voltage, which if with no initial phase offset, a point target in time at radial range $r(t) = r(nT)$ will produce a baseband signal given by,

$$V_{I/Q}[t] \propto e^{-j\frac{4\pi}{\lambda}r(t)}, \tag{6.1}$$

where T is pulse repetition interval, n is pulse number, λ is operational wavelength and j is the imaginary unit. The proportionality here shows that the signal return strength is not being considered and will be maintained throughout the derivation, but is otherwise subject to the Radar Range Equation described in equation 2.7 in Sec. 2.1.3.

6.1.2 Helicopter Rotor Micro-Doppler Simulation

This section will derive a selection of equations that allow a modelling of the phase shift induced in a coherent radar signal for a single motor and rotating blade structure. The result of this derivation has been used to model helicopter scattering, and afterwards Sec. 6.1.3 will use multiple motors together to simulate the returns of a consumer drone. With the single motor model, superpositions can be crafted to recreate or mimic single and multi-rotor systems; however initially the derivation and considerations for the scattering of a single motor is presented to lay the foundation for more complex systems.

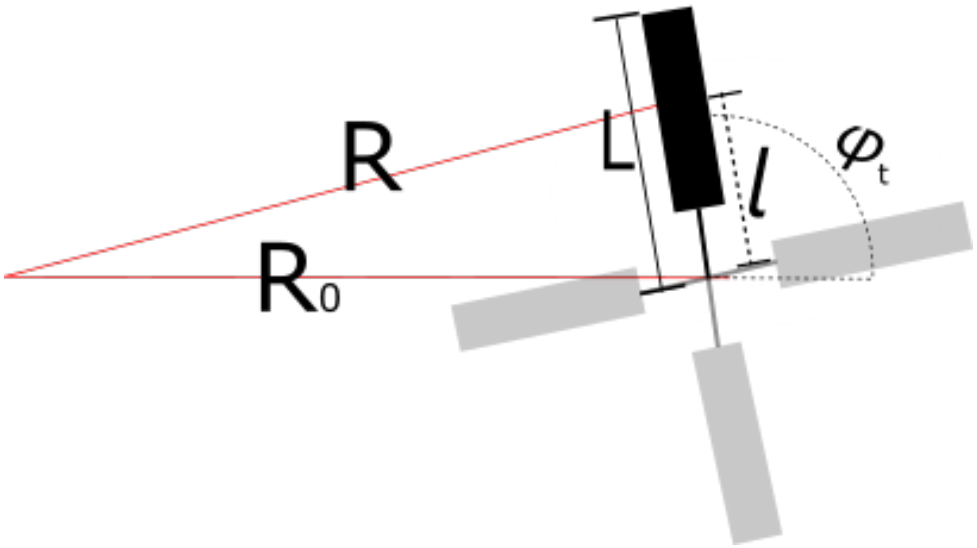


Figure 6.1: Top down view of a drone blade relative to radar position.

Figure 6.1 shows a drone blade at a distance from the radar with key measurements of R_0, the target's radial range to the radar, l, the distance from the motor hub to an infinitesimal length of the blade, L, the total length of the blade, R, the radial distance of the blade infinitesimal to the radar, and $\varphi(t) = \varphi_t = \varphi_0 + \Omega t$, where φ_0 is initial blade angular offset and Ω is motor speed. For a single rotating blade, using the cosine rule for a generic triangle $c^2 = a^2 + b^2 - 2abcos(C)$ to produce,

$$R(t)^2 = R_0^2 + l^2 - 2R_0 l cos(\pi - \varphi_t) \ , \tag{6.2}$$

which after solving for R and rearranging gives,

$$R(t) = R_0 \sqrt{1 + \frac{l^2}{R_0^2} + \frac{2l}{R_0} cos(\varphi_t)} \ , \tag{6.3}$$

which can be simplified using the fact that for drones, $\frac{l}{R_0} \ll 1$. First removing the squared term and Taylor expanding the square root to second order, $\sqrt{1+x} = 1 + \frac{x}{2} + O(x^2)$ produces the following simple equation for the radial position of a blade length infinitesimal, l, to be ,

$$R(t) = R_0 + l \, cos(\varphi_0 + \Omega t) \equiv R_0 + l \, cos(\varphi_t) \ , \tag{6.4}$$

and so a rotating point target with radius l yields the equation for baseband I/Q signal, by substitution into equation 6.1,

$$V_{I/Q}[t] \propto e^{-j\frac{4\pi}{\lambda}(\, R_0 + l \, cos(\varphi_0 + \Omega t)\,)} \ , \tag{6.5}$$

As a starting point, to model the entire blade itself as a point target, there is a choice to use the center of mass of reflection, $l = \frac{L}{2}$, but if implementing a Fourier transform over a time window shorter than a full rotation this will not reproduce the blade flashes at the anticipated velocity induced by a blade of size L observed in frequency-modulated continuous-wave spectrograms [87]. Using $l = L$ will rectify this particular issue, but cannot faithfully be used to model an accurate return. These two different blade models, $l = \frac{L}{2}$ and L will be respectively referred to as the Center of Mass Point target (CMP) and Blade Tip Point target (BTP) models.

Next, treating the blade as an integral sum over point targets along the blade is a conceptual improvement of the model, and has been used to model the μDoppler modulations caused by a helicopter's single rotor and multiple propeller blades [267], [270]. The spectral lines either side

of a helicopter's main body return, and are referred to as HElicopter Rotor Modulation (HERM) lines [87]. Integrating over the full length of the blade from motor center to tip at L, and using $\beta \equiv \frac{4\pi}{\lambda} cos(\varphi_t) \equiv \frac{4\pi}{\lambda} cos(\varphi_0 + \Omega t)$, gives,

$$\begin{aligned} e^{j\frac{4\pi}{\lambda}R_0} V_{I/Q}[t] &\propto \int_0^L \exp\{-j\beta l\}\, dl \\ &= \int_{-\frac{L}{2}}^{\frac{L}{2}} \exp\left\{-j\beta(u+\frac{L}{2})\right\} du \\ &= \frac{e^{-j\beta\frac{L}{2}}}{-j\beta}(e^{-j\beta\frac{L}{2}} - e^{+j\beta\frac{L}{2}}) \\ &= \frac{2}{\beta} e^{-j\beta\frac{L}{2}} sin\left(\frac{\beta L}{2}\right), \end{aligned} \tag{6.6}$$

where $u = l - \frac{L}{2}$ and $sin(x) \equiv \frac{1}{2j}(e^{jx} + e^{-jx})$. A different u-substitution can be used to shift the integral to be symmetrical around zero if a different blade root location is desired.

Continuing by multiplying the top and bottom by L, allows a further useful simplifcation of the equation,

$$\begin{aligned} e^{j\frac{4\pi}{\lambda}R_0} V_{I/Q}[t] &\propto \frac{L}{\frac{\beta L}{2}} e^{-j\beta\frac{L}{2}} sin\left(\frac{\beta L}{2}\right) = L e^{-j\beta\frac{L}{2}} sinc\left(\frac{\beta L}{2}\right) \\ &\equiv L \exp\left\{-j\frac{L}{2}\frac{4\pi}{\lambda} cos(\varphi_t)\right\} sinc\left(\frac{L}{2}\frac{4\pi}{\lambda} cos(\varphi_t)\right), \end{aligned} \tag{6.7}$$

where the penultimate step used $\frac{sin(x)}{x} \equiv sinc(x)$. This leaves the equation of the baseband return of a rotating thin wire blade to be,

$$V_{I/Q}[t] \propto L\, e^{-j\frac{4\pi}{\lambda}(R_0 + \frac{L}{2}cos(\varphi_t))}\, sinc(\frac{L}{2}\frac{4\pi}{\lambda} cos(\varphi_t)), \tag{6.8}$$

and will be referred to as the Thin Wire Model (TWM). To observe the three spectra and spectrograms generated by the three models (CMP, BTP and TWM) derived in this section, 2048 timeseries complex values were generated with a sampling rate of 7300 Hz, $\lambda = 24\,cm$and $L = 0.19\,m$. Fig. 6.2 shows the Fourier transformed timeseries signal of the CMP (top), BTP (middle) and TWM (bottom) which yields the characteristic helicopter spectral (HERM) lines for a single spectra.

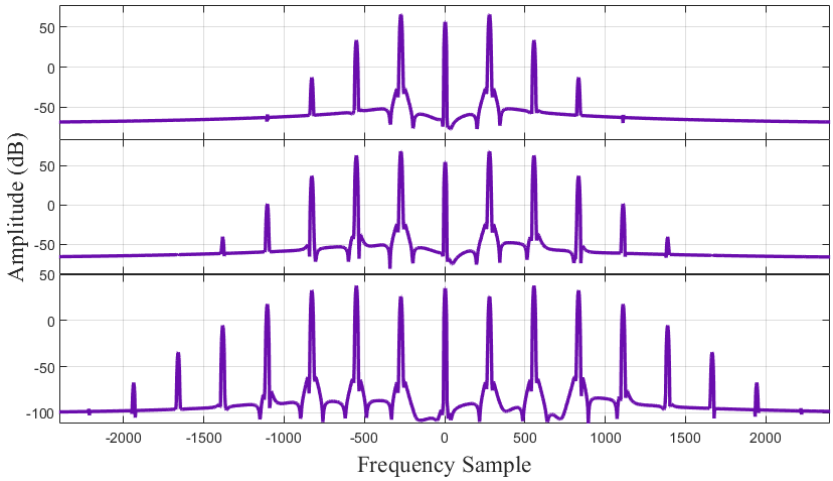


Figure 6.2: Doppler spectra acquired by, top: CMP ; middle: BTP; bottom: TWM, with a sampling rate of 7500 Hz for 2048 timesteps. Each model used a single rotor and $\Omega = 100\,Hz$.

These spectra yield evenly spaced HERM lines around the zero Doppler. No body returns were used in these simulations. The frequency modulation pattern of the three models is identical. The attenuation in Doppler is gradual for the point target models, but the second sideband of the TWM is higher than the first, which is qualitatively different to real expectations. The expectations upon any of these models' abilities to produce an accurate amplitude weighting of the sidebands are low, as there are many important radar principles that are not considered in the derivation. These include, the radar cross-section fluctuations of a rotating blade, the effect of radiation polarisation on returns, and the electromagnetic scattering regime in the case where $L \sim \lambda$ which is a more pertinent consideration in drones than helicopters; in our experiment in the next section, the operational wavelength was 24 cm and the length of the drone blade was ~19 cm. Sec. 6.2.2 describes an artificial sideband attenuation that will approximate the amplitude modulation observed in real data. We will only consider the TWM from now, and further focus of analysis will be to improve the likeness of the sideband's frequency pattern compared with real returns. The TWM was chosen over the other two models for the simple reason of it having the highest number of sidebands to be artificially modulated. A rigorous power spectrum simulation extension of this model may not use this choice, but as the frequency spectrum composition of the three models are the same, the choice to use TWM in this work was effectively arbitrary, and similar results could be achieved using any of the three models.

6.1.3 Multi-Rotor Drone Micro-Doppler Simulation

To augment the TWM to be useful for the purposes of modelling drones, it is important to reflect that there are, firstly, many blades on a motor, and many motors that in flight will each have a different instantaneous rotation speed. Denoting the number of motors on the drone as M and the number of blades per motor as B, summing the contributions from each individual scatter point in the integration gives the following equation,

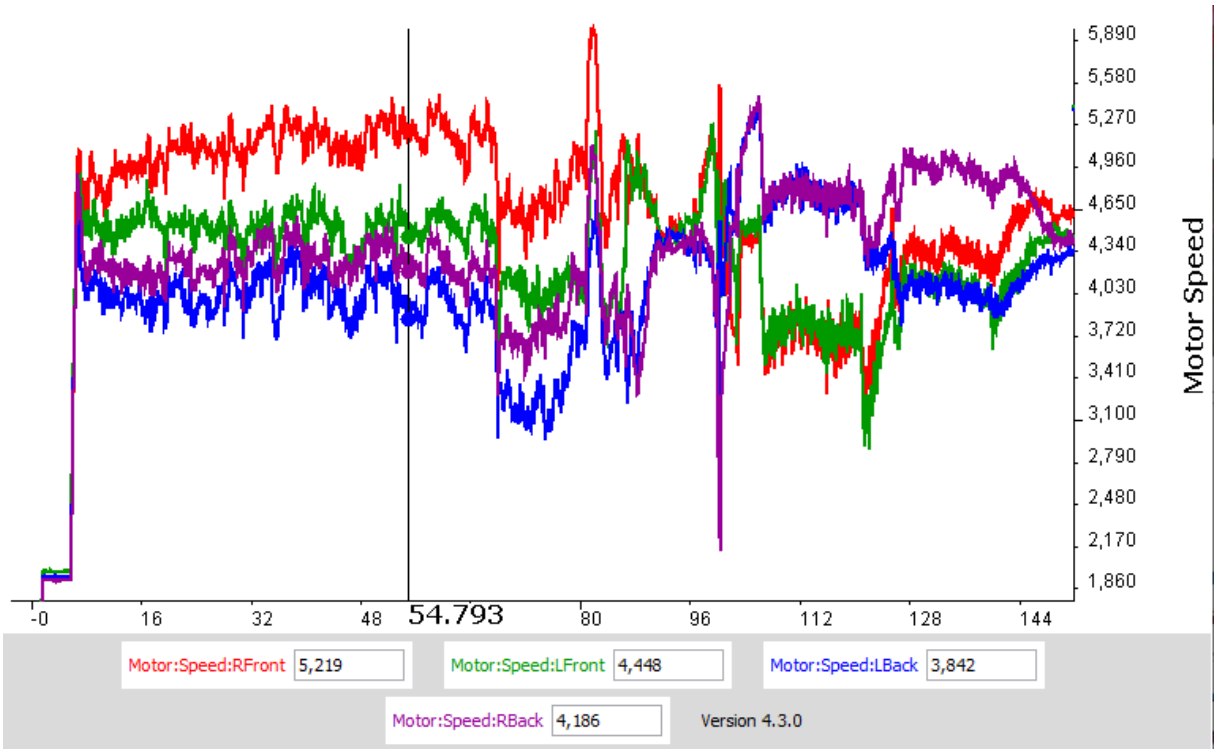
$$V_{I/Q}[t] \propto L \sum_{m=1}^{M} \sum_{b=1}^{B} \Big(\dots \\ e^{-j\frac{4\pi}{\lambda}(R_0+\frac{L}{2}cos(\varphi_t^{m,b}))} \; sinc\Big(\frac{L}{2}\frac{4\pi}{\lambda}cos(\varphi_t^{m,b})\Big)\Big) \tag{6.9}$$

where $\varphi_t^{m,b} = \varphi_0^m + \Omega^m t + \frac{2\pi b}{B}$, in which each motor can have a different initial blade angular offset, φ_0^m, and a unique rotation rate, Ω^m, and the blades are assumed to be evenly distributed around the motor hub, $\frac{2\pi b}{B}$.

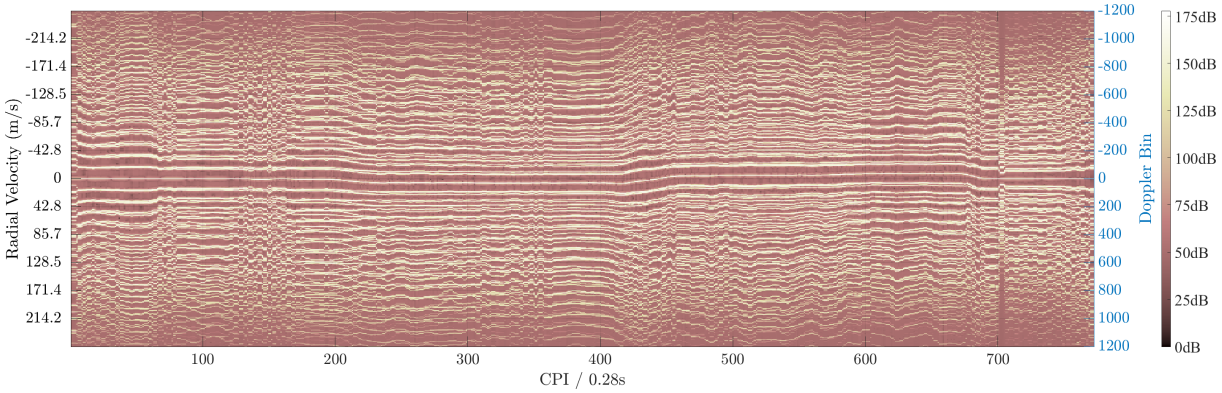
This equation as presented is a simplification of other works [267], [270], that include the height of the drone relative to the radar and blades with a root that is not at the centre of the motor hub, and [90] which also includes the tilt of the drone's motor plane and a divided and weighted blade model. The tilt and height of the drone was not included in this derivation, as in a long ranged application the angle correction will be very small assuming the drone is flying kilometres from the radar at a height below $\sim 100\,m$ - which was the case in our radar setup. That being said, this simplification could have a more significant impact when drones possess a large tilt from a high-acceleration manoeuvre.

When using this equation to create the spectrogram, M and B will define the hyperparameters of the drone. Further than these, $\Omega^m(t)$ and $R(t)$ are parameters related to the operation of a flight. There is total freedom to extract these parameters from a real flight via software, to synthesise them based off a real drone flight's trajectory [90] or motor voltages [271], or possibly to synthesise them in their entirety for a non-existent drone and arbitrary trajectory using a simulation platform that allows for the modelling of drone dynamics [272], [273]. In

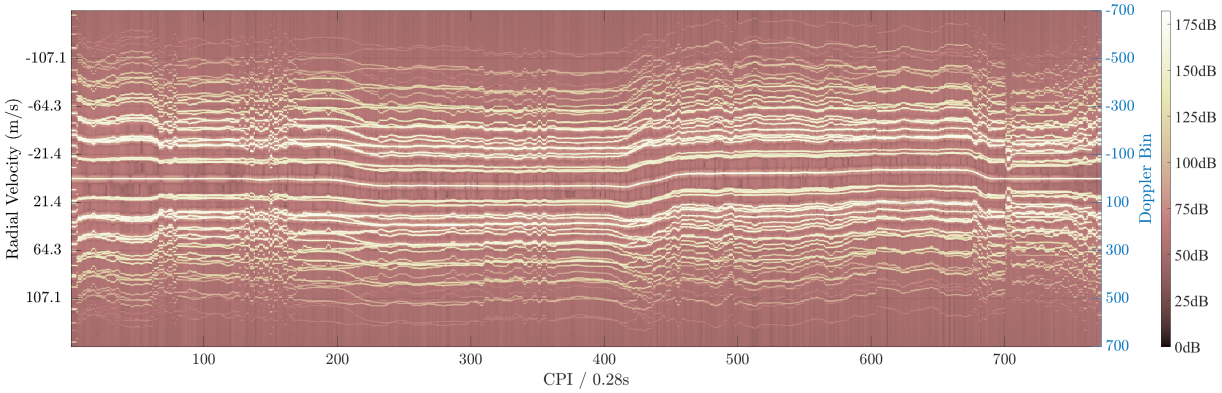
this work, the motor speeds used to create the spectrogram were taken from *CsvView* [274] as displayed in Fig. 6.3a, and were resampled to one value per CPI per motor, φ^m, using equation 6.9 to produce Fig. 6.4 for the three scattering regimes mentioned. This shows drone spectrograms using the measured motor speeds taken from a real drone flight. For comparison all three rotor models were used on different time windows to highlight the difference in simulation outputs. Again, in this image the top part is with CMP, the middle part is with BTM and bottom part is of the TWM. In each case the characteristic HERM lines are generated and they are all appearing at similar frequency offset - but it is the TWM model, that has the most number of sidebands and, as discussed, this is the model that will be used in the remainder of this work. There are further aspects to understand the relationship between motor speed sampling and simulation output which are expanded in Sec. 6.3.



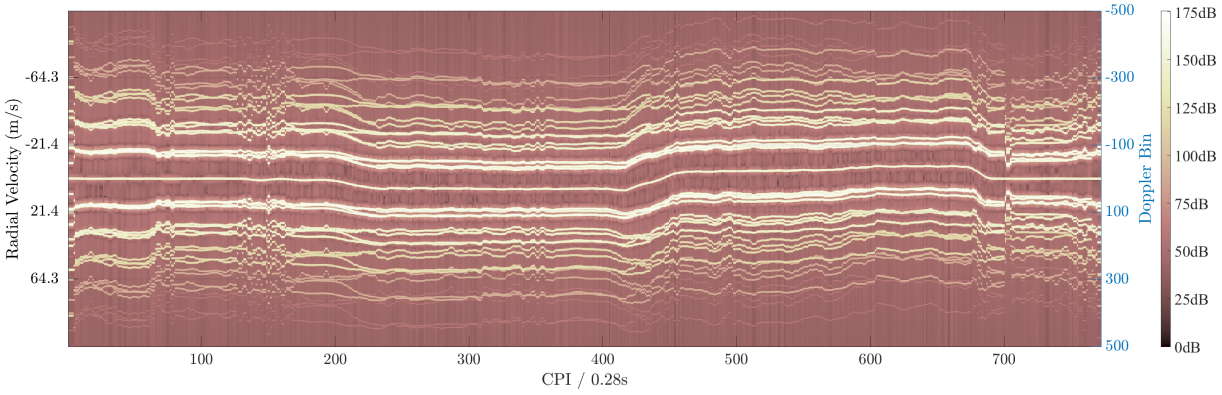
(a) Real motor speeds of a flight as viewed in *CsvView* [274].



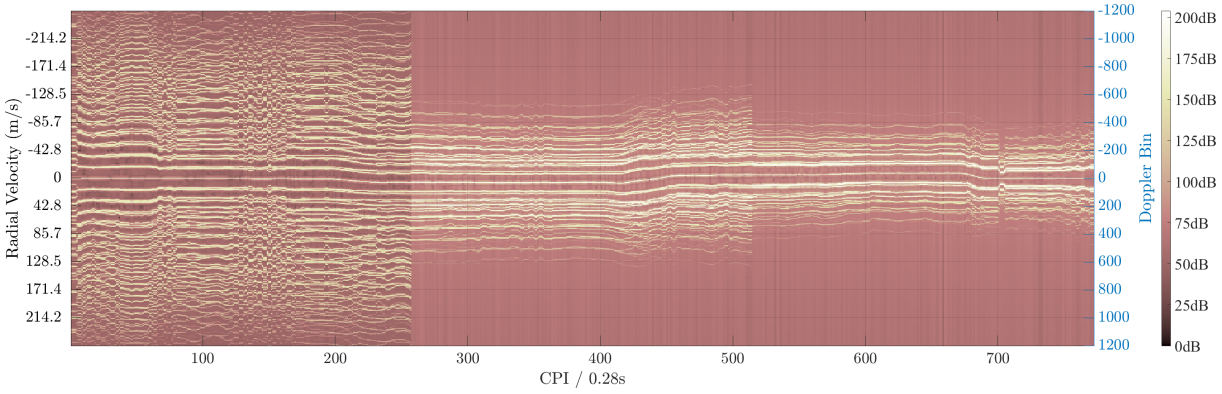
(a) Thin Wire Model



(b) Blade Tip Point Target



(c) Center of Mass Point Target



(d) Using TWM→BTP→CMP in a single spectogram, with shared axis and colorbar.

Figure 6.4: Spectrograms produced with the three scattering choices with real motor speeds.

6.2 From Simulation to Synthetic Data

In this section, two steps that add essential realism to the simulated returns used throughout this work are introduced. These steps were originally published in [1], and formed the method for converting simulated baseband timeseries samples into a realistic, high-fidelity drone μDoppler simulation. Sec. 6.2.1 describes injecting the simulated signal into a real recorded noise background, and Sec. 6.2.2 shows a method for modulating the amplitude response of the sidebands after FFT to make the appearance of the μDoppler more tunable and consistent with the observations of real data.

6.2.1 Injection into Realistic Background

Real radar data is used to provide the recorded background to which the simulation results data can be added. One option is to select the background from the same location as the real flight location, but shifting the time axis in order to use a time window when the drone wasn't in the scene. To ensure that active opportune targets are avoided, later work made use of night-time radar recordings to provide the basis of obtaining the sample background data. The timeseries data for the sample background data is extracted via the same method as described in Sec. 4.1.3 previously, but in this case there will not be the presence of a single target's returns for the whole duration of extracted data. The complex timeseries data from the simulated signal was summed elementwise with the extracted background data, which when Fourier transformed produces the target spectrum embedded in a real background. A spectrogram generated with this data will have the same approximate noise and clutter levels for each CPI as does the authentic flight. The simulated target was scaled to appear above the noise floor to have the same Signal-to-Noise Ratio (SNR) as the real target data, allowing a direct comparison of results. Using this method, some artefacts of the background appear in the final spectrogram results caused by opportune targets' presence in used resolution cells, such as stray birds or additionally the effect of weather on non-rigid clutter. It would be possible to gather backgrounds from everywhere within the radar area of regard to create a repository of real backgrounds that would allow the detection of

drones in unlikely, hard to fly places such as near the ground or in heavily built up areas.

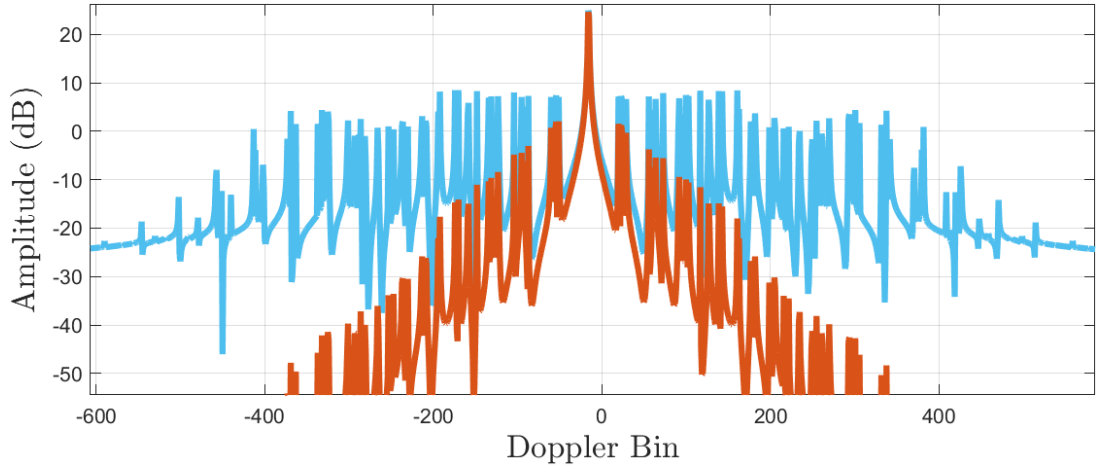
6.2.2 Phenomenological Sideband Attenuation

The amplitude modulation of sidebands post-FFT were observed for a single motor system in Figure 6.2 using equation 6.9, and for a multi-rotor system in Fig. 6.4. These simulations did not consider the main body. Typically, for real data, the strength of the first pair of HERM lines of a standard drone are around $15-20\,dB$ less than the main body, and then they decay further for the successive sidebands by $5-10\,dB$. For the two point target models the sidebands get successively smaller, but in the BTP the first sideband was weaker than the second. Drones at close range will produce many harmonics of the HERM sidebands due to the rotation period of a blade being much shorter than the CPI. The sidebands in the CMP model decay much more rapidly than would be anticipated with real data, with dB drops of 30 and 50 to the successive sidebands. The BTP model has smaller sideband dB drops of $4, 30$ and 35, which is closer to real expectations of the processing used. The TWM, after the second sideband, has still smaller drops of $4, 13$ and $22\,dB$. The expectations on any of the models' abilities to produce accurate amplitude modulation of the sidebands are low, as there any many important radar principles that are not considered in the derivation. These include, the reduction in RCS of blade when rotating through tangentially to radial axis, the effect of radiation polarisation on returns, and the electromagnetic scattering regime in the case where $L \gtrsim \lambda$ (which is a more pertinent consideration in drones than helicopters). To extend the model to better capture EM scattering phenomena to more accurately predict the power of returns would add considerable complexity to the model and may be less flexible for a wider variety of real drones. Instead, empirically adjusting the sideband strength by attenuating returns as a function of Doppler frequency offsets from the main body was done to make the simulation tunable and to better match real expectations. Due to the simplicity of the model used in this simulation, the TWM was chosen due to it having the largest number of sidebands present to be artificially modulated, with the goal of manually maximising the visual appearance likeness to real data returns. The solution implemented was to apply an inverse Fourier transform to a frequency weighting function, and multiplying this elementwise with the

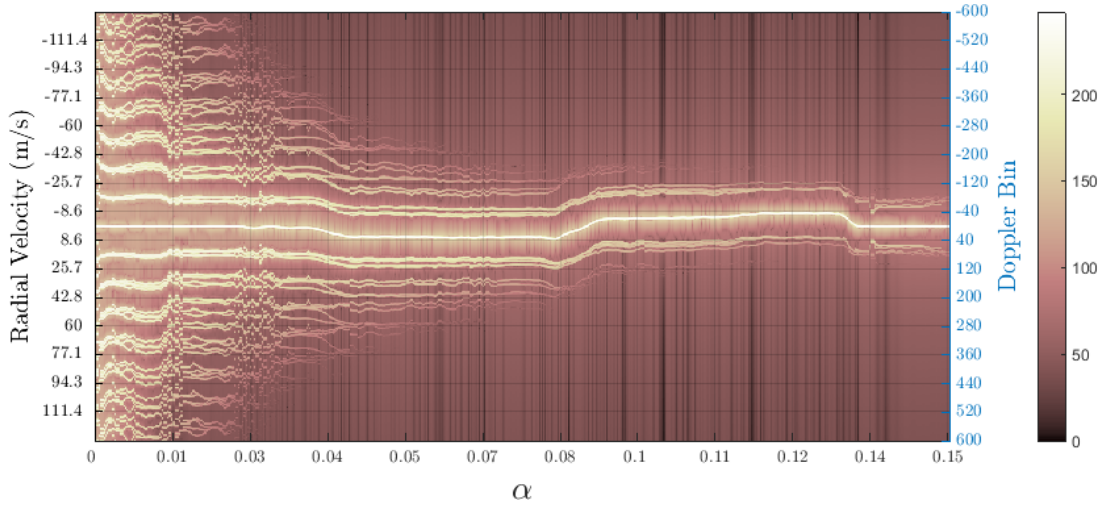
simulated timeseries data. The amplitude modulation envelope was a back-to-back exponential decay of the form,

$$e^{-\alpha|\nu|}\,, \tag{6.10}$$

where ν is frequency offset, and α is a user defined hyperparameter that when $\alpha > 0.1$ tightly constricted the sidebands to the main body. The amplitude envelope was centred around the Doppler bin of the body return to ensure the sidebands decayed symmetrically. See Figure 6.5 for the comparison of a raw spectra with and without the artificial amplitude modulation, without any background injection. This method ensures the sideband power decreases logarithmicly as the frequency offset from the main body increases, which was the more common observation of real data and not present consistently using directly the simulation output. A downside of this approach is that the attenuation is a function of Doppler shift, and not sideband harmonic which is the case in real data, but future work can improve on this by attenuating the returns of each motor individually to better reflect real observations of sideband power. Fig. 6.5a shows the effect of the filter as used on the simulated signal from equation 6.9, and Fig. 6.5b shows a simulated spectrogram using recorded motor speeds with α varied linearly in time from 1 to 1000, which as it is increased, so the power and number of sidebands are reduced.



(a) Simulated spectrum as equation 6.9 before (blue) and after (orange) the application of phenomenological sideband attenuation using $\alpha = 0.017$



(b) Simulated spectrogram varying sideband attenuation factor α from $0 \rightarrow 0.15$

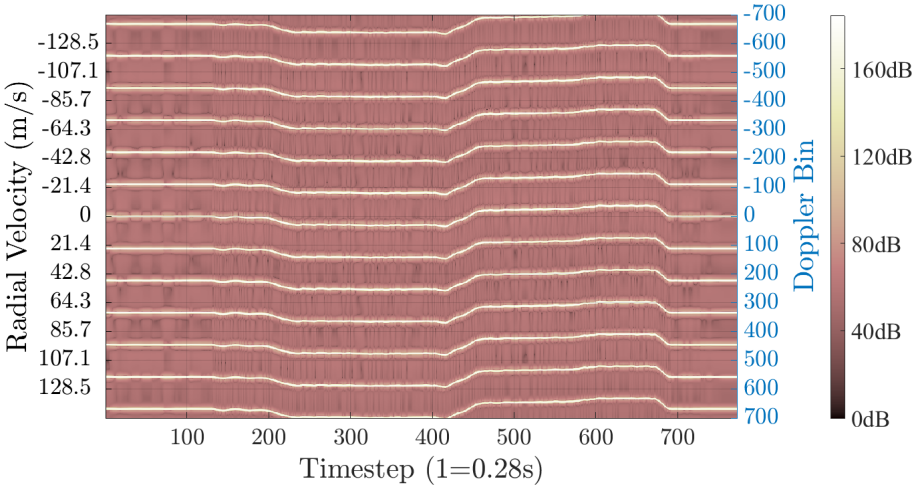
Figure 6.5: Effect of phenomenological sideband attenuation on a simulated spectra and spectrogram.

6.3 Motor Speed Sampling for Micro-Doppler Simulation

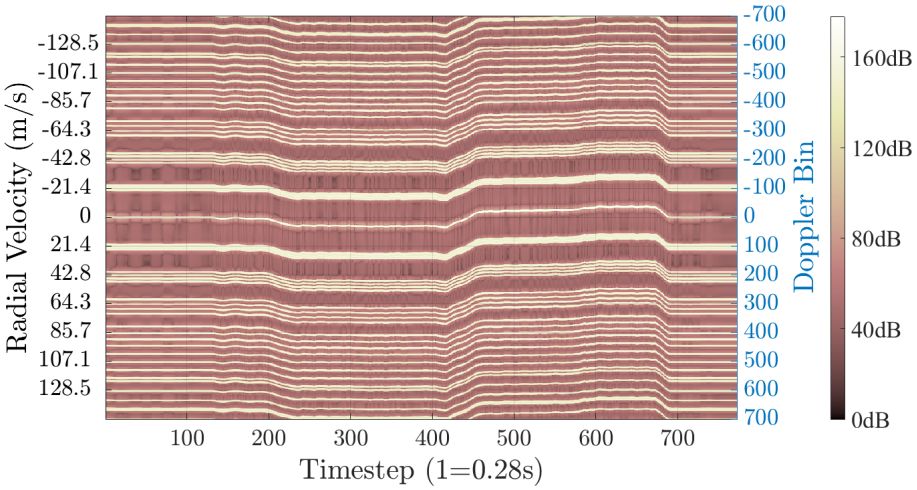
A major technical contribution of this project was the exploitation of real motor speed recordings in the production of synthetic data, originally reported in [1], [3]. At first, Sec. 6.3.1 provides an introduction of the effect of motor speeds on spectrograms with basic simulation implementations and discussion with existing literature on the subject. Then in Sec. 6.3.2, an investigation of the resampling of drone log files for the implementation in the simulation is explored. An analysis of the choice of motor speed sampling for drone sideband modelling had not been published prior to this work. The approaches introduced in this work are but one option to solve the proposed challenge, and future research may improve on the specific implementations shown. With that said, this work concretely shows in this and the following chapter that a drone simulation is only as good as the motor speed resampling applied to either real truth motor speeds or generated motor speeds.

6.3.1 Introduction to the Effect of Motor Speeds

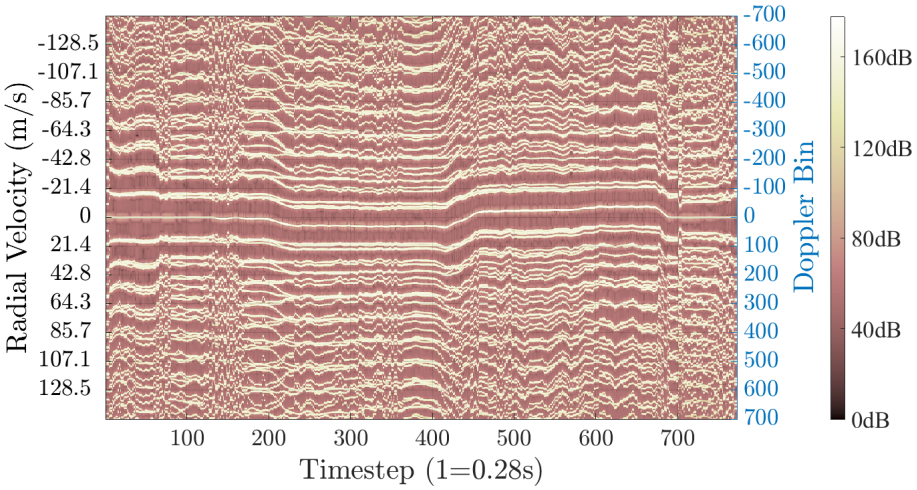
Motor speeds were gathered from real flights using [274] as discussed in Sec. 4.1.3. Drone motor speeds, Ω_m^t, are expected to be around 40 Hz for larger, ~20 kg drones, while nano- or micro- racing drones could be up to 250 Hz. This depends for any given drone primarily on the propeller length according to the propeller trust equation, and secondarily on the pitch or angle of attack of the blade that is fixed. Figure 6.6 shows simulated spectrograms generated using the TWM (equation 6.9) with different motor speed choices. Fig. 6.6a uses constant motor speeds across all motors and time, and uniform single sidebands are observed in this instance. Fig. 6.6b shows the result obtained when each of the four motors are set to a different fixed speeds of 85, 90, 95 and 100 Hz respectively, and now there are four distinct sidebands within each harmonic grouping that result from the differing speeds of each motor. Fig. 6.6b uses real motor speed values that were extracted from a drone flight and resampled to be constant over each CPI. Further detail of resampling recorded motor speeds is explained following in Sec. 6.3.2.



(a) Four Fixed Motor Speeds each at 100 Hz



(b) Four Fixed Motor Speeds at 85, 90, 95 and 100 Hz



(c) Real Recorded Motor Speeds Resampled to 1-per-CPI as [1]

Figure 6.6: HERM Line spectrogram from equation 6.9 simulation model applied to a quadcopter of blade length, $L = 0.19$ m.

The comparison of these examples makes clear the importance of motor speed selection to how representative a simulation is of real world measurements. In literature regarding Multi-Rotor Micro-Doppler (MRμD), the author is only aware of one other work that attempted to use realistic motor speeds in a multi-rotor simulation. This was

- [90] - Bennett, Harman, and Petrunin *"Realistic Simulation of Drone Micro-Doppler Signatures"* (2021)

that reverse engineered the trajectory of a real drone flight via a physics simulation to estimate the motor speeds that were used to create the trajectory. This work successfully captured the multiplicity of rotors that matched the Doppler distribution of sidebands as found in real data. Otherwise, the Martin-Mulgrew equation [267] has been used in the creation of drone model specific micro-Doppler spectrograms using simulated parameters chosen to reflect the physical nature of the targets. Many of these simulations assumed constant motor speeds for the entire flight without consideration to their fluctuation over time [122], [126], [129], [247], [250], [255]. As shown in Figure 6.6, the inclusion of motor speeds would positively increase the realism of synthetic data and could have an effect on its ability to support classification, as will be explored afterwards in Chapter 7.

6.3.2 Re-sampling Raw Recorded Motor Speeds

The task of resampling the raw recorded motor speeds, Ω^m, to be used for each pulse of a CPI in equation 6.9 is addressed in this subsection. This transformation could be done in a number of ways, which would lead to different levels of simulation fidelity. The recording of four real motor speeds necessitated the consideration of all four motors possessing different speeds, which was also done in [90]. Beyond this, the level of motor speed variation between and within consecutive CPIs will have considerable implications on how the motor blade returns are integrated in a FFT, and has not been explored in literature prior to this study. Two regimes of motor speed resampling are defined within the described simulation framework.

In flight, the motor speeds are recorded at an approximate constant rate, of 30 Hz for the *Inspire* and 20 Hz for the *Matrice*. A host of resampling techniques could be applied to convert the original recorded signal. The simplest implementation is to resample the recordings to yield a single motor speed per CPI, and use this constant value of Ω_m^t for the consecutive pulses in a CPI, while using different motor speeds for each successive CPI. The approach in this work is referred to as per-Coherent Processing Interval (Per-CPI) sampling. The resultant spectrogram and sampling approach are shown in Figure 6.7. The original motor speed signal that was shown in Fig. 6.3a is shown again in Fig. 6.7b plotted alongside the resampled signal and the original captured signal, with the real and synthetic spectrograms in Fig. 6.7a and 6.7c respectively.

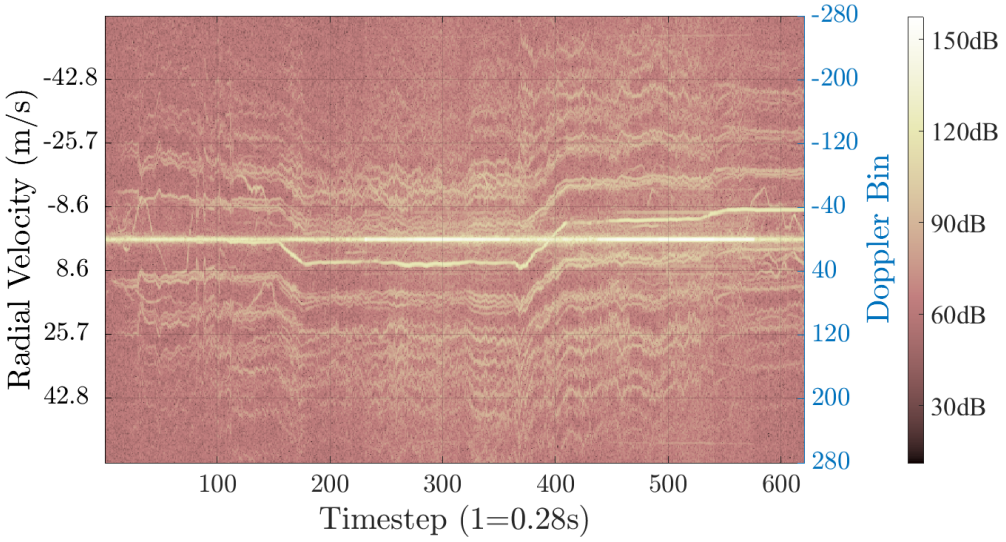
The simulation here is injected into a real background and has had the sidebands attenuated to make for better comparisons with the real data. A piece-wise cubic spline [275]with degree, 3, was used to uniformly under-sample and smooth the motor speed signal at each CPI. This approach sensibly recreated the global frequency modulation structure of the real drone signature. Smoothing of the resampled motor speed signal, as observed in Fig. 6.7b, was necessary to avoid constant discontinuities within the sidebands of the spectrogram. The most noticeable differences between the real and resultant simulated data are both artefacts of using Per-CPI sampling:

- Motor speeds are not restricted to be similar across overlapping CPIs, resulting in motor returns jumping discontinuously across Doppler bins in successive timesteps. The spline approach that resampled the motor speeds was not reactive to large fluctuations in motor speeds, as can be seen for numerous timesteps in the first 150 timesteps in Fig 6.7c.

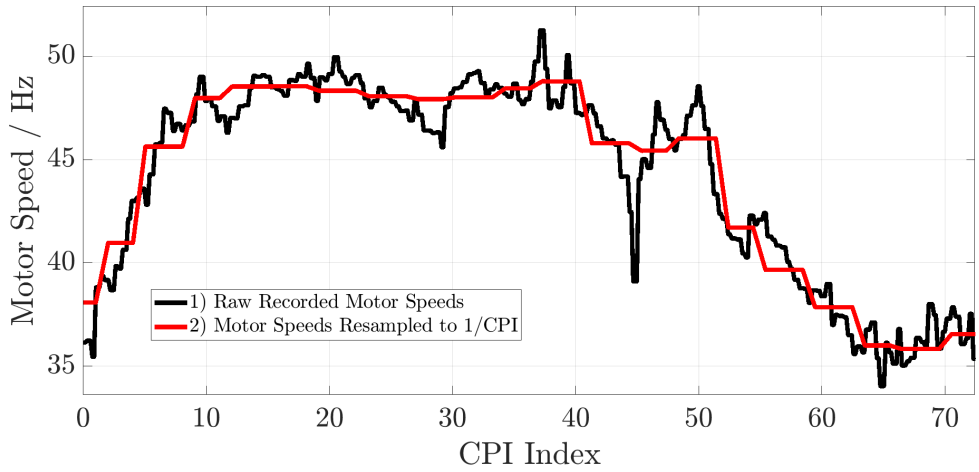
- Instances where the drones motor speeds are fluctuating rapidly over the duration of the CPI, will only have one sample for its representation in the simulation. Long window CPIs that incorporate many pulses will exist over acceleration of the blade rotation, which will lead to incoherent integration and signal spread over multiple Doppler bins.

In order to improve on these observations, the motor speeds were sampled to less than the duration of the CPI, which will be referred to as sub-Coherent Processing Interval (Sub-CPI) sampling. The following steps make reference to Fig. 6.8b that show the motor speed signal over the course of a CPI comprising 4096 pulses at 7.3 kHz, for the originally 20 Hz *Matrice* motor speed signal. At first, the motor speeds were down-sampled (from 30 Hz) via decimation to 8/CPI (~14.3 Hz), and then up-sampled by an integer factor (of 4096/8 = 128) to PRF (7.3 kHz) via time domain 0-padding to the required signal length and application of a low-pass interpolating filter. Applying the present motor speed signal to equation 6.9 followed by FFT produced HERM lines that were spread widely over the spectrum with insufficient integration gain. Smoothing these values significantly using a rolling-average with a long window length (between approximately 1,000 to 3,500 compared to the total CPI length of 4096) produced highly convincing sidebands with more realistic bandwidth and better continuity across CPIs.

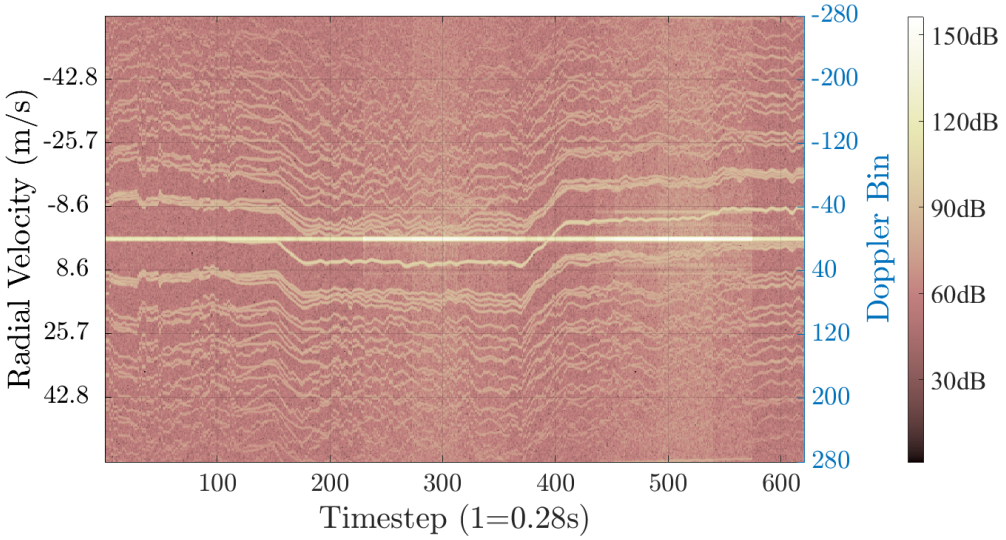
The effect of the rolling average is demonstrated in Figure 6.9 where the length of window ranges linearly from 1 to 4095, without any sideband amplitude modulation or background injection. This is presented using the TWM without any sideband modulation and without a real injected background. At the higher end of this scale, the sidebands are continuous and have a non-constant spectral width that is not true of real returns. It can be seen that with low rolling average window lengths, the sidebands are very widely spread to the extent that individual motor's contributions cannot be resolved until the window length is sufficiently long. In other radar systems with different sampling rates and integration durations, returns may be much more heavily modulated by the effect of this window length.



(a) Original real spectrogram of *Inspire* and Radar# 1

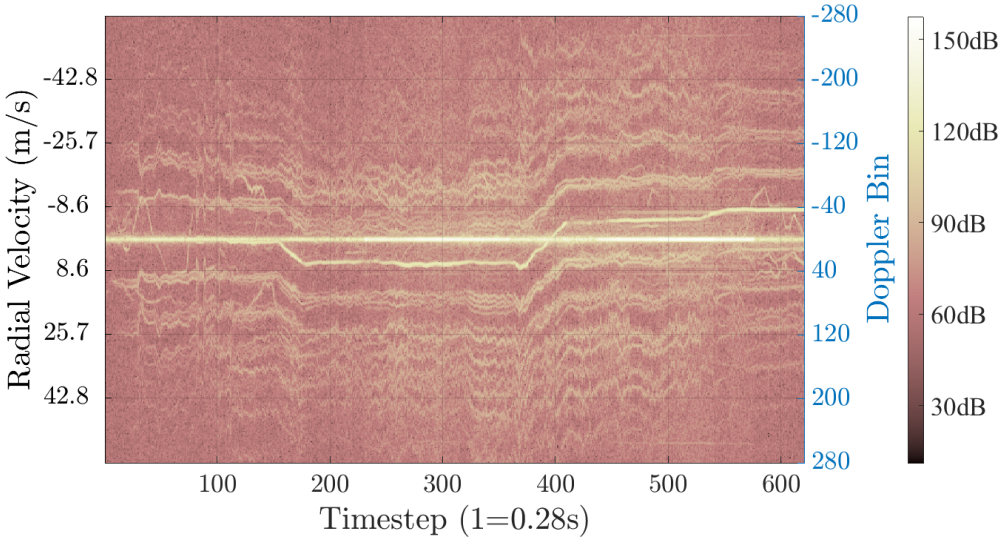


(b) Per-CPI Motor Speed Signal Resampling

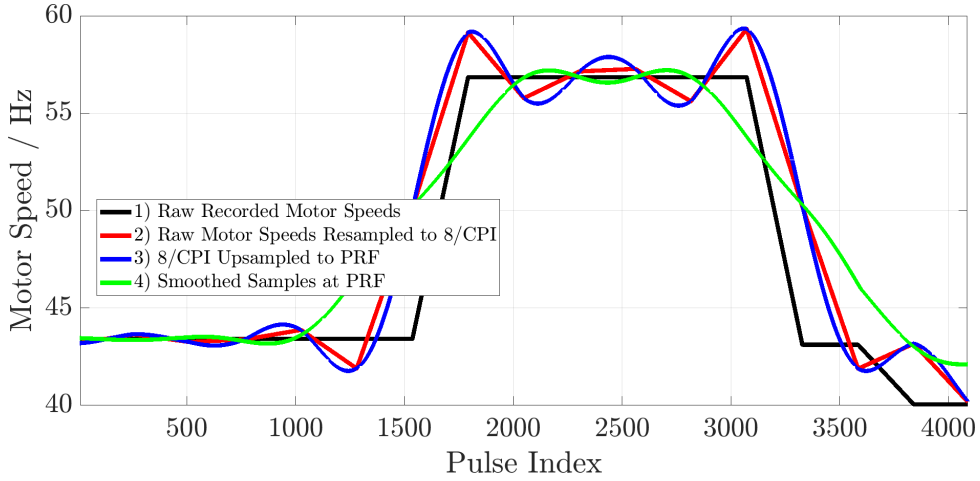


(c) Resultant Synthetic Per-CPI Spectrogram

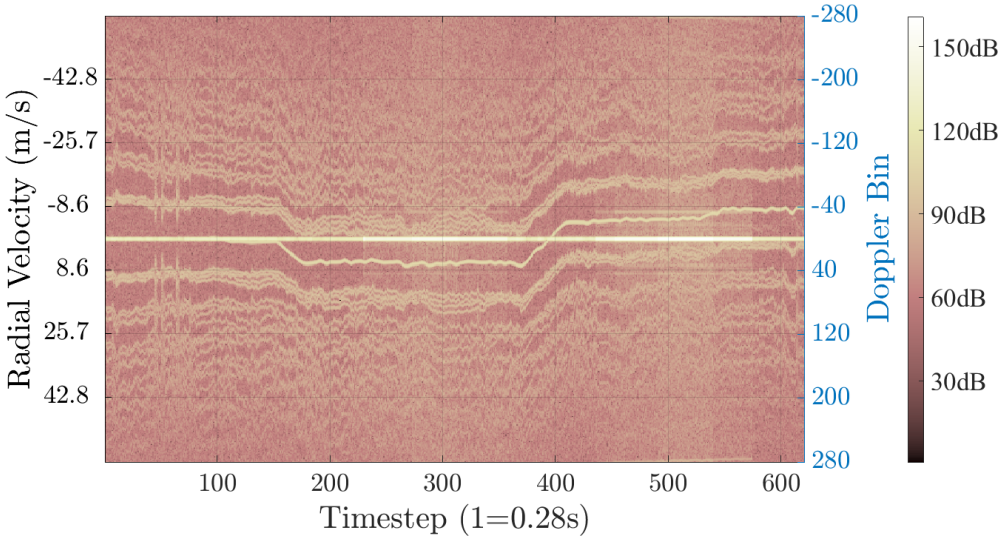
Figure 6.7: Per-CPI Sampling Approach and Result.



(a) Original Real spectrogram



(b) Sub-CPI Motor Speed Signal Resampling Approach



(c) Resultant Synthetic Sub-CPI Spectrogram

Figure 6.8: Sub-CPI Sampling Approach and Result.

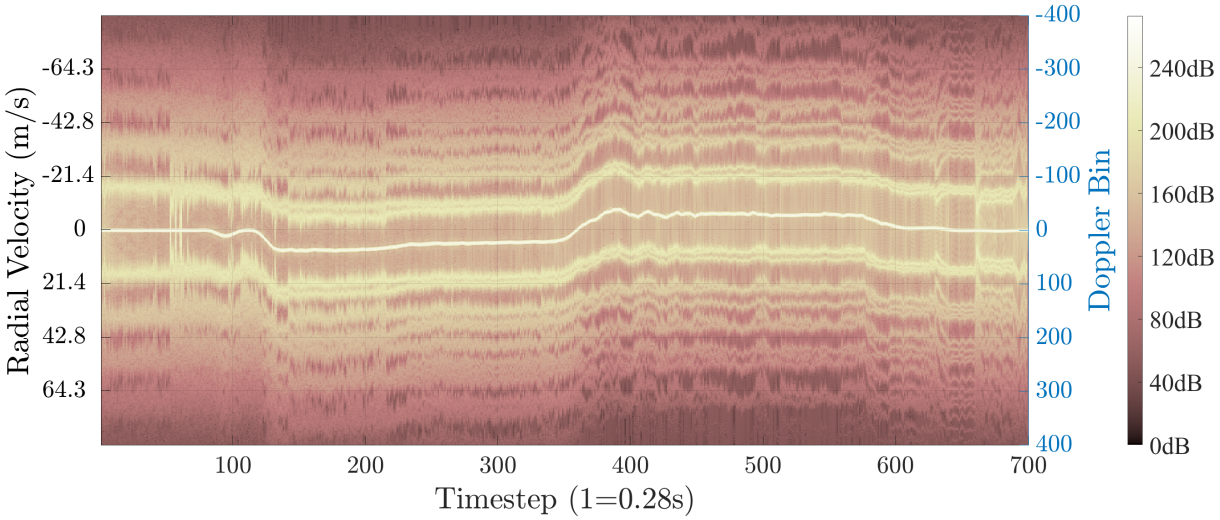


Figure 6.9: The effect of rolling average window on Sub-CPI sideband spectral width varying the window from 1 to (1 - K), where K = 4096, the length of the FFT.

6.4 Chapter Summary

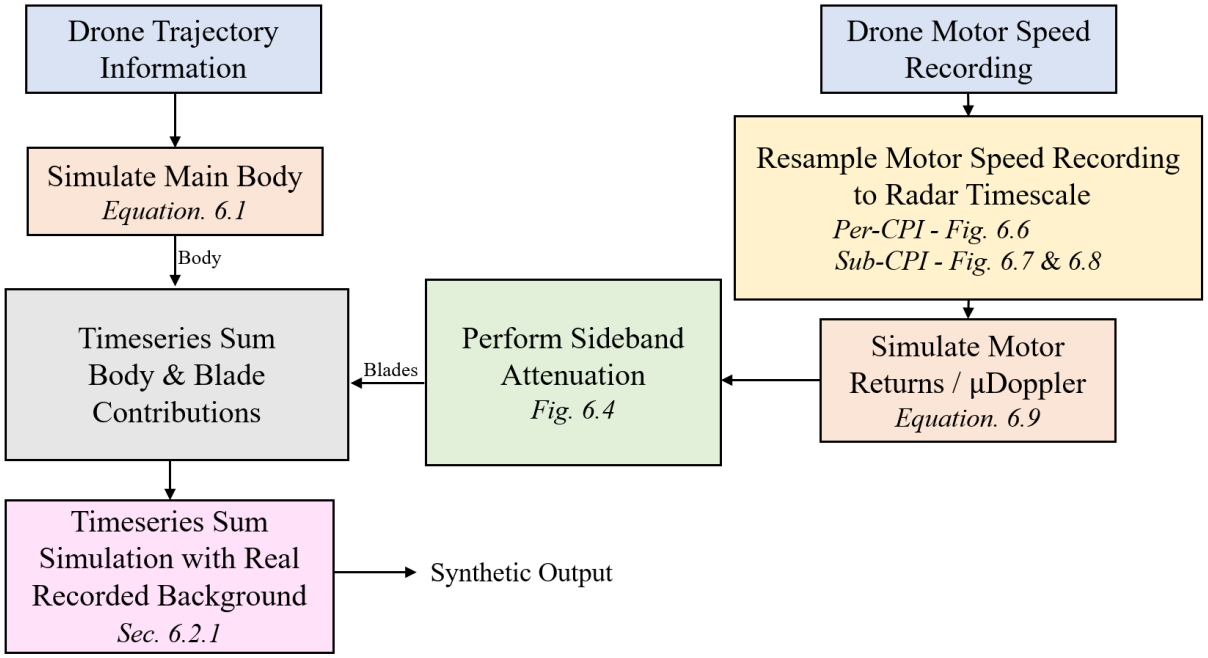


Figure 6.10: Flow diagram for the creation of a synthetic spectrogram as performed in this chapter.

This chapter presented a method for creating a realistic simulation of drone radar returns using flight log files. Presented was:

- a derivation of a rotating blade and motor structure (Sec. 6.1.2) starting from the baseband return of a point target (Sec. 6.1.1),
- the extension from single to multiple rotors (Sec 6.1.3) for drone modelling,
- the treatment of the amplitude response of sidebands as a function of Doppler frequency offset to make them more consistent with real observations (Sec 6.2.2),
- the summation of a simulated signal with a real radar recording to allow the injection of the synthetic target into a real background (Sec 6.2.1),
- a further analysis of the resultant simulated μDoppler produced by more granularly re-sampling the motor speed data above the rate of CPI (Section 6.3).

These steps, illustrated in Figure 6.10 to create high-fidelity, finely-controlled synthetic data may be used for a host of applications, with the most urgent one being to overcome the training data limit for deep learning applications. Previous simulations without consideration of multiple-rotors and/or differing speeds did not possess the required fidelity to seriously entertain synthetic training for application to real data. This will be the main subject of the following Chapter 7, where the synthetic data defined in this chapter is tested against real data in a series of experiments aimed to understand how well a CNN utilises it for classification. The following Chapter 7 will share the approach taken to create a large dataset of synthetic data; while in this chapter only the creation of a single flight was considered. Sec. 8.2.1 in Chapter 8 will also use this simulation approach and will demonstrate its capability to match and exceed benchmarks with real data. To ensure that proclaimed findings of synthetic training results are robust and to fully assess their effectiveness as compared to real data, the experiments will be conducted with an equivalent amount of synthetic data, purposefully withholding the ultimate advantage of increased data quantities and variation of target parameters. This allows results to be presented as a function of simulation fidelity, enabling others and future work to benefit from the understanding gained as well as the primary advantages of using synthetic data for deep learning and other applications.

Chapter 7

Classification with Synthetic Data

The μDoppler simulation approach from Chapter 6 showed several advances to the creation of realistic synthetic drone data. The present chapter uses this data to train a deep learning classifier for classifying real data in an attempt to validate the approach and thus help overcome the limit of availability of radar data for multi-rotor drones. The methods presented are universally applicable to all radar systems, and indirectly to all drone targets. Section 7.1 defines a benchmarking synthetic dataset. With this, the supervised Convolutional Neural Network (CNN) classifiers that were trained with real drone and bird L-band radar data in Chapter 5 will be revisited with synthetic data. A series of experiments using different versions of motor speed sampling in Section 7.2 are aimed to show what is required for reliable performance, and how future work may proceed to better utilise the proposed approaches to build a trustworthy classifier. Section 7.3 summaries the results and merits of the approach beyond the experiments performed in this chapter. Synthetic data can enable a much more extensive training that will allow deployed radar systems to have awareness of known threats within the scope of the domain.

7.1 Creating a Synthetic Micro-Doppler Dataset

This section demonstrates the creation of a synthetic drone dataset for classification of real drone targets. The mathematical creation of highly-realistic drone data from the previous chapter will be used in this section to create synthetic data to replace the real data of drones possessing Micro-Doppler (μDoppler). The effectiveness of the classifier trained on synthetic data will be analysed, and the results are shown to justify its use and motivate future work to extend upon the findings shared. Sec. 7.1.1 will briefly summarise the motivation for the chapter and show where previously the benefits of synthetic data have been demonstrated for deep CNN classification of real radar drone data. Sec. 7.1.2 following will formalise the steps taken in Chapter 6 for the construction of a large dataset of synthetic data that is equal in quantity and with similar return variation as used in the real data benchmark from Chapter 5. Section 7.2 afterwards will contain the results for a variety of simulation approaches.

7.1.1 Motivation

To both investigate new classification models, and to also improve the performance by using larger NN models, the creation of synthetic training data via simulation would make for an extremely beneficial addition to the state of the art of automatic target classification using radar. [250] uses the Martin-Mulgrew simulation model injected into a real training dataset to allow the classification of unseen models in the test set, improving accuracy from 82% to 91% thanks to the synthetic data. [119] also uses synthetic data to perform classification of birds, cars and drones replacing the real training data with simulated. Both of these works did not consider Multi-Rotor Micro-Doppler (MRμD) for their simulations. Simulating the raw and processed returns from a target that resembles the real data accurately enables many useful applications, such as validation of handcrafted feature extraction methods of spectral data, the variation of noise backgrounds and target parameters, and their use in synthetic training that is the subject of this chapter. This work will use the classification results with the optically pre-trained, AlexNet-based CNN ($\mathcal{A}$CNN) presented in Chapter 5 as the real benchmark performance against which

Table 7.1: Composition of All Drones and Birds (ADB) data subsets. *Simulated Data

Target	Train Samples	Test Samples	Validation Samples
Bird	3625	1818	371
Inspire	-	612	-
*Inspire**	1444	-	127
Matrice	-	671	-
*Matrice**	1058	-	170
Mini	1129	535	45
Total	7256 (1129 + 2502* + 3625)	3636 (1818 + 1818)	713 (45 + 297* + 371)

to assess the new, synthetically trained models.

7.1.2 Creation of Benchmark Synthetic Data

To concretely explore the value of the synthetic data created in this process for training of safety-critical classifiers, we experiment with using a training and validation set that is equivalent to the real benchmark in the number of samples contained, and uses synthetic spectrograms that are parameterised to be as identical in spectra to the real set as possible. The real All Drones and Birds (ADB) dataset from Sec. 5.2.5 was replicated with the μDoppler producing drone targets (*Inspire* and *Matrice*) replaced with equivalent synthetic data.

Table 5.8 is reproduced in Table 7.1 that shows the quantities of synthetic data and real data used. As detailed in Chapter 6, the construction of realistic synthetic data from a simulation can be performed by

1. simulating the drone body (eq. 6.5) using GPS truth,
2. simulating the drone rotor μDoppler (eq. 6.9) using sampled motor speed,
3. applying amplitude scaling to adjust sideband SNR (Sec. 6.2.2), and
4. extracting real radar background (Sec. 6.2.1) and adding it to the drone simulated body and rotor spectra.

This approach can be summarised in the following summation for each timestep, t, in a CPI,

$$\begin{aligned} V_{CPI}^{synthetic}[t] = & S_{Body} * e^{-j\frac{4\pi R(t)}{\lambda}} \\ & + S_{Blade} * V_{IQ}[t] \cdot \text{IFT}(e^{-\alpha(|\nu|-\nu_{body}[t])}) \\ & + V_{BG}[t] \end{aligned} \tag{7.1}$$

where V_{IQ} from equation 6.9 is summed with an amplitude attenuation envelope using IFT, the inverse Fourier transform operation, $X_{BG}[t]$ are the timeseries samples from the recorded background data, S_{Body} and S_{Blade} are scaling factors for the body and blade reflectors respectively, α is a hyperparameter that controls the number of HERM lines present, ν is Doppler frequency and ν_{Body} is the central Doppler frequency of body returns. This equation defines an approach to creating a wide variation of realistic, simulation spectra that has total control of many parameters. The list of steps enumerated previously are reiterated to illustrate the approaches flexibility.

- A choice of drone physical (number of blades, length of blades, blade and body size) and operational parameters (trajectory flown, and motor speeds and associated sampling choices; see Sec. 6.3),
- An arbitrary choice of background to inject the data, that could include challenging close to the ground positioning or amongst a bird flock.
- Tuning the SNR, spectral width and amplitude of each of the reflecting components simulated,

From this outline for a single flight, the choice of parameters used to assess synthetic training with a real dataset comparison was to recreate the original real dataset. S_{Body}, S_{Blade} and α needed to be separately set for both the *Inspire* and *Matrice* targets, repeated for each radar and also for both the Per-CPI and Sub-CPI versions. The parameters of simulation were set

to match the body SNR, and sideband count and SNR, at the nearest and furthest ranges used in the real set. Then, S_{Body} was determined for an arbitrary flight by observing its range and interpolating between the nearest and further values obeying the inverse 4^{th} power scaling as per the RRE. The sideband factors (α and S_{Blade}) were similarly set at the furthest and nearest ranges used, and then both were interpolated linearly between the smallest and greatest values for the arbitrary drone flight that together provided a suitable non-linear scaling that produced a realistic sideband appearance and power.

Figure 7.1 shows the real and synthetic Per-CPI and Sub-CPI comparison of the *Matrice* drone, at long and short range for Radar#1, and at short range for Radar#2. Any imperfections in SNR across the real and synthetic datasets will be absorbed from using the RGB colormap and would not be expected to have a large impact on results. Table 7.2 lists the parameters and motor sampling used in their creation. At the longest ranges shown in Figs. 7.1d - 7.1f, the real sidebands fluctuated in their appearance and power. This simulation as model presented does not attempt to model such fluctuations and synthetic parameter values were set so that sidebands were always visible in the synthetic data. This could be easily introduced into this dataset by randomly increasing the value of α, or with an analysis of what possible drone manoeuvres and conditions cause scattering to reduce, such as blade flex or observation from the body.

Figure 7.1: Real and Synthetic *Matrice* spectrograms as used for Classifier Training, with (a-c) Radar#1 at 1.4 km, (d-f) Radar#1 at 1.4 km), (g-i) Radar#1 at 1.4 km; (a,d,g) Sub-CPI, (b,e,h) Per-CPI, (c,f,i) Real.

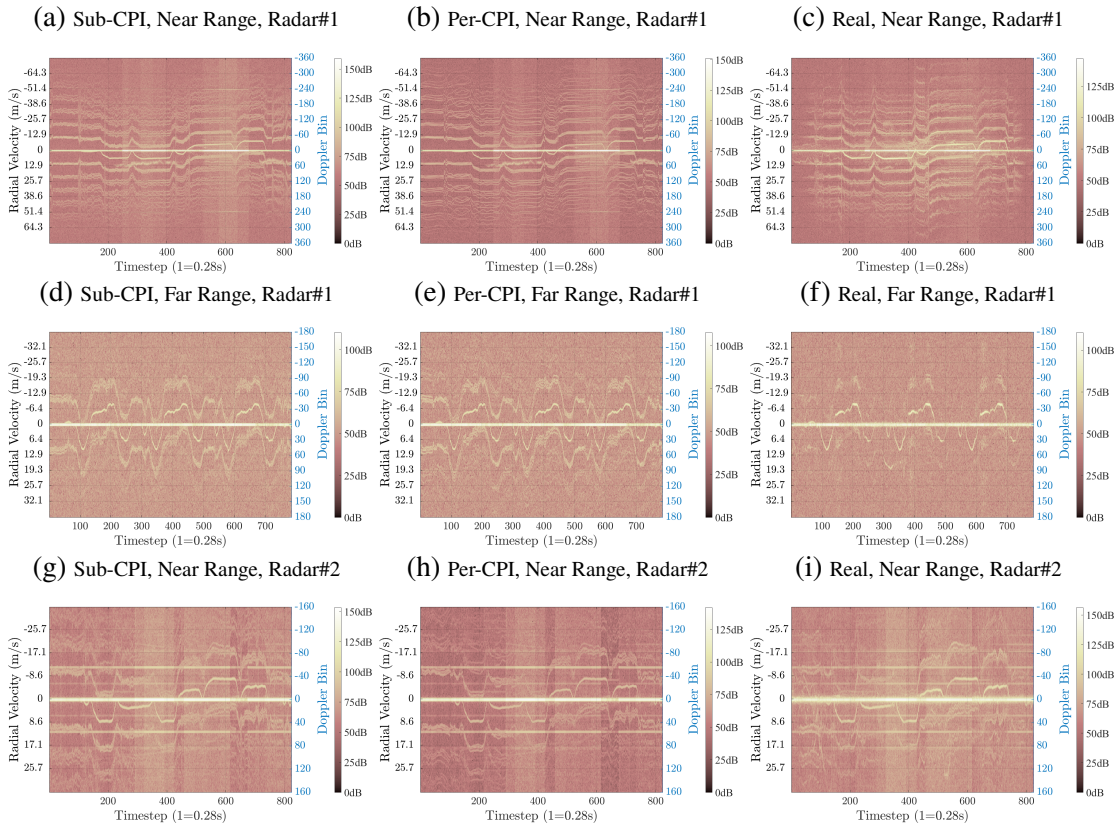


Table 7.2: Simulation Parameters Used in Benchmark Synthetic Dataset Creation ($2E^{-3} \equiv 2 \times 10^{-3}$)

		Matrice	*Inspire*	*Matrice*	*Inspire*	*Matrice*	*Inspire*
Synthetic Version	Motor Sampling Function, Ω_t^m	$\boldsymbol{S_{body}}$ Radar#1 Min/Max, Radar#2 Min/Max		$\boldsymbol{S_{blade}}$ Radar#1 Min/Max, Radar#2 Min/Max		$\boldsymbol{\alpha}$ Radar#1 Min/Max, Radar#2 Min/Max	
Sub-CPI	Real Speeds sampled to PRF/8 & Long Window Rolling Average	$E^{-5}/2.5E^{-1}$ $2E^{-2}/.25$	$2E^{-2}/1$ $2E^{-2}/2E^{-1}$	5/40 5/35	7/32, 5/32	$5.5E^{-3}/1.1E^{-3}$, $7E^{-3}/1.7E^{-2}$	$5.5E^{-3}/1.1E^{-2}$, $5.5E^{-3}/8E^{-2}$
Per-CPI	Real Speeds sampled to CPI	$E^{-5}/1.4E^{-3}$, E^{-10}/E^{-7}	$E^{-13}/5E^{-2}$, $E^{-12}/2E^{-7}$	5/32 7/30	7/30, 10/30	$5.5E^{-3}/$ $1.1E^{-2}$, $2.7E^{-3}/E^{-2}$	$E^{-2}/6E^{-3}$, $E^{-2}/7E^{-3}$

The background for the synthetic target was extracted by time-shifting the real flight to a time when there was no control targets present. Extracting the raw data in the same way produces a spectrogram with the same local (but not time-dependent) clutter properties as the real flight for each timestep of the flight. The simulated target's 1-D timeseries returns are summed element-wise with the background signal extracted, injecting the simulated target into a real background. This approach was taken to maximise the similarity of the real and synthetic data background properties, but this is limited as the real data came from regular intervals over a two-year period and the background data came from a single morning on the 3^{textrd} of March 2022 between 1 and 3AM local time. The variation of weather and wind on clutter is present in the real data set but not as much for the synthetic cases. The weather conditions and locations could be altered to yield varied backgrounds for training data, in order to prepare the classifier for returns expected in the field of operation, but this was not attempted in this study. A discussion of further options of backgrounds to use is presented in Sec. 9.2.

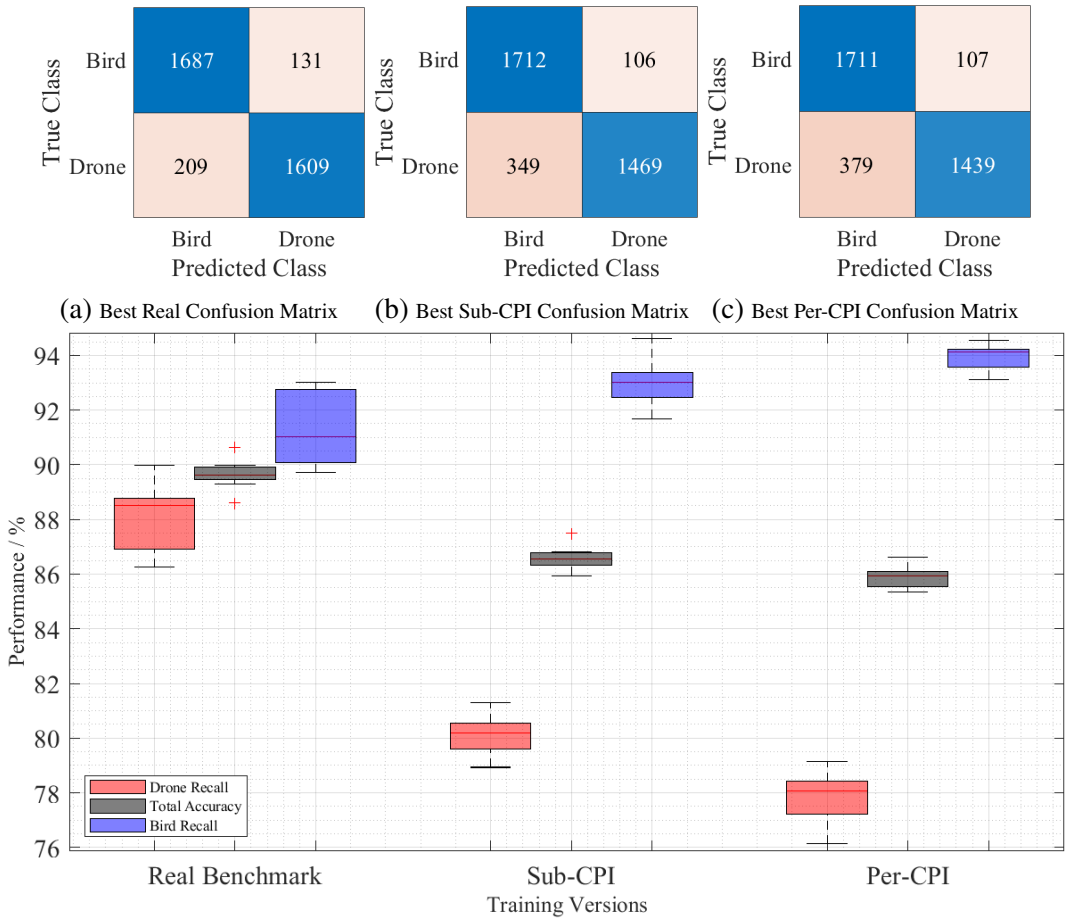
7.2 Synthetic Training for Real Classification

Using the simulated datasets described in the previous section, the results of a variety of synthetic training approaches are shown and the implication of their results discussed. Sec 7.2.1 presents and compares results of the $\mathcal{A}$CNN previously benchmarked using real data, with synthetic μDoppler data using both Per-CPI and Sub-CPI motor speed sampling. Sec. 7.2.2 extends this by including synthetic μDoppler data from a number of lower fidelity motor-speed options. Finally, Sec. 7.2.3 repeats the results with using half real and half synthetic μDoppler data.

7.2.1 Synthetic Micro-Doppler Training Results

An identical training procedure as described in Section 5.2.5 for the real ADB dataset was used with the synthetic datasets. Nine instances of repeated training on each of the one real and two Per- and Sub-CPI synthetically trained classifiers occurred, with the same random seeding as each other; identically as was done in Chapter 5. The same Bayesian optimised parameters as found for the real dataset were used for synthetic training. The training and validation set contained only synthetic μDoppler data, so the results presented are achieved without the use of any real radar data from large drones - only their motor speed and trajectory recordings. With this, the use of real *Mini* data is present in the three evaluation subsets, that did not possess any μDoppler to be simulated. It is also outside the scope of the simulation study to perfectly simulate the body returns, but this would require much more care to accurately sample the GPS recording and could benefit from using real-time kinematic positioning [22] to allow a precise simulation of sub-CPI integration of radial measurements. This idea would also apply to large drones, but as μDoppler is the dominant classification feature, it is the primary concern for simulation[1]. The three classifier results are shown using the nine repeated models with the same initialisation seeds as for the real benchmark in Figure 7.2.

[1]Sec. 8.2 contains results where the training set exclusively comprises simulated data of the LDB dataset containing only real birds and synthetic drones



(a) Best Real Confusion Matrix (b) Best Sub-CPI Confusion Matrix (c) Best Per-CPI Confusion Matrix

(d) Boxplot summary of results of the three classifiers for all 9 training instances

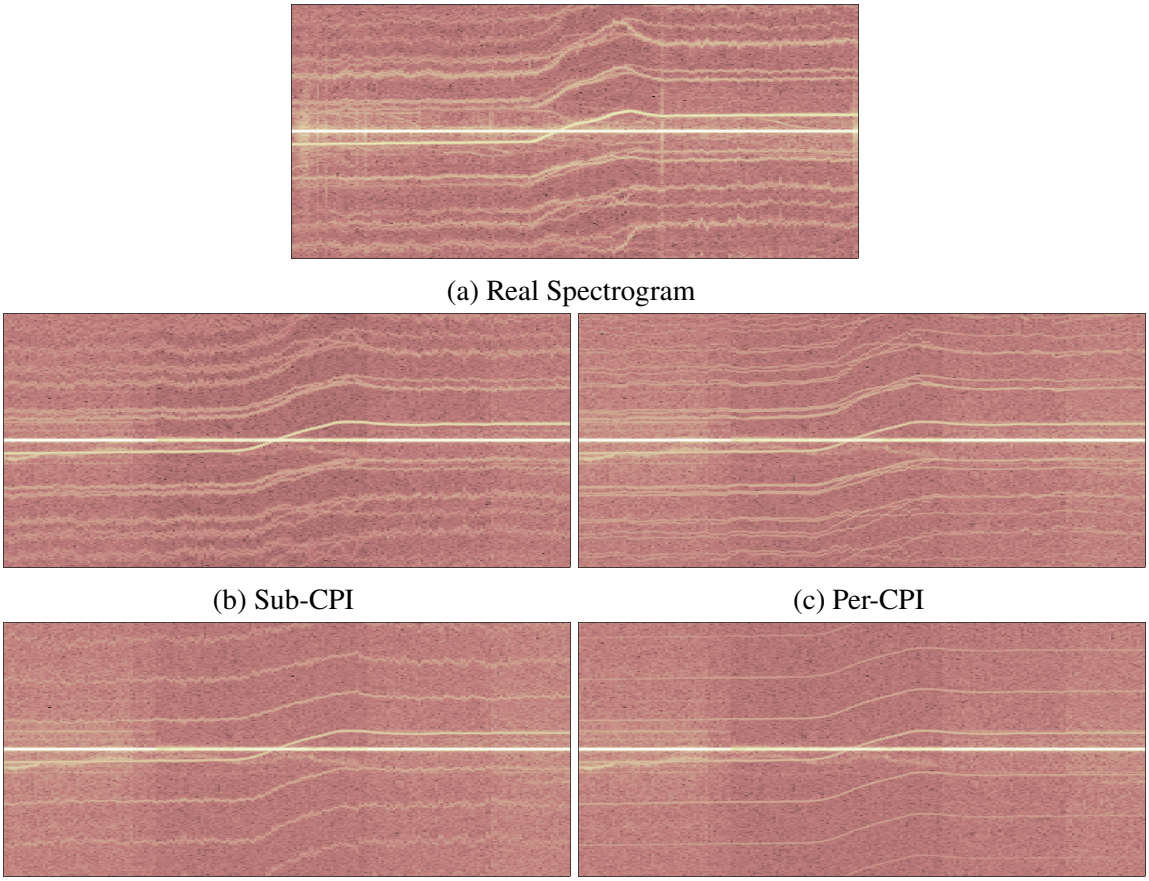
Figure 7.2: Summary of ADB synthetic training results.

The ~90% real benchmark that was biased towards birds slightly surpassed both synthetically trained models that were at ~87% and ~86% for the Sub- and Per-CPI simulations respectively. This shows that the synthetic data was successfully able to induce training for the classification of unseen real targets. The fact that performance using idealised synthetic data was only 4% from the real data that possessed signature variability and fluctuations as measured by a real system in a real urban environment, shows that synthetic data can be used to support classification in a meaningful way without the classifier ever seeing real drone μDoppler. The bias towards birds did grow stronger using synthetic data, with drone recall, i.e. the likelihood that a drone prediction was true, falling from ~88% to ~80% and ~78%. It is noticed that the performance bias towards birds found for the real dataset was greatly exacerbated for the synthetic dataset,

increasing the difference of recalls from 2.5% to 13% and 16%. These result that were trained without any real μDoppler data demonstrate the potential of the proposed approach, with some reservation. Further work can continue to improve the fidelity and variety of data, with the utilisation of greater quantities of data that is one of the key motivations of using synthetic data in the first instance. The following section provides a greater expansion on these results with lower fidelity versions of these simulations to bring greater insight to how synthetic data fidelity effects classifier performance, which was also reported in [3].

7.2.2 Effect of Synthetic Data Fidelity

The previous section provided encouraging evidence that the $\mathcal{A}$CNN classifier trained with synthetic data can give comparable results to one that is trained on real data. In this section we explore further how the simulation fidelity can impact upon the classification performance. As well as the Sub-CPI and Per-CPI models, two other lower fidelity synthetic datasets were constructed that employed the same simulation parameters defined in the previous Section 7.1. The only difference between any of the synthetic versions is the choice of motor speed sampling. Firstly, the Sub-CPI model was repeated, but just one of the four motors were considered. This is referred to as the *Single-Motor* model. Also, the approach to simulate HERM lines with four motors of identical speeds was done and tested on real data providing context for the reporting in the literature of this simulation approach [119], [122], [129], [247], [276], and will be referred to in this work as the *Fixed-Speeds* model. The fixed rotational frequency was constant across the flight and sensible values within a range typical of the drone model was randomly chosen for each flight. The five models trained in the following section have their spectrograms shown for the *Inspire* and Radar#1 in Fig. 7.3. The difference in spectral appearance between the Single-Motor (Fig. 7.3d) and Fixed-Speeds (Fig. 7.3e) clearly shows the effect of Sub-CPI sampling. To any human interpretation of these spectrograms, the multi-rotor Per-CPI is much more similar to the real data than the Sub-CPI, Single-Motor example. However, this did not translate directly the results achieved.



(a) Real Spectrogram

(b) Sub-CPI

(c) Per-CPI

(d) Single-Motor

(e) Fixed-Speeds

Figure 7.3: Fifty-Six seconds of Real and Synthetic Spectrograms for the *Inspire* at 2.5 km range with Radar#1.

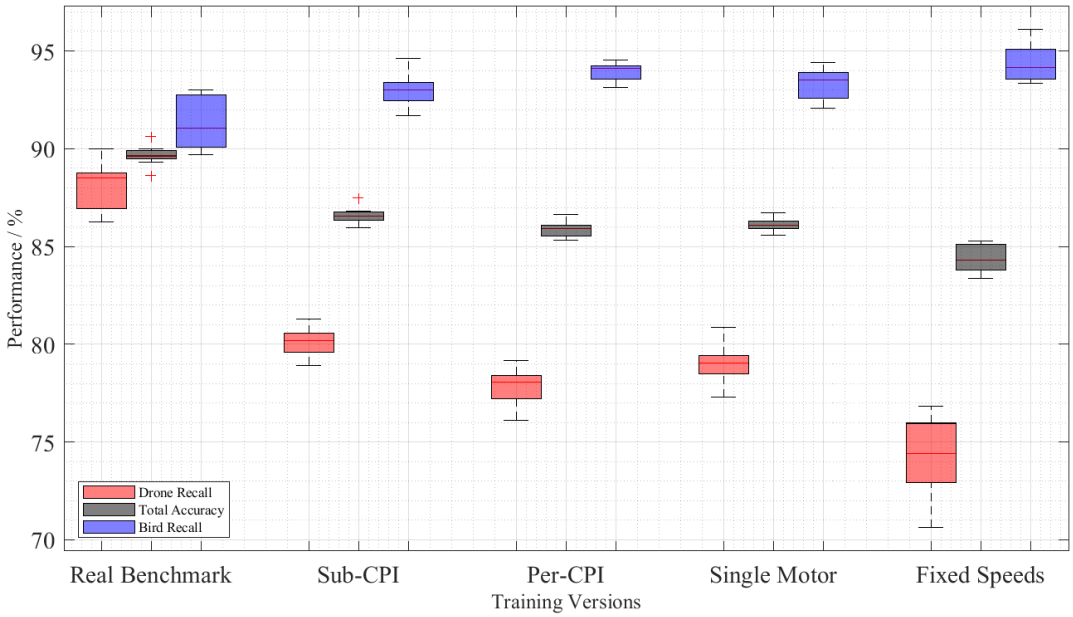


Figure 7.4: Summary of ADB synthetic training results with all synthetic data fidelities.

The real baseline achieved a mean value of 89.7% for total accuracy, with drone recall (88.0%) and bird recall (90.5%) worse than for drones by 2.5%, which is a reasonably balanced performance slightly biased towards selecting edge cases as birds. The best performing synthetic model was the Sub-CPI version, which at 86.6% was within 3.1% of the best performing real model. The Per-CPI and Single-Motor synthetic versions were ~1.5% lower at 85.9% and 86.1% respectively, and the Fixed-Speeds model's mean total accuracy fell a further 1.4% to 84.4%. The median and spread of accuracy of the synthetic models are approximately identical for the multi-rotor Sub-CPI and Per-CPI and Single-Motor, Sub-CPI model but the performance of the fixed-speeds fell significantly further, demonstrating that motor speed variation is an essential component of information in the drone spectra. Observing the drone recall as a function of the μDoppler fidelity, the reliability of drone classifications decreased rapidly to a mean value of 74.5% for the least accurate, Fixed-Speeds model. The median drone recalls of the two multi-rotor models incorporating all four motor speeds fell, from the real baseline of 88.4% to 80.1% (Sub-CPI) and 78.8% (Per-CPI). The Single-Motor version's drone recall was between these values at 79.0%. Each of these models had mean drone recalls to within 2.1% of each other, with the final, fixed speeds model considerably worse at 3.5% from the next best recorded recall.

The decrease in drone recall against training data fidelity signifies the model is becoming over-fitted to the shape of the synthetic sidebands and struggles to recognise real sidebands as effectively. The degradation of drone recall with reduced synthetic data fidelity shows that the more realistic the simulated data, the more accurate the classifier is, crucially producing less drone false positives. Bird recall increased for the two multi-speed multi-rotor models from the mean real baseline of 90.5% to 93.0% for Sub-CPI and 93.9% for Per-CPI. Again, the Single-Motor version sat between these values at 93.3%, with the lowest fidelity Fixed-Speeds version ~1% above the alternatives. This indicates that synthetic training with both motor speed variation and a multi-rotor model minorly reduces the models bias towards birds. As the synthetic fidelity decreases, the bird recall gets higher, thus showing that the model is biased

Table 7.3: Summary of ADB synthetic μDoppler training results for all synthetic data fidelities.

Training and Validation Data	Mean Accuracy	Mean Drone Recall	Mean Bird Recall	Mean Large Drone Accuracy	Mean Small Drone Accuracy
Real Baseline	**89.7 $\pm$ 0.5**	**88.0 $\pm$ 1.2**	**90.5 $\pm$1.3**	**94.0 $\pm$ 1.1**	**73.3 $\pm$2.2**
Sub-CPI	86.6 $\pm$ 0.5	80.1 $\pm$ 0.8	93.0 $\pm$ 0.9	85.4 $\pm$ 0.7	67.3 $\pm$1.4
Per-CPI	85.9 $\pm$ 0.4	77.8 $\pm$ 1.0	93.9 $\pm$ 0.5	83.0 $\pm$1.0	65.4 $\pm$2.1
Single-Motor	86.1 $\pm$ 0.1	79.0 $\pm$ 1.0	93.3 $\pm$ 0.8	84.2 $\pm$1.0	66.6 $\pm$ 1.8
Fixed-Speeds	84.4 $\pm$ 0.7	74.3 $\pm$ 2.0	94.5 $\pm$ 1.0	78.3$\pm$2.6	64.8 $\pm$ 2.2

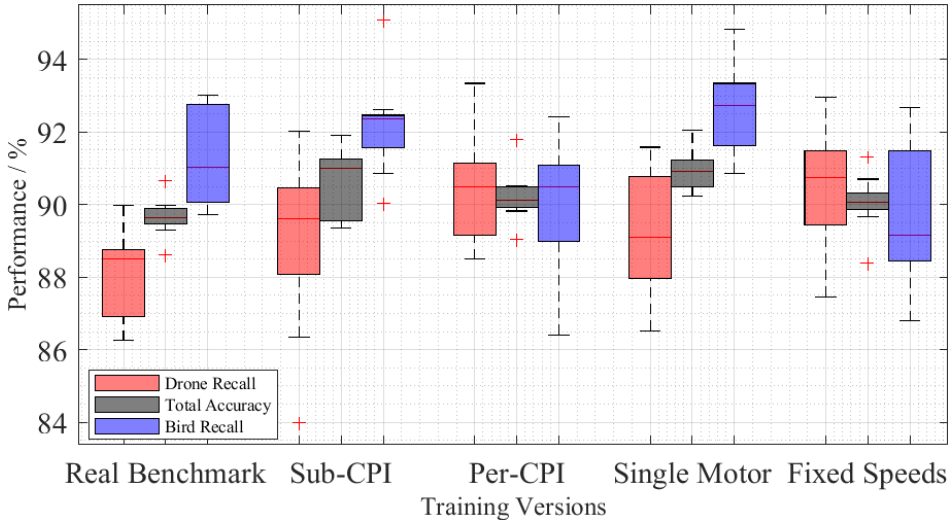
towards predicting difficult targets as birds, which is an operationally ineffective outcome. Table 7.3 also displays the accuracies of the small and large drones in the dataset averaged over all 9 trained classifiers. The *Mini* drone was always misclassified significantly more frequently than the large μDoppler producing targets. We observe that the small drone accuracy decreases across the datasets at a similar rate as the overall drone recall despite itself not changing in the training sets. This shows that the inclusion of synthetic μDoppler data can be added alongside real small drone data with a reasonable degradation of their performance, that follows the trend of motor speed fluctuation being more important than multiple motors.

7.2.3 Training with Real and Synthetic Data

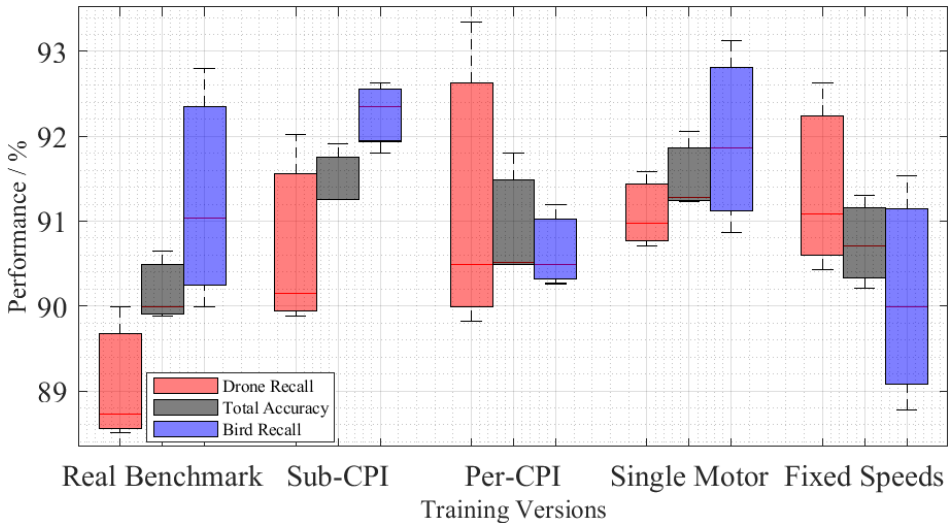
As real small drones were included in the previous study, it is also of interest to know if synthetic and real μDoppler data can be mixed in training. The process of the previous section was repeated with half of the large drone target training data as real, and half using the synthetic μDoppler dataset described previously. The order of exchange was to use alternating real and synthetic training samples for every flight, so the dataset was the same size and included equal representation of real and synthetic μDoppler samples for every flight.

Table 7.4: Composition of All Drones and Birds (ADB) data subsets. *Synthetic Data

Target	Train Samples	Test Samples	Validation Samples
Bird	3625	1818	371
Inspire	722	612	-
*Inspire**	722	-	127
Matrice	529	671	-
*Matrice**	529	-	170
Mini	1129	535	45
Total	7256 (2380 + 1251 + 3625)	3636 (1818 + 1818)	713 (45 + 297* + 371)



(a) Best 9 Accuracies



(b) Best 3 Accuracies

Figure 7.5: Synthetic Training comparison with half real and half synthetic simulated data with different numbers of best models assessed.

Table 7.5: Summary of ADB half real and synthetic training results with all synthetic data fidelities.

Training and Validation Data	Mean Accuracy	Mean Drone Recall	Mean Bird Recall	Mean Large Drone Accuracy	Mean Small Drone Accuracy
Real Baseline	**89.7 ± 0.5**	**88.0 ± 1.2**	**90.5 ±1.3**	**94.0 ± 1.1**	**73.3 ± 2.2**
Sub-CPI	90.3 ± 0.6	89.7 ± 0.9	90.9 ± 1.0	95.0 ± 0.6	77.1 ± 2.0
Per-CPI	90.5 ± 0.2	88.9 ± 1.1	92.0 ± 1.3	94.7 ± 0.7	75.0 ± 2.5
Single-Motor	90.6 ± 0.4	89.1 ± 0.6	92.2 ± 0.8	94.8 ± 0.4	75.5 ± 1.6
Fixed-Speeds	90.2 ± 1.4	89.5 ± 1.3	90.9 ± 2.2	94.6± 1.3	77.4 ± 2.0

Figure 7.5 shows the results for the five training datasets (1 real and 4 synthetic μDoppler) with both, all of the nine classifiers trained in Fig. 7.5a, and also with only the top three accuracies within these in Fig. 7.5b. A summary of results including individually the large and small drones are shown in Table 7.5, which with comparison to Table 7.3 shows in this case that large drones, small drones and birds *all* had their individual accuracies increased from the real baseline. The mean of best three accuracies for all the half real-synthetic datasets outperformed the mean wholly real trained results, emphatically for the Sub-CPI and Single Motor datasets (by 1.2%), and marginally better for the Per-CPI (by 0.5%) and Fixed Speeds (by 0.7%) datasets. Seeing that removing a portion of real data and replacing it with synthetic data improved the classification accuracies is an interesting result that further validates the merit of simulation approach used in this work.

This result could be attributed to the fact that addition of synthetic data that was similar but not identical to the real existing data, increased the overall variability of data - which led to better training - even if the synthetic data itself had less variability than the real data. This motivates us to believe that using synthetic data with a wider range of simulation parameters to further increase the variability of the dataset will lead to better classification results, by better allowing the classifier at test time to interpolate across the learned features of the training dataset. The balance for the better performing Sub-CPI and Single-Motor datasets were, like the real baseline,

generally biased towards birds. The Fixed Speeds and Per-CPI classifiers had similarly spaced accuracies but their bias towards either category was not consistently towards birds. Compared to the prior studies that showed minor variation in accuracy and relative balance across synthetic classification training datasets, the variation and direction of imbalance were less repeatable. This is likely in-part due to the fact that validation data contained only synthetic μDoppler, but training was with a mix of data. The comparison of spread of the box plots to the results in the previous sections that were trained wholly on synthetic or real μDoppler data indicates that mixing data causes the classifier to learn in a more stochastic manner.

The Sub-CPI and Single-Motor data both contained Sub-CPI motor speed sampling. The results of this experiment generally suggest that motor speed variation is the more crucial factor to synthetic μDoppler data's effectiveness. Interestingly, the model with absolute best performance was the Single-Motor, Sub-CPI model beating the multi-rotor sub-CPI accuracy by a negligible 0.05% which corresponds to two 5.6s spectrogram samples, which despite their obvious omission of rotor duplicity, did not prevent it from claiming the top spot of this systematic study. The multi-rotor sub-CPI simulation with larger averaging of classifier models did perform better in this and the other studies presented. Nevertheless, their similar performance across the problems strongly suggests that the Sub-CPI implementation is not optimally implemented. The likely source of these observations is the lack of variety of μDoppler within flights, and across the spectra. The following, final section will discuss the benefits of the proposed approach in context with the results shared in the previous three sections, and where the application with further modifications to the simulation approach could be a key enabling tool for radar based drone classification.

Table 7.6: Results of synthetic training in Section 7.2..

Training Dataset	Best Accuracy	Mean 9 Repeats
Half Sub-CPI	91.9	91.0
Half Per-CPI	91.7	90.2
Half Single Motor	92.1	90.9
Half Fixed Speeds	91.3	90.1
Real	**90.6**	**89.7**
All Sub-CPI	87.5	86.6
All Per-CPI	86.0	85.9
All Single Motor	86.8	86.0
All Fixed Speeds	85.2	84.3

7.3 Chapter Summary

The results of this chapter are summarised in Table 7.6. The proposed synthetic dataset enabled training of $\mathcal{A}$CNN tested on 3,636 real, urban drone and bird spectrogram samples from two radars differing in transmit power and system noise, to 86.6% accuracy. This was within 3.1% of the equivalent results with training classifier on only real data that yielded 89.7% accuracy. This was achieved with replacing the data of drones that exhibit μDoppler with simulated data in training and validation, while intentionally not availing of any of the key benefits that come with high fidelity synthetic data, that are increased quantities and increased variation of data used. Alongside this, a 50:50 combination of the synthetic and real data was able to improve the accuracy beyond the real benchmark by 1.3% to 91% under the same conditions. This demonstrated that inclusion of synthetic data as presented was able to directly benefit classification.

Similar performance using the Sub-CPI, multi-rotor and Sub-CPI, *Single-Motor* datasets compared to the drop-off with the multi-rotor Per-CPI and Fixed-Speeds dataset, showed that it is more important to have Sub-CPI sampling - or at least some variation of motor speeds across the temporal pulses of a CPI - than to model multiple rotors within the simulation. This was further highlighted in the study with real and mixed data in Sec. 7.2.3 through the closeness of single

and multi-rotor performances with the Sub-CPI technique, despite the large difference with the Per-CPI approach that used multiplicity of rotors, and would be expected to be far superior when visually comparing the resultant spectrograms as was in Figure 7.3. The multi-rotor Per-CPI approach when wholly training the classifier was on average 0.2% less accurate than the Sub-CPI Single-Motor implementation. The Sub-CPI multi-rotor approach was by far the best in terms of visual fidelity, but statistically only marginally the best when averaged across classifier accuracy for all repeated training occurrences. This raises a question why performance improvements are not matched in the $\mathcal{A}$CNN compared to the apparent fidelity upgrade when examined visually.

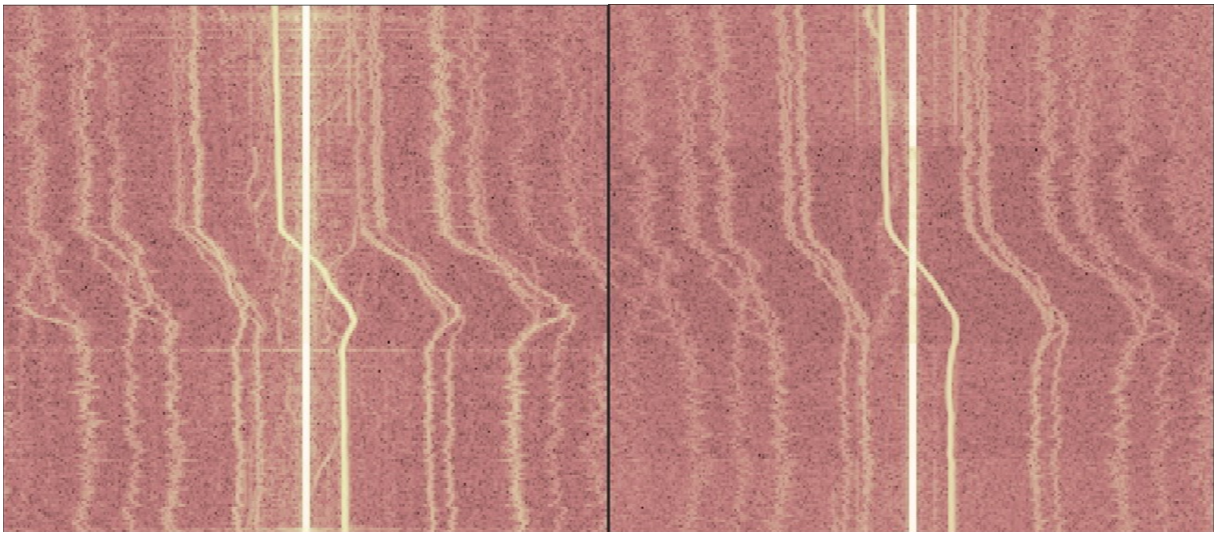


Figure 7.6: Side-by-side real (left) and synthetic Sub-CPI (right) spectrogram of a drone manoeuvre of the *Inspire*.

Figure 7.6 shows the real and the Sub-CPI MRμD synthetic spectrograms used in the study. The global frequency modulation of the sidebands is captured perfectly, and the local spectral width and variation across successive CPIs of the simulated sidebands are resembling those real sidebands where the SNR is strong. The most noticeable difference of the two models is the amplitude scaling of the sidebands as Doppler offset from the main body increases. Frequently and clearly in this real example, the strength of sidebands does not always decrease smoothly with increasing Doppler offset. The variation of amplitude scaling set by the α parameter in

the simulation could be improved with a modified function that factored in the motor speed to attenuate the power as a function of harmonic and not as a linear function of frequency, or also to vary the return strength from individual motors to mimic the unpredictable scattering of each motor, that is observed with real data. The rolling average used in the Sub-CPI process also could be varied across motors to more closely follow the pattern found in real spectra. The surprisingly strong performance of the Sub-CPI, Single-Motor dataset for the 50:50 real-synthetic mixed training results could be attributed to the increased variation within the training data compared to the multi-rotor versions that were more self-similar across all samples. Using a less restricted variation of simulation parameters, as was used in this work, would also increase the variation of the training dataset, and in addition by using increased quantities of data with this method, could allow the training of robust classifiers using only target-less radar recordings and drone logs. Future work seeking to improve the fidelity of spectrograms may additionally inspect the phase of simulated returns with comparison to real measurements, such as bi-frequency based time-frequency plots [15], [277]. Analysis using such techniques, may offer avenues to better analyse and iteratively improve the simulation fidelity but was not considered within the scope of this work.

The simulation techniques introduced in this and the previous sections will have great implications for other radar systems beyond the L-band staring radar used here. For radars with higher sampling rates, more dynamism of the motors will be coherently processed and so more care in the Sub-CPI resampling of motor speed recordings will be required to ensure the fluctuations of real drones are appropriately addressed, either by control of sideband width with the rolling average used here or by another more effective approximation of motor rotational frequencies. Higher frequency radar systems will have more dynamic and complex propeller scattering across different blade lengths and shapes [278], and so the sideband attenuation method would need to be tuned accordingly to suit the desired fidelity.

The limited extent of background recordings used for the synthetic data was discussed, and additional variation would further improve the results in this section generally. The process

allows a method for operators to collect the known ambient data from the local environment, and specify the training data to cover all the locations anticipated in the setting, thus providing a degree of performance guarantee. Creation of synthetic μDoppler will require a validation process with real data and a variety of targets to understand the parameters required to produce the best dataset for their specific application, and this and the prior chapter provided a skeleton approach to enabling this. Sec. 8.2.1 of the following chapter will feature further use of the defined synthetic data approach in investigations with unsupervised learning and convolutional neural network approaches.

Chapter 8

Unsupervised Spectrogram Classification

This final research chapter of the thesis explores unsupervised learning, and specifically the application of a Convolutional Autoencoder (CAE) for classification of drones and birds using L-band radar returns. This chapter explores how a CNN classifier can be built using radar data alone without having to rely upon optical pre-training, and includes an inspection of the unsupervised learning results that can potentially offer a method to better understand the origins of classifier performance.

The beginning of this chapter, Section 8.1, briefly discusses the existing knowledge and where the approach has been proven effective in recent radar literature. In Section 8.2, an elementary autoencoder architecture is used to explore the bird and drone L-band staring radar data that was collected in this project and used throughout the work on supervised classification reported in Chapter 5 and 7. Self-supervised representations learned with a basic deep learning CAE are compared to other, machine learning dimension reduction techniques of PCA, tSNE and UMAP. The potential of autoencoder pre-training to benefit classification of bird and drone targets compared to optical transfer learning is demonstrated. Additionally, an intrinsic inspection of the unsupervised autoencoder performance is performed, which may help in selecting the optimal pre-training state that will better enable supervised classification.

8.1 Introduction to Autoencoders

Previously, Sec. 3.2.5 briefly overviewed autoencoders for radar applications, but for brevity and due to the great scope of work that has been performed elsewhere, many details were reserved for the present, dedicated section. This section will begin by introducing a basic CAE concept via its neural network structure in Sec. 8.1.1, with the autoencoder principle and the useful properties that can be achieved with appropriate training. Learning with autoencoders is more challenging due to the much more complex gradient decent challenge in unsupervised learning, thus many extensions of the basic model are commonly utilised to improve autoencoder latent representation in all domains, including recent works with radar data where it has been shown to combat the challenge of data availability. Four of these common extensions and radar works that utilise them are reviewed in Sec. 8.1.2.

8.1.1 Architecture and Outputs

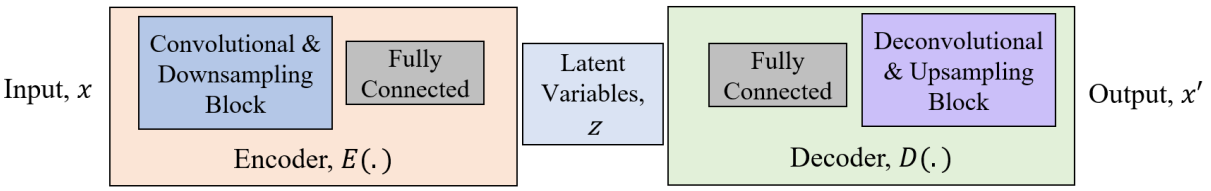


Figure 8.1: An example symmetrical CAE schematic.

A basic autoencoder is a neural network that has an input and output that are the same size and dimensionality. Figure 8.1 shows an elementary (convolutional) neural network structure arranged as an autoencoder, where the input, x, and output, x', are both the same shape. This feed-forward network is trained as usual with backpropagation. The loss function however is not dictated by the label value. Instead, learning is based on the error computed by comparing how far away the output values are from the input values, i.e., how dissimilar x and x' are. Thus, as training progresses, the output will morph from unintelligible noise to *reconstructions* of the original input[1].

[1]This is shown later in Sec 8.2.3 with Figure A.6 and A.7, for the ADB radar dataset.

The encoder of the CAE is effectively indistinguishable to the CNN classifier architecture. Oppositely, the decoder of a CAE needs to perform dimensionality expansion from the latent variables, z, to the size of the original input, all in a back-propagatable way. The decoder contains inverse functions of those seen in the encoder. These are, transpose or *de*convolutional layers [279] replacing convolutional layers; max-*un*pooling [280], [281] or alternatively up-sampling replacing max-pooling. Fully convolutional autoencoders have been implemented on a Field Programmable Gated Array and were shown to produce effectively perfect reconstructions of optical image data [282]

Autoencoders can be under- or over-complete. An AE is under-complete if the hidden layer size is less than the input dimensionality, where the layer of the smallest bottleneck is referred to as the latent variables, z. In this more frequent configuration the network is forced to, in the ideal case, learn a compressed representation of the data [195] where key, pertinent features of the entire training set are extracted and may be utilised to efficiently process similar, unseen samples. An undercomplete CAE effectively performs neural-network based dimensionality reduction that can be suitable for potentially very large and complex data. Other linear (PCA) or non-linear (tSNE, UMAP) dimensionality reduction algorithms can do this with significantly less implementation effort, however, generally their use depreciates as the size and variety of a dataset grows. The deep learning approach learns highly-relevant and abstract non-linear features that make it a powerful compression algorithm that can be used for image [283], [284] or audio [285] compression, of which such applications would be of great interest to radar operators that seek to reduce the bandwidth of data transfer between platforms - of which research in automotive radar has already been demonstrated [186]. The additional motivation for autoencoders is that as they are simply neural network structures, after an initial training they can be modified and subject to further training to utilise the information encoded in the first stage for a more precise application task [192], of which classification is one example.

Latent representations are powerful tools in machine learning. As an initial training stage, an autoencoder can be optimised for a dataset. Then, taking multiple autoencoders that act on a

certain *modality* or *view* of data, and stacking or fusing of the learned representations with final training in an end-to-end manner on the complete dataset, brings together separately validated components for a wider, more context aware model. This act of *fusion* or *multi-modal* learning [286] is a huge part of modern AI development, and has benefits to bring to radar situational awareness as has already been shown in a few selected radar works, including [152] using bird and drone data and [287] on SAR imagery.

The output of a basic autoencoder, $AE(.)$, can be summarised by the following equation,

$$x' = AE(x) = D(E(x)) = D(z)\,. \tag{8.1}$$

A basic autoencoder attempts to reproduce the original image via a loss function that punishes the difference between the two [285]. The overall loss is summed or averaged across every shared pixel in x and x' and this is back-propagated through the network in the same manner described in Sec. 3.2.1. As a basic CAE learns from the training set, the output x' will take the appearance of the original sample, provided the learning rate is appropriately set so that the optimisation gradient does not explode or vanish. With this achieved, the quality of learning is not guaranteed but can be scrutinised in a number of ways. At first, if unseen test samples are not faithfully reproduced then the model has become severely over-fitted to the training data. However, even given that unseen test samples are accurately reproduced, it is possible the network has learned the identity function by effectively mirroring the parameters in the encoder and decoder halves with off diagonal matrix components set to 0, and the learned diagonal weights small enough to absorb non-linearities present in the activation functions. Further literature expressing and combating this effect are in [288], [289].

Thus another alternative view of autoencoder behaviour for a dataset is to observe the latent variables z for samples in the dataset x where $z = E(x)$. If some or all of the data is labelled, the distribution of latent variables can be compared to the class label. Going further, the distribution of samples in the latent space can be scrutinised with respect to apparent features of the samples,

which could be done in the absence of labels. Plotting of the latent variable distribution can be used to visualise and test the encoder if convolutional features were learned from the dataset that respect the differences between samples' semantic labels. It will be of interest to observe features relevant to the radar target data such as presence of μDoppler. A separation of such a key feature such as drones' μDoppler motivates further tuning to be used as an initialisation for further training on a downstream task. However, a separation of labelled, latent variables is not the pinnacle of latent representation success. The decoder can be assessed in an even more rigorous test by sampling the latent space, $x' = D(z)$, and observing if realistic samples are generated as the latent space is traversed. Sampling the decoder for all values of z may not produce realistic samples. It is commonplace to display the decoder output for a uniform sample of z in order to show the plausibility of samples that are fabricated by the decoder, which in this context can be referred to as a generator.

Later, Sec. 8.2.2 will show and analyse all of these described outputs with the collected L-band bird and drone staring radar data. Before this, the following section will review autoencoder use in radar literature. As will become apparent, there is limited work using very simple autoencoders aside from [98] and [208] to be published to date that the author is aware of. There is much wider use of common extensions to the autoencoder loss and learning process. All of the upcoming radar citations show the power and effectiveness of autoencoders for a wide variety of applications, with many showing excellent results in classification, even with very small amounts of training data, as well as target clustering, synthetic sample generation and noise removal. What is perhaps missing is a full assessment of the autoencoder's unsupervised learning merit for a basic autoencoder as was described in this section. Understanding the deficiencies of the basic model will enable more targeted future work to make the most of the autoencoder techniques.

8.1.2 Extensions and Radar Applications

The interest in literature for CAE based approaches is great and continuously evolving. Several techniques have become very common in the field of computer vision, and have also been used in radar studies. These recent works have been in the domains of drone detection [194], human activity recognition [99], [150], automotive radar [290], low-bandwidth inter-radar intelligence communications [186], maritime and terrestrial land SAR target recognition [291]–[293], ISAR [151], passive radar [57], weather radar [294], [295], and interference suppression [57], [290]. Four common techniques that these publications used are briefly discussed along with some detail of radar implementation examples.

The Variational Autoencoder (VAE) [296] first described in the PhD thesis [297] and otherwise in [184], [297] raised the ceiling of representation learning greatly and is important to cover when reviewing representation learning. The fact that latent vectors are fixed to what has been learned before inspired improvements to getting the autoencoder to learn a latent *distribution*, rather than learning through fixed latent vectors. The VAE achieves this by injecting stochastic noise alongside the encoder output (where we consider the latent variables, z to not be part of the encoder). Choosing a Gaussian distribution for the noise sampling forces the encoder to learn the features in-tune with the random input. This allows training to optimise the model for reconstructions with an additional penalisation on the encoder's mapping to ensure a representation is learned with a probability distribution that matches the distribution used for sampling. This ubiquitous technique in representation learning has featured in a number of radar publications [99], [147], [150], [186], [291].

Also used to improve the latent representation from basic CAEs are denoising autoencoders [298]. This can implemented as a basic CAE, with a small difference - samples from the dataset are used at the output of the decoder, while the input to the encoder is the sample but with some corruption, such as additional noise, for example Gaussian or salt-and-pepper noise [288], [299]–[301]. This improves the latent representation of the autoencoder as it is not trying to recreate the input exactly, so the risk of learning an identity function is reduced, which forces

the representation learned to use the corrupted input image efficiently to understand the essence of the input producing the output without the noise [302]. Radar applications of this method have been shown with: [194] using a denoising autoencoder to effectively remove the amplitude variation along the range axis for drone detection in a 1-D problem; [254] doing the same but for larger and more complex range-Doppler maps; [290] reconstructing range Doppler maps to an impressively high fidelity after random measurement samples were, discarded simulating FMCW interference in an unpredictable EM environment; and [57] that in passive radar, trained an interference suppression CAE for range-Doppler maps.

Similar in essence to the principle of denoising autoencoders, contrastive learning approaches are another way to learn robust latent representations, by using the autoencoder to translate between different versions or realisations of the same data [303]–[305]. The samples of the dataset are split into an augmented pair, where one of the views could be the original input, such as noise-injection, or for radar, different DSP choices, as long as it is a different representation of the same sample. Then, two encoder networks compress the information to a low dimensional latent space, z. Then after this, another non-linear learnable network such as a few perceptron layers, called the *projection head* [303] is added after z to a new final representation. A loss function is used that maximises the similarities between the final latent vector pair, and simultaneously trains both encoders to maximise similarities of the two learned representations. After training, the projection head is removed after the termination of the encoder, along with the augmented model branch. This produces an encoder only, that could be further trained with a standard AE approach, or directly used for target clustering, classification or other downstream tasks with supervised fine-tuning. This method has been used in radar studies [194] for drone detection and other applications [193], [293], [305].

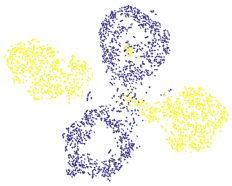
The last technique to be mentioned is the Generative Adversarial Network (GAN) [190], [306]. An encoder-like *Discriminator*, predicts if its input is real or randomly-generated, while a decoder-like *Generator* attempts to fool the discriminator that it accepted a real sample from the dataset. The discriminator is fed a real or synthesised sample at random and backpropagation

trains its parameters at the result of each estimate. This adversarial process reaches a Nash equilibrium, where the capabilities of the generator and discriminator become evenly matched, and can have a runaway increase in performance. The hidden layers of a GAN will possess very robust features that empower its sample generator and discriminator, which although is not exactly of the form of the autoencoder, it is a very powerful method of unsupervisedly training a neural network architecture. A thesis published in 2022 [271] investigated simulation of drone returns, trained a GAN with this data, and created and tested various similarity metrics to assess the synthetic samples that were outputted. However, no examples of their synthetically generated spectrograms were included. Other works show the output of the GAN, where each succeed in capturing many important features of the training data in use cases of human activity recognition [100], [307], a forward looking imaging airborne radar [308] and weather radar data [294].

An immediate use for autoencoders is as a pre-training stage for subsequent fine-tuning via supervised learning for classification. [99] compared SVM on handcrafted features, supervised-CNN, and variational autoencoders, with and without convolutional layers for classification on 12 human activities. Autoencoders have been described as a method of network initialisation for CNN classification in [98] where a basic autoencoder was compared to an optically pretrained classifier showing superior results for a radar data set of only 700 samples. [193] conducted a similar experiment that compared the same variational CAE with a contrastive learning frame work. [254] uses a denoising autoencoder to reduce the clutter variation of a range-Doppler map in classifier pre-training, improving results from near random to a much better standard on a challenging dataset of 14 human activities. Additionally, [150] performed classification on the latent variables of a variational CAE with a smaller dataset of 200 total samples. Autoencoder pre-training for SAR target classification has had works with a regularised CAE [309], dual-encoder CAE [151], a variational CAE [291] and U-shaped fusion network [287].

A few works have used fusion networks to simultaneously process different modalities of a radar dataset. Notably, a hierarchical fusion network was created in [152] that was tested on two datasets comprising of airborne targets (drone and bird) and ground targets (humans, wheeled

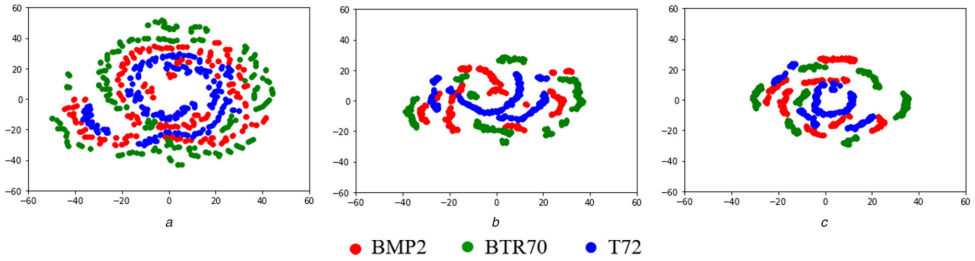
and track vehicles). They used four domains of radar data (time-domain, frequency-domain, time-frequency maps and eigenvalue spectrum) and found that fusing the representations of each of the domains under effect of an autoencoder (hierarchical learning classifier) was better than separately considering the effect of each domain (multi-domain classifier). They illustrate the representations from each individually trained encoder showing latent scatter graphs with classes labelled by colour for the ground targets dataset (but not the bird and drone dataset), showing that each modality generated a different understanding of the data, that when fused together via backpropagation, was the best performing classifier of 10 alternatives.



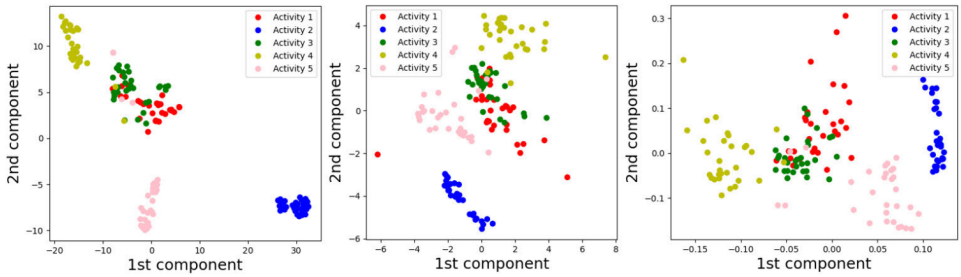
(a) Pedestrians (Blue) and cyclists (Yellow) full dataset [148] with tSNE



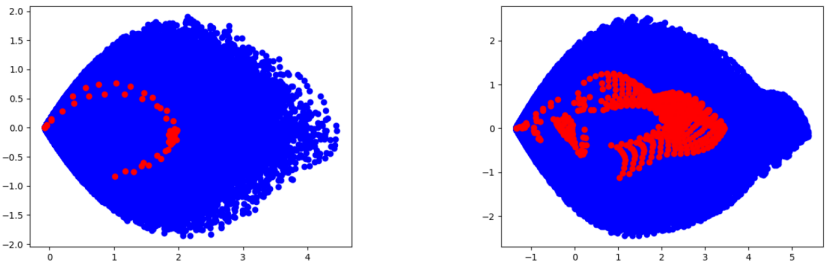
(b) Maritime SAR targets (Red) and non-targets (Blue) [291] of variational CAE and PCA



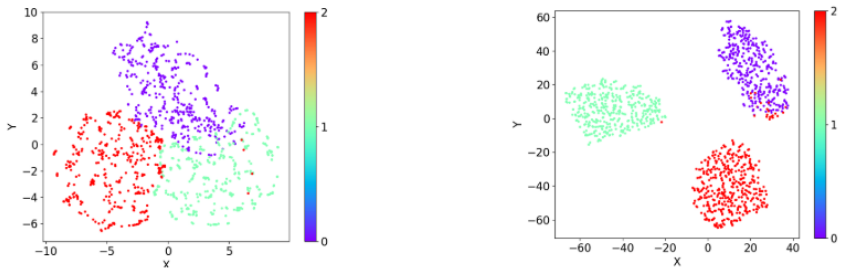
(c) SAR military vehicles with original full dataset, CAE and regularised CAE [151] via tSNE



(d) Five human activities [150] via variational CAE with tSNE, Multi-dimensional scaling and Locally Linear Embedding



(e) Weather radar precipitation phenomena (red) and other (blue) of two full datasets [295] via PCA



(f) Clutter region measurements; original dataset and CVAE decomposition [147] via tSNE

Figure 8.2: Visualisations of datasets and latent representations in existing radar literature.

Several of the earlier cited work in this section included latent representation visualisations. Figure 8.2 shows examples from six published works comparing labelled, scatter plots. Figs. 8.2a and 8.2e both show the entire multi-dimensional dataset decomposed to 2D via PCA and tSNE. Fig. 8.2a via tSNE fairly effectively discriminates the two classes with two main clusters for each class. On a smaller dataset, Fig. 8.2c (left-most) shows that the different vehicles inherently possess a degree of separability that is expanded on the two other autoencoder based decompositions. This idea of decomposing the entire dataset was also used in [310] to compare real and synthetic data, by viewing tSNE decompositions of the gathered features for the whole dataset for each layer in a convolutional network. They showed that there were some differences between the data originally and in the features of the early layers, but by the end of the model processing, the derived features for the real and synthetic samples were strongly overlapping.

Fig. 8.2d shows the decomposition of human activities of a variational CAE with latent dimensionality of 50, and also the raw, 6400-D dataset decomposed with the same 3 techniques, where tSNE shows reasonable separation of the classes that is not too significantly better than the other decompositions in Fig. 8.2d by their own admission. Figs. 8.2b, 8.2c and 8.2f of SAR data had much more data available for the training and validation of their models. Fig. 8.2f [147] used a variational CAE to cluster clutter originating from three different regions of terrain that air vehicles would need to be detected against. Slightly more difficult a task was attempted in [291], shown in Fig. 8.2b, that used a standard variational CAE on marine SAR data of ships and no targets. The aim of their work was to apply a clustering algorithm to the resultant latent space to perform classification, which they did via k-Means to 95.47% accuracy, and by Gaussian Mixture Models to 95.73%. This latent representation bottleneck had a dimensionality of 256, which after decomposition to 2-D showed a impressive and reasonable separation of the classes, with ‘No Ships’ tightly clustered and ‘Contains Ship’ extended out from this in two directions. Another use of this approach with SAR military vehicle targets (MSTAR dataset [311]) was conducted in [151], and has autoencoder created latent representations shown in Fig. 8.2c.

8.2 Autoencoders with Bird-Drone Radar Data

Now that the fundamental concepts of autoencoder applications and implementations in radar have been reviewed, the remainder of this chapter will use the urban bird and drone L-band spectrogram collected in Chapter 4 for an assessment on the practicability of using autoencoders for safety critical radar tasks. At first Sec. 8.2.1 will perform a comparison of classification performance between the optically pre-trained image classifier AlexNet ($\mathcal{A}$CNN) extensively benchmarked in Chapter 5, with radar only unsupervised autoencoder pre-training and fine-tuning on the same dataset ($\mathcal{R}$CNN). The training and testing is further expanded using synthetic data (as was used in Chapter 7) to extend the analysis of the comparison of the two approaches and assess the utility of the CAE architecture as an alternative to the more established transfer learning from the optical domain approach.

Section 8.2.2 will investigate unsupervised learning results with a basic CAE on their own merit. Here the focus is to perform a thorough analysis of achieved autoencoder latent representations through the three outputs of its operation described in Sec. 8.1.1: annotated 2-D latent scatter labelled by class colour, reconstructions of seen and unseen input images, and sampling the decoder to produce synthetic examples. This analysis is shown for both the LDB and ADB datasets and, also a bird only dataset. The goal is to illustrate if and in what regard serviceable latent representations can be achieved using a very simple CAE.

Finally, Section 8.2.3 will bring together these two desirable aspects of autoencoders - supporting classification as CNN initialisation and benefits in interpretability - through a presentation of supervised classification results as a function of the quality of the unsupervised representation, to provide an analysis of optimising the CAE to benefit downstream classification results.

8.2.1 Radar Classification with Unsupervised Pre-training

Autoencoders are known to be an effective method for pre-training deep neural networks before fine-tuning for the downstream task of classification in machine learning [192]. Radar

applications with real and augmented data have used this approach with success in other data limited domains [98], [312]. This section will compare the classification performance of a radar pre-trained CAE and an optically pre-trained CNN binary classifier, using real data to begin with. This analysis is then extended to realistic synthetically generated spectrograms to allow for the assessment to be made on a greater variety of data. Comparing optical pre-training with initialisation from a basic CAE was also attempted in [98], where a 3 convolutional-layer basic CAE and an equivalent CNN without initialisation, was compared with VGG and GoogLeNet that comprise 16 and 22 convolutional layers. Their experiment with amounts of training data ranging from 200 to 700 total samples observed the CAE to provide the best accuracy at the large training data limit. This challenging task shows an advantage to CAEs that this approach has merit, and this section performs a similar test but with a different application *and* perspective of the classification task. Their experiment occurred with smaller train (<800) and test (<250) set sample counts with clean, indoor measurements of 12 human activities, while this work uses larger datasets (>5000 and >2500) of a 2 class problem with a much greater level of wider inter-class signature variation.

Table 8.1: Quantities of data used in the LDB (left) and ADB (right) for training of $\mathcal{A}$CNN and $\mathcal{R}$CNN.

Target	Train Samples	Test Samples
Bird	2572	1303
Inspire	1558	618
Matrice	1012	676
Mini	-	-
Mavic	-	-
Total	5142 (2570 + 2572)	2597 (1294 + 1303)

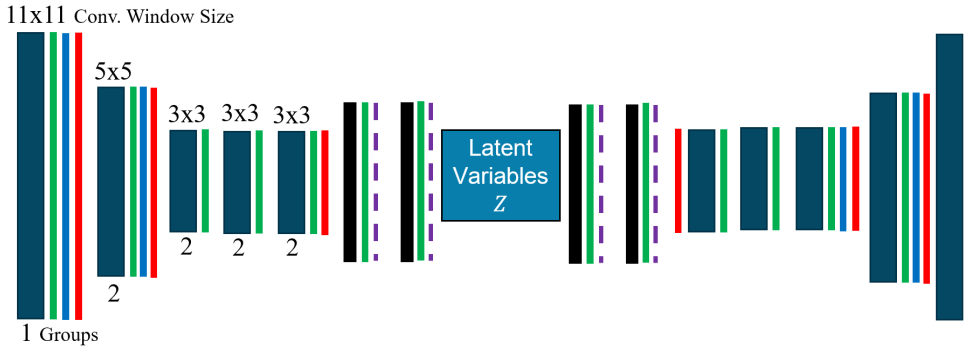
Target	Train Samples	Test Samples
Bird	4465	2250
Inspire	1450	968
Matrice	1410	482
Mini	1235	626
Mavic	367	177
Total	8924 (4462 + 4465)	4503 (2253 + 2250)

We will consider two datasets: Large Drones and Birds (LDB) and All Drones and Birds (ADB). Large and small drones are again distinguished by if their returns were accompanied by rotor micro-Doppler as described in Sec. 4.1.2 and used in Section 5.2. The targets used and quantities of each are shown within Table 8.1. We begin by creating a CAE that is of similar dimensionality

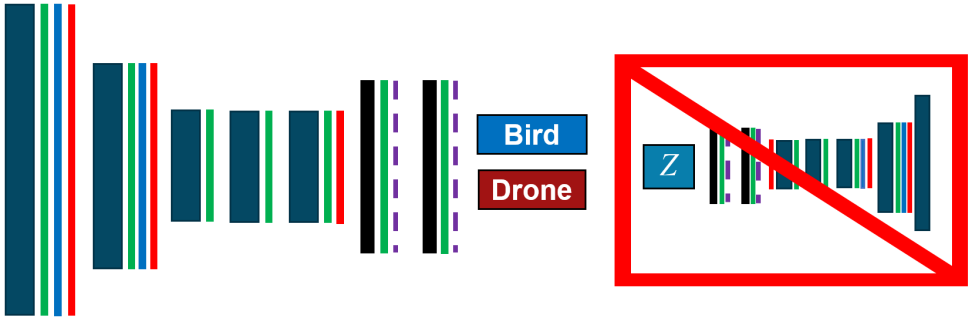
to the optically trained transfer-learned model $\mathcal{A}$CNN benchmarked previously in Chapter 5 with real data, and Chapter 7 with synthetic data. To draw reasonable comparisons with the AlexNet CNN classification result, a CAE from two "back-to-back" AlexNet layer configurations was created. As comparatively far less bird and drone data will be used to train this end-to-end neural network compared to the AlexNet supervised pre-training, we reduce the capacity of the model by creating an AlexNet-like Convolutional Autoencoder (*Alexnet-like* CAE), of which its encoder differs from the original in the following ways:

- One channel input is used, as compared to the original three expecting a RBG input.
- Two groups are used through the whole convolutional depth, as with two classes (compared to 1000) the necessity to force extremely general features is not a strong requirement.
- The 4096 linear neurons in the two fully connected layers are reduced to 1024 here, as the mapping from convolutional features to predicted label is much simpler in the two class and unsupervised case compared to optical training.

With this, Figure 8.3 shows the *Alexnet-like* CAE that will be the largest CAE that will be trained in this work. The 1000 class output is replaced with another fully connected layer comprising 32 neurons. From these new 32 neurons, the second half of the network proceeds inversely to AlexNet, with reciprocal operations where required, such as; transpose or deconvolutions instead of convolutions, max-un-pooling to replace max-pooling (see *Model 4* of [280]), and unflattening rather than flattening to go from the final fully-connected output to the first deconvolutional layer. The order of the decoder layers shown in Figure 8.3 was such that it replicated the dimensionality of the feature maps in the encoder layers, which allowed an arbitrary CNN classifier structure to have a symmetrically configured decoder. This was the generic CAE decoder convolutional layer configuration used throughout this work.



(a) CAE Constructed Similar to AlexNet



(b) Converting a CAE into a CNN classifier for supervised training.

Figure 8.3: CAE and $\mathcal{A}$CNN Classifier Schematics.

The conversion of the *Alexnet-like* CAE to classifier was done by first, taking all encoder layers to before the latent variables and adding on the class outputs, then fine-tuning the classifier in a supervised manner with the same radar training data. This CNN classifier trained exclusively on radar data is referred to in the remainder of this work as the Radar (pre-trained) Convolutional Neural Network Classifier ($\mathcal{R}$CNN), which in this section is the *Alexnet-like* $\mathcal{R}$CNN. Training of the *Alexnet-like* CAE took place with layer normalisation [169], which was replaced when fine-tuning the $\mathcal{R}$CNN layers by local response normalisation [171] as was also used in the $\mathcal{A}$CNN classifier training. Both the *Alexnet-like* CAE pre-training and $\mathcal{R}$CNN fine-tuning were constructed and performed using PyTorch v1.10.2. A challenge within autoencoders and representation learning in general is the number and complexity of interaction between model, training and data hyperparameters. With this, training a model from scratch that is (twice) the size of AlexNet for a 2-class problem and a factor of 1000 less training data would not be postulated as the ideal choice. Thus, to maintain the defined neural network architecture but considering a range of model sizes, the *Alexnet-like* CAE model introduced previously was

Table 8.2: Configurations of the five convolutional depths of CAEs trained

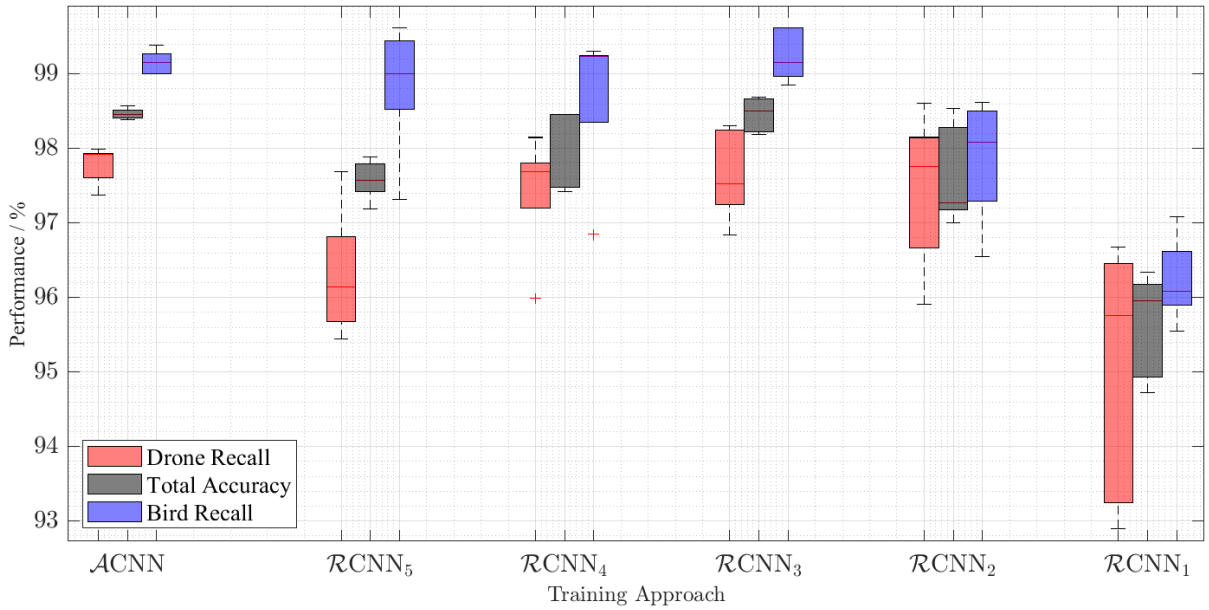
	Model				**Pre-Training**	**Post-Training**
Model	*Conv. Window / Strides / Padding / Groups*	*Max-Pool {window size; stride}*	*# Channels*	*# Conv. Features*	*Learning Rate* $(*\ 10^{-x})$	*Learning Rate* $(*\ 10^{-x})$
$\mathcal{A}CNN$	11-5-3-3-3 / 0-2-1-1-1 / 4-1-1-1-1 / 1-2-1-2-2	{3;2} after 1st, 2nd, 5th	3-96-128-192-192-128	9216	[11]	8 LRs, each repeated twice. *Found per dataset*
$\mathcal{R}CNN_5$	11-5-3-3-3 / 0-2-1-1-1 / 4-1-1-1-1 / 1-2-2-2-2	{3;2} after 1st, 2nd, 5th	1-96-128-192-192-128 and *half*, and *quarter*	9216	6, 5, 4	5, 4, 3
$\mathcal{R}CNN_4$	11-5-3-3 / 0-2-1-1 / 4-1-1-1 / 1-2-2-2	{3;2} after 1st, 2nd, 4th	1-96-128-192-192 and *half*, and *quarter*	13824	6, 5, 4	5, 4, 3
$\mathcal{R}CNN_3$	11-5-3 / 0-2-1 / 4-1-1 / 1-2-2	{3;2} after all	1-96-128-192 and *half*, and *quarter*	13824	6, 5, 4	5, 4, 3
$\mathcal{R}CNN_2$	11-5 / 0-2 / 4-1/ 1-2	{5;4} after all	1-48-192 and *half*, and *quarter*	9216	6, 5, 4	5, 4, 3
$\mathcal{R}CNN_1$	43 / 0 / 4 / 1	{5;4} after all	1-96 and *half*, and *quarter*	9216	6, 5, 4	5, 4, 3

replicated with smaller convolutional widths and depths. A summary of the models and training parameters used for deriving results is as follows. In total, 5 *Alexnet-like* CAE depths were trained with 3 channel widths at 3 learning rates (45 CAEs), from which 3 CAE classifiers were trained scanning the learning rate, totalling 135 classifiers. The full information of model structure and training parameters is shown in Table 8.2, of which every autoencoder had a latent dimensionality of 32. The # Conv. Features column is the size of the input to the first fully connected layer, which was kept as close as reasonably practicable to that of the $\mathcal{A}$CNN with the reduced encoder size.

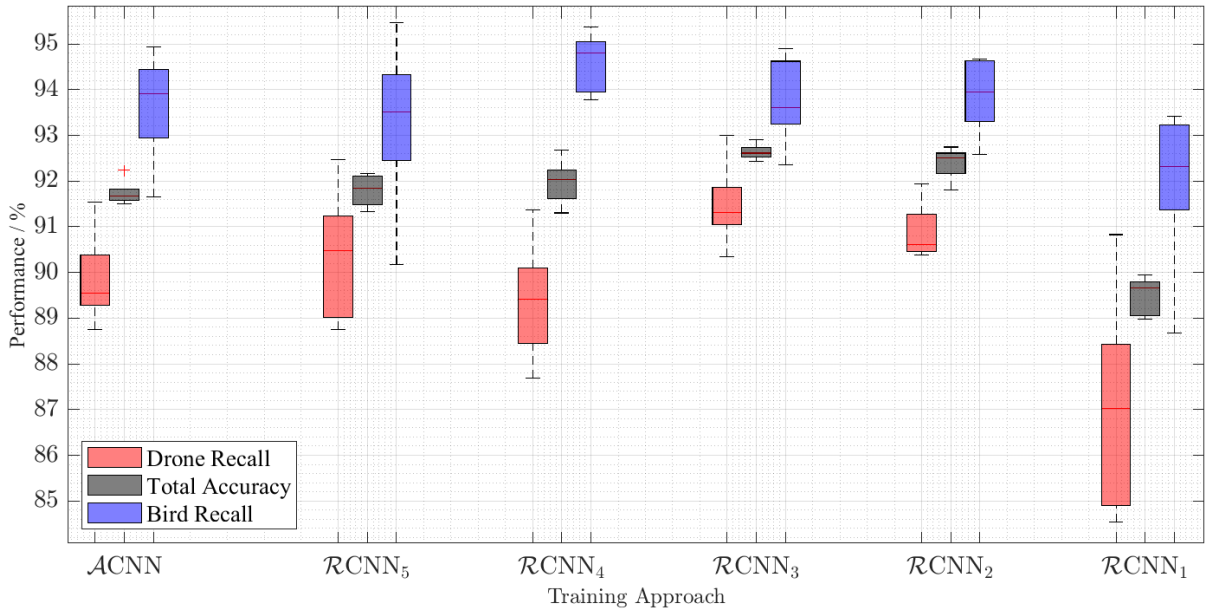
The 2nd epoch of *Alexnet-like* CAE training was used, after which a maximum of 30 epochs of fine-tuning training as the $\mathcal{R}$CNN occurred. The $\mathcal{A}$CNN was trained on the radar data for 45 epochs. For every classifier, test results were again gathered from the epoch that had the best sum of train and validation accuracies, which ensures the model is well trained and minimally overfitted. For the $\mathcal{A}$CNN, a range of optimal learning rates was found for each dataset, which were chosen based on results from a wide learning rate sweep that informed where best performance would be found. The AdaM optimiser and batch size of 32 was used for training all *Alexnet-like* CAEs, $\mathcal{A}$CNNs and $\mathcal{R}$CNNs using 10^{-6} L2-Regularisation with epsilon value set to 10^{-8}. Binary cross entropy loss was used for all training. $\mathcal{A}$CNN was post-trained on radar data, with the convolutional weights frozen. The $\mathcal{R}$CNNs were fine tuned throughout the whole model both to increase the discrimination of existing CAE features that only received two epochs of unsupervised training, and also to account for the fact that no checking of robustness of learning at the CAE stage after only 2 epochs. Fully fine-tuning AlexNet on radar data with unfrozen convolutional layers would carry concerns of overfitting to the limited training set. 27 *AlexNet-like* $\mathcal{R}$CNN classifiers were trained per depth, and in total 16 $\mathcal{A}$CNN classifiers were trained. It was found that a minority of $\mathcal{R}$CNNs saw performance above 51% accuracy, while all $\mathcal{A}$CNN models were trained to a high-standard. This informs that not all CAEs learned adequate features to support downstream classification, and thus we will present an aggregation of the best trained classifiers in Figure 8.4 and Table 8.3.

Table 8.3: Best accuracies of the optically and radar data pre-trained models. Subscript denotes number of convolutional layers.

	$\mathcal{A}$CNN	$\mathcal{R}$CNN$_5$	$\mathcal{R}$CNN$_4$	$\mathcal{R}$CNN$_3$	$\mathcal{R}$CNN$_2$	$\mathcal{R}$CNN$_1$
LDB	**98.58**	97.88	98.46	**98.69**	98.54	96.34
ADB	**92.23**	92.17	**92.68**	**92.90**	**92.74**	89.95



(a) Boxplot comparing results with All Drones and Birds (ADB) dataset.



(b) Boxplot comparing results with Large Drones and Birds (LDB) dataset.

Figure 8.4: $\mathcal{A}$CNN compared with 5 depths of similar $\mathcal{R}$CNN.

The results from Figure 8.4 show that the $\mathcal{R}$CNN pre-trained and fine-tuned just on radar data was able to surpass or be very similar to the accuracy gained from the $\mathcal{A}$CNN benchmark. For the ADB dataset in Fig. 8.4a, 4, 3 and 2 convolutional layer CAEs produced classifiers that exceeded the performance gained from the $\mathcal{A}$CNN classifier based on optical pre-training. The 5 layer $\mathcal{R}$CNN was within 0.1% of the $\mathcal{A}$CNN, showing that even for this large model relative

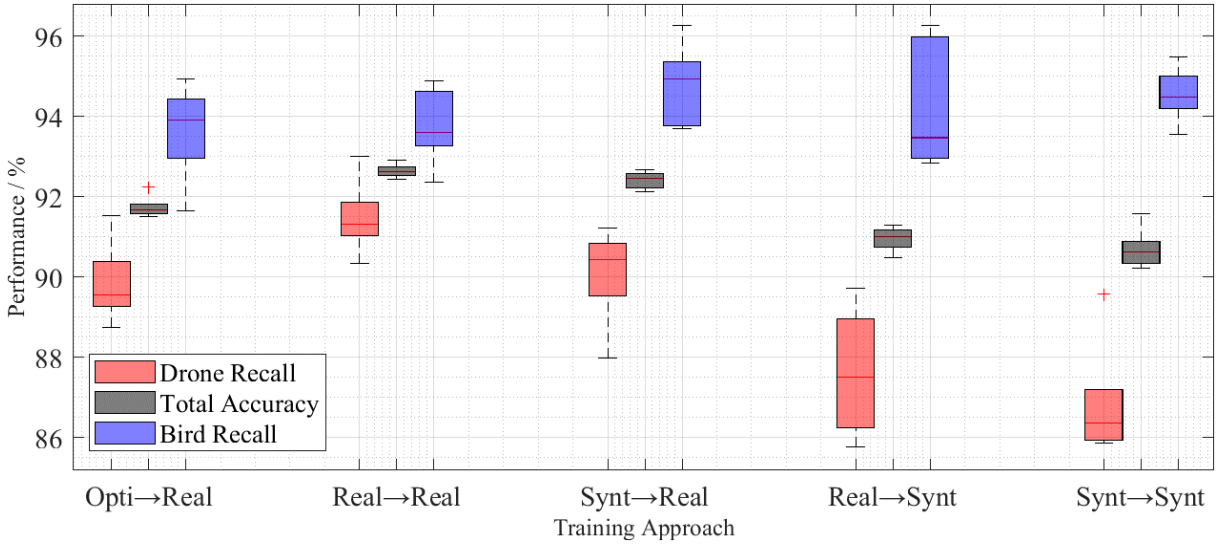
to the size of the dataset, it was able to overcome the less than ideal volume of parameters and could, with a different training regime, possibly surpass it. As was also found in an aformentioned study [98], the 3-layer $\mathcal{R}$CNN here gave rise to the best performing classifiers. The two key differences between these models is that this work used a fully connected layer with the autoencoder pretraining, and here two groups of the same size of convolutions in parallel were used while they concatenated features from two groups of different sizes. For the LDB dataset in Fig. 8.4b, only the 3 layer $\mathcal{R}$CNN surpassed the result achieved with $\mathcal{A}$CNN. Also and similarly to the ADB case, $\mathcal{R}$CNN$_1$ and $\mathcal{R}$CNN$_5$ were the only two convolutional depths not to achieve drone accuracies above 98%. Both $\mathcal{R}$CNN$_3$ and $\mathcal{A}$CNN had all of the 9 best trained models above 98% on the easier LDB dataset that contained a widely varied collection of real urban data.

As per Chapter 7, this assessment is extended by replacing some of the real drone data with equivalent synthetic data. This second experiment's purpose is to understand if the classifier model can be built using simulated data. The synthetic dataset construction was identical to Section 7.1 and [3], where large drone data was simulated and all the small drone data, if used, was kept as real. The same investigation was carried out with the same $\mathcal{A}$CNN and $\mathcal{R}$CNN models. For this assessment the $\mathcal{R}$CNN$_3$ model was used for each dataset, sweeping the other parameters as before described in its corresponding row in Table 8.2. The different training permutations of real and synthetic data are listed with the results in Table 8.4. For comparison, the accuracy from $\mathcal{A}$CNN (the optically pretrained CNN) is also listed. The results for the ADB and LDB datasets are again shown in a boxplot of accuracy and balance via class recall in Figure 8.5. Fine-tuning on synthetic data reduced accuracy by 5% in the all-synthetic-drones LBD set, and by only ~1.5% for the partially synthetic ADB case. The optically pre-trained $\mathcal{A}$CNN benchmark was also surpassed by the $\mathcal{R}$CNN when fine-tuning on real data regardless of whether the pre-training was conducted with synthetic data or not, for both the LDB and ADB cases. This reinforces that real data is imperative when seeking maximum performance. None-the-less, the LDB Synthetic→Synthetic classifier being able to classify targets at >93% shows that acceptable performance can be induced even when no real drone data is present

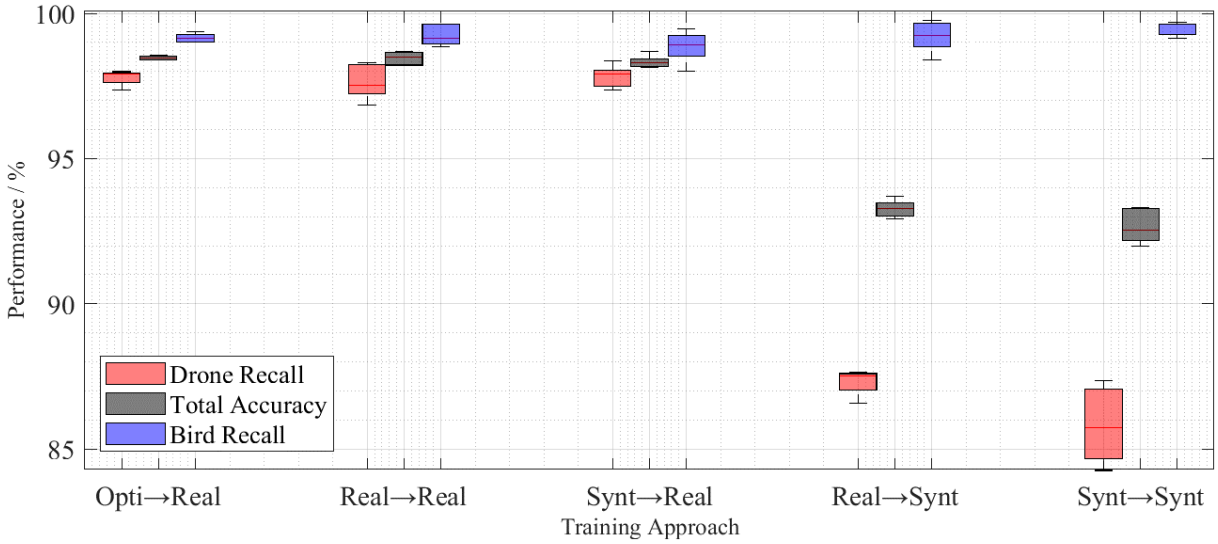
Table 8.4: Best accuracies of the optically, real and synthetically radar data pre-trained models

Model	Training Configuration	ADB Accuracy	LDB Accuracy
$\mathcal{A}$CNN	Sup. Optical → Sup. Real Radar	92.23	98.58
$\mathcal{R}$CNN	Unsup. Real CAE → Sup. Real Radar	**92.90**	**98.69**
$\mathcal{R}$CNN	Unsup. Synthetic CAE → Sup. Real Radar	**92.68**	**98.69**
$\mathcal{R}$CNN	Unsup. Real CAE → Sup. Synthetic Radar	91.28	93.69
$\mathcal{R}$CNN	Unsup. Synthetic CAE → Sup. Synthetic Radar	91.59	93.31

what-so-ever, however the classifier does exhibit stronger bias towards predicting challenging



(a) Boxplot comparing results with All Drones and Birds (ADB) dataset.



(b) Boxplot comparing results with Large Drones and Birds (LDB) dataset.

Figure 8.5: Comparison of optically pre-trained $\mathcal{A}$CNN and 4 real and synthetically Radar (pre-trained) Convolutional Neural Network Classifier ($\mathcal{R}$CNN) classifier results.

edge cases as birds. As discussed in Sec. 7.2.3, further variation of the synthetic data beyond the restrictions to recreate the real original data would better allow the deep learning model to learn more general, universal features of drone targets.

Results as presented here could be improved vastly with larger quantities and variation of

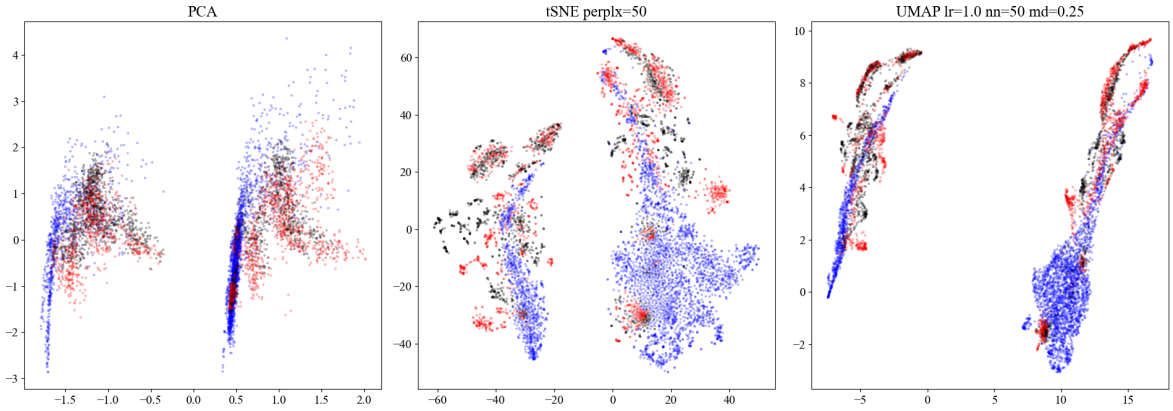
simulated data. This work has shown that it is possible with radar data alone to build and pre-train a CAE and fine-tune it as a supervised CNN that can match optically pre-trained AlexNet CNN models. This method also allows the use of synthetic data in parts of training without too significant a comprimise, supporting and extenting the work for Chapter 7. This is a significant finding that will empower classifiers to possess deep, synthetically engineered features from excess quantities of simulated data, before fine-tuning on a modest amount of real, synthetic or mixed data to direct the learned features towards classification.

8.2.2 Latent Representation of Bird and Drone Data

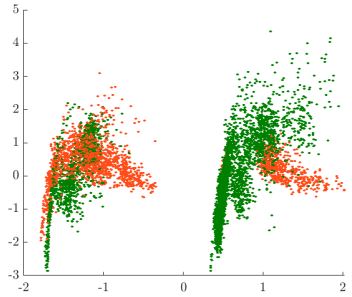
Having established the potential for the CAE to be able to classify drones and birds in the Sec.8.2.1 previously, next an exploration of the latent representation of birds and drone targets is conducted to attempt to understand how well the unsupervised approach has understood the data without labelling. The visualisations of the learned representations can show if the convolutional layers have learned meaningful features. This section will show various views of the representation learned by a simple CAE on real L-band bird and drone radar data. This will be, an assessment of the encoder through class-labelled latent variable scatter plots, the decoder through sample generation by latent sampling, and the end-to-end Autoencoder (AE) by observing the input sample reconstructions.

Shortly the function of the CAE on the LDB dataset will be considered, but first as performed in other published studies, we will show the decompositions of the original dataset with PCA, tSNE and UMAP to gauge both the operation of the techniques and the initial separability of the dataset. These techniques will be used to visualise the latent dimension if it is greater than 2. The decomposition of the training set of the LDB data is shown for these three techniques in Figure 8.6. The operation of these techniques, UMAP and tSNE, are effectively to preserve the distance between points from the original high-dimensional space to the 2D embedding space. Thus, if two points are very close together in the 2D embedded space, then these two points are also (relatively) close together in the original space. Otherwise, the axis values and positioning are

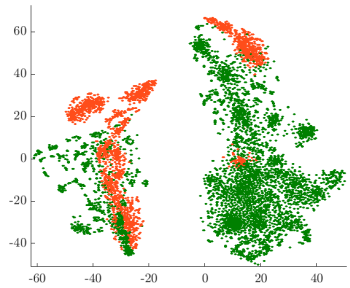
fairly meaningless - it is the spatial relationship between points that these techniques preserve. 800-1500 samples per class were used in each labelled latent scatter plot in the remainder of this work, with an equal number of random samples across the classes.



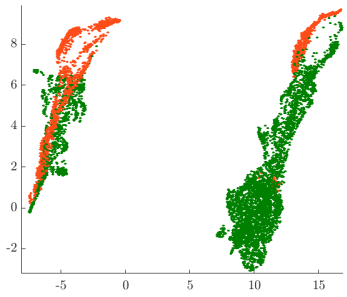
(a) PCA, tSNE and UMaP with LDB Train Dataset



(b) Radar Source of PCA



(c) Radar Source of tSNE



(d) Radar Source of UMaP

Figure 8.6: Various Representations of LDB Training Dataset Decomposition. Birds (Blue), *Inspire* (Red) and *Matrice* (Black) and Radar#1 (Green) and Radar#2 (Orange).

The data is presented in two formats. For the top row (Fig. 8.6a) the data labels are bird (blue), *Inspire* (red) and *Matrice* (black). The bottom row (Fig. 8.6b-Fig. 8.6d) is the same data but now the data labels are the respective radar i.e Radar#1 (green) and Radar#2 (orange). This is useful as the two radars have different power budget and background statistics, and we can see if the learned representations are split between target features, or radar source. The two clusters are not entirely split by the radar source, but many sectors of the representations are exclusively comprised of either radar.

A new, Simple Convolutional Autoencoder (*Simple*CAE) structure was created inspired by the results of Sec. 8.2.1 using 3 convolutional layers. Some further changes were made to

encourage better learned representations, as the $\mathcal{A}$CNN was found not provide consistent latent representations compared to other model configurations tested. The $\mathcal{A}$CNN was created for the express purpose to performing a comparison with the AlexNet configuration, and for this section the creation of an original neural network aimed to showcase representations learned using a basic CAE implementation. The model was tweaked as follows, each choice with the motivation of encouraging compression of features towards the bottleneck, in contrast to AlexNet which increases the number of features up to the final classification layer. The size of the convolutional layers were changed to 17→9→3, that with the size of the input image, now 169×169, makes the largest window 1/10 of the sample input. Only the sized 17 convolutional window possessed a stride which was 4. The max-pooling was after only the first convolutional layer, and had again a window size of 3 and a stride of 2. The number of channels outputted from each successive convolutional layer was reduced closer to the latent space (64→32→16, as opposed to 48→64→96 used previously) to improve the compression of information towards the bottleneck. Only one fully connected layer between the convolutional block and the latent variables was used to promote more learning within the convolutional layers, unlike in AlexNet using two, where many features need to be non-linearly combined to make a single prediction given its great number of classes. The radar study [98] did not use any linear fully connected layers between the convolutional layers and the latent variables. However in this work with greater levels of noise from the urban, outdoor long ranged environment and the use of two radar systems differing in performance, features of noise learned from the data can be introduced into the latent representation superimposed with target features easier with a fully-connected layer. No dropout after fully connected layers was used, as had been in all training previously, as it was found to significantly slow and reduce the repeatability of representations learned.

Figure 8.7 shows this *Simple*CAE architecture used to create the latent representations in this section. For its training, a sweep of learning rate order of magnitude was used without any learning rate variation. Consistently, 10^{-5} gave the latent representation the clearest defined structure as well as the best separation of labels that will be seen throughout this section. The effect of learning rate on unsupervised classification will be considered in Sec. 8.2.3. Using this

basic model, we will explore changing the latent dimensionality used to explore the effect of the compression bottleneck on the overall training. The size of the latent dimension will be denoted in this section as a subscript, for example *Simple*CAE$_2$.

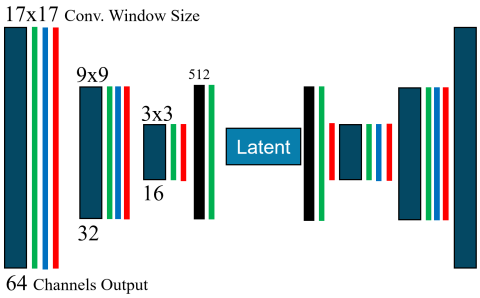
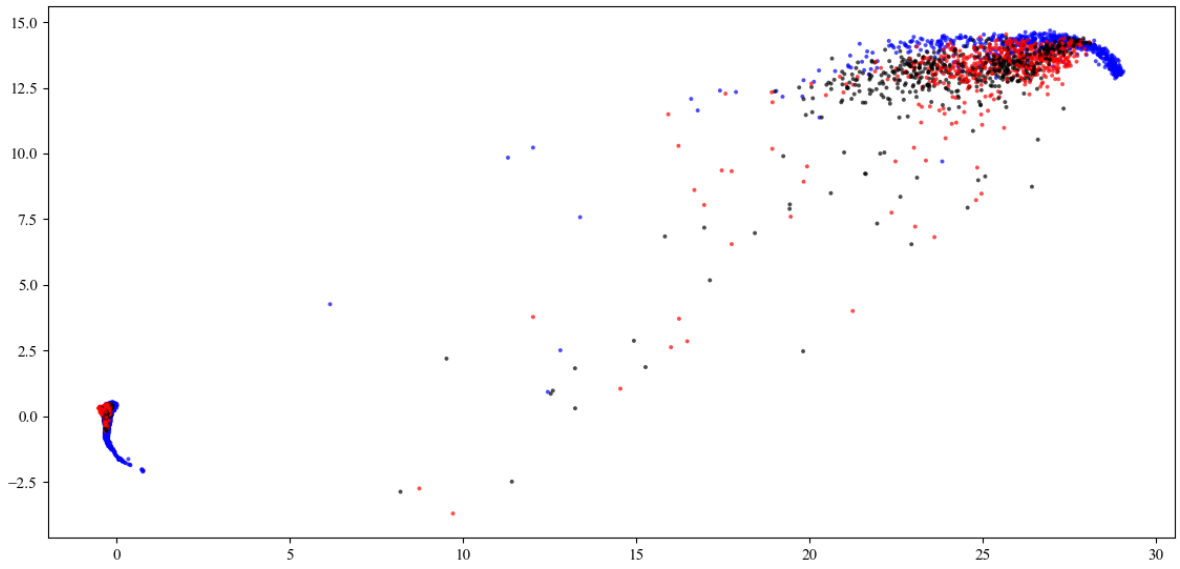
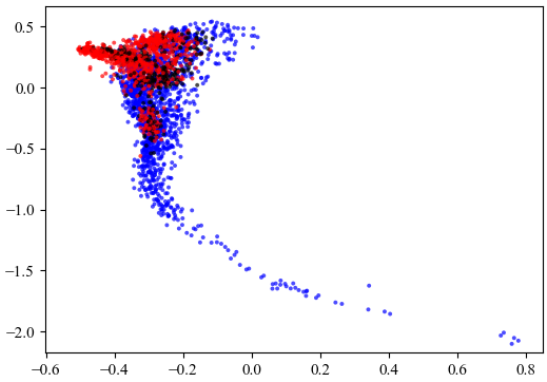


Figure 8.7: *Simple*CAE architecture used to create the representations shown in this section.

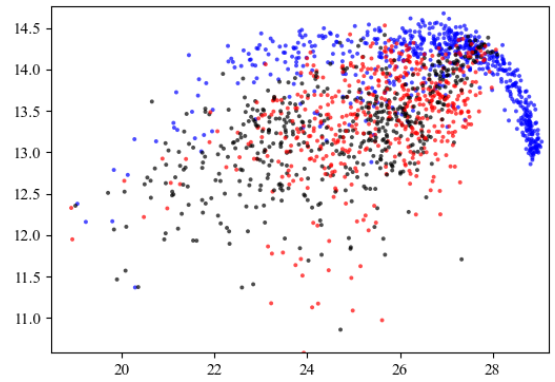
We begin by observing the latent representation in Figure 8.8 for the *Simple*CAE$_2$ for only 2 latent variables. Figure 8.8a shows the latent scatter plot of the LDB test set of the *Simple*CAE trained for 50 epochs. The plots for the entire dataset are shown labelled according to target type. It can be seen that two clusters of data are formed again, and zoomed in plots are shown for each of the clusters in Figs. 8.8b and 8.8c. It can be seen that the right cluster has very good separation of labels, with a very small section of overlap between the large drones and birds. The left cluster also has a fair separation of labels although the spacing between variables is much tighter. In Figs. 8.8d and 8.8e the radar source is shown relative to the points. In Fig. 8.8d, it can be seen that the radar source is not as impactful in the CAE latent representation, with only the top left corner of the left cluster mapped to Radar#2 drones. In the right cluster, the radar source is fairly spread out across the distribution, where there is not a region that is exclusively mapped to a unique radar. With comparison to the full dataset decomposition earlier in Figure 8.6, the radar of origin is significantly less relevant to the natural unsupervised clustering of the data. This shows that the *Simple*CAE$_2$ has comparatively learned more about the target data, and become less concerned about the radar specific features that occurred when measuring the target.



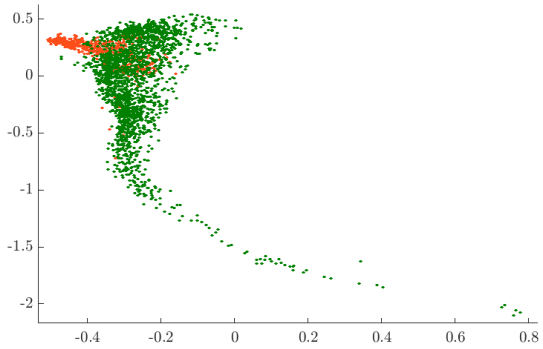
(a) All samples with class labelling



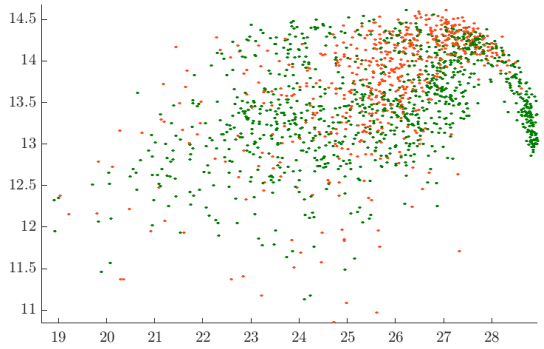
(b) Zoom into left cluster with class labelling



(c) Zoom into right cluster with class labelling



(d) Zoom into left cluster with radar labelling

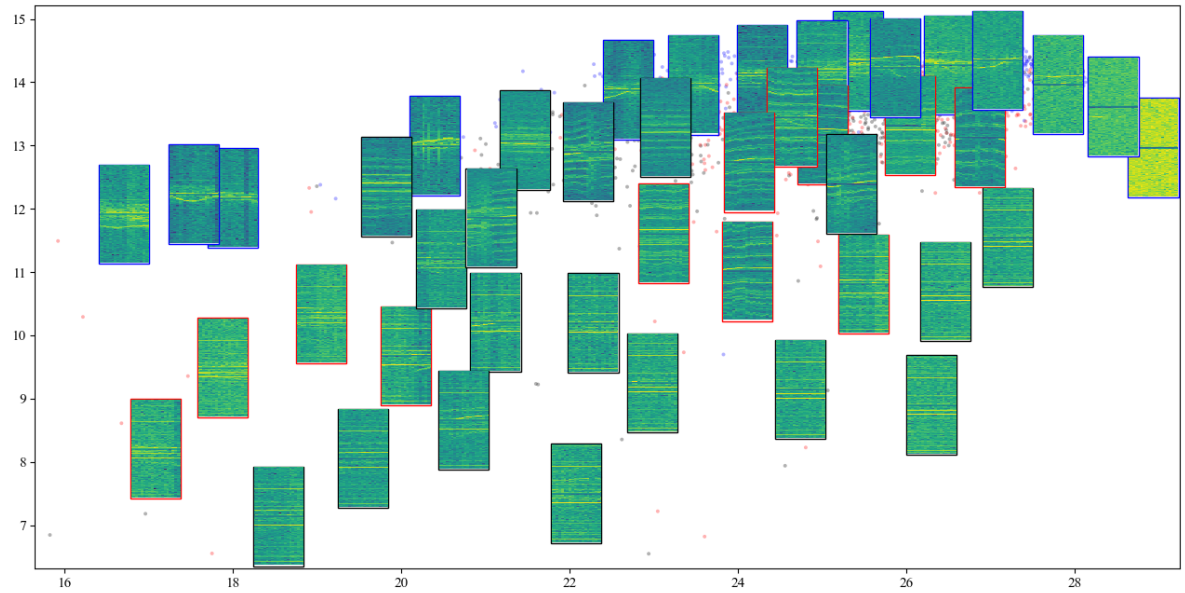


(e) Zoom into right cluster with radar labelling

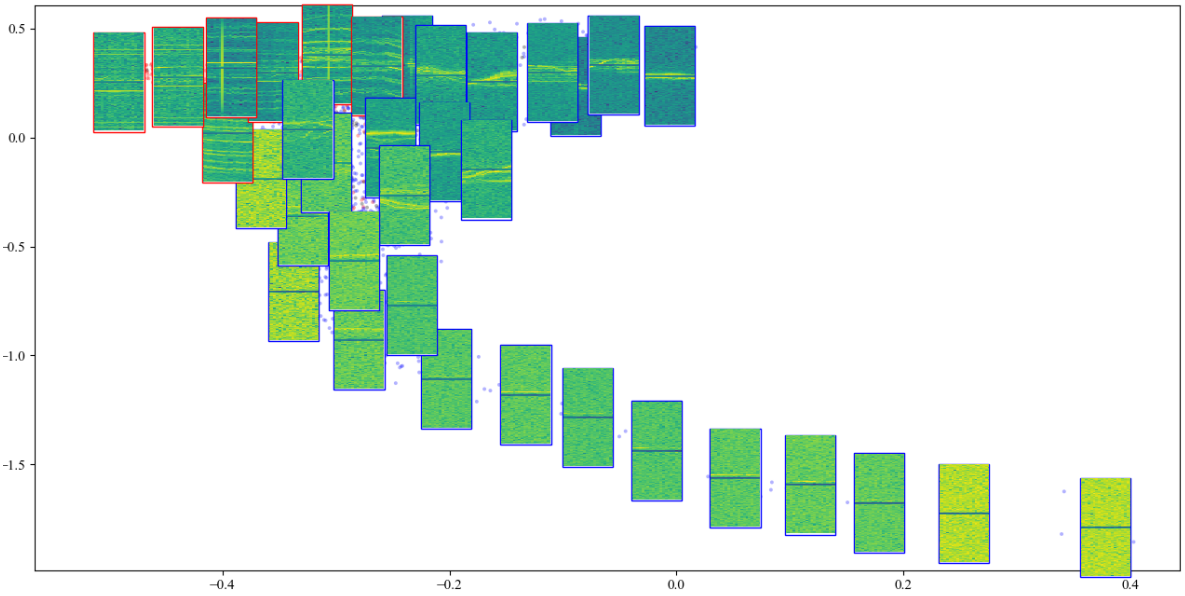
Figure 8.8: CAE on LDB test set with 2 latent variables. *Matrice* is Black, *Inspire* is Red and Birds are Blue, while Radar#1 is green and Radar#2 is orange.

Figure 8.9 shows the same labelled latent representation as Figure 8.8 but with the samples that were processed by the decoder annotated on to the position of its latent coordinate, where Fig. 8.9a shows the right cluster and Fig. 8.9b shows the left. The color of the bounding box of each spectrogram image is the same as its class of origin. Targets can be visibly assessed in these samples for a sense of how the latent space handles different spectrogram features. The left cluster is more mixed, with the top left region dedicated to Radar#2 drone signatures, while the top right shows strong bird flocks, and the bottom right extension containing birds or no target. The right region in Figs. 8.8c and 8.9a have a more defined boundary of separation of labels than the structure of the data that lead to this representation. Again, single birds and targets with no body Doppler that were obscured by clutter are seen on the far right extension. Moving left from this point enters a confusion zone where the labels are strongly overlapping. Beyond this point the distribution splits into two where birds are high on the y-axis, and drones exist at lower values on the y-axis. Generally as the y-axis value decreases, the samples go from birds, to targets with strong μDoppler, to drones with feint μDoppler and strong clutter lines. Overall, the annotated samples show the latent space has an interpretable gradation of features that matches the variation of real data. Quantitatively assessing this gradation of physical spectrogram features in a latent representation is of interest as future work.

Up till now only the learned representation of the encoder has been shown, from which the decoder creates the reconstruction that ultimately decides the neural network loss. This is loosely tied to the encoder mapping without a variational component or other extension to the approach. We will observe how the bird and drone input samples were reproduced at the decoder output, $D(E(z))$. Figure 8.10 shows the reconstructions of 4 train and test samples for the drone and bird classes from the LDB dataset. In the left column, the original spectrogram sample is on the left, while its reconstruction after a full forward-pass of the *Simple*CAE$_2$ is shown on the right. Additionally, the central spectra in the middle of these spectrograms shown are displayed in the right column, with the original spectra in black and the reconstructed spectra in red.



(a) Sample annotated latent distribution of right cluster.



(b) Sample annotated latent distribution of left cluster.

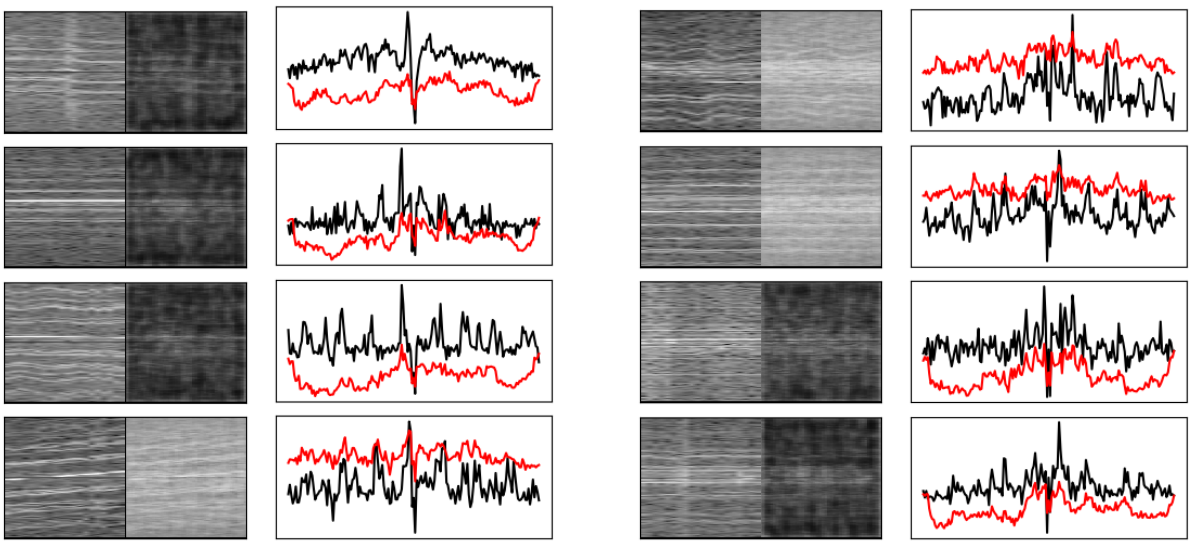
Figure 8.9: CAE on LDB test set with 2 latent variables.

For each of the train and test examples, it can seen that the signature has been somewhat replicated in the reconstructions. The background level of the reconstruction was often not well replicated, but in spite of this, target features were occasionally reasonably captured in the reconstructed spectra and spectrograms. The depth of background color, corresponding to the SNR of target signal, of the reconstruction is in fact unrelated to the input as can be observed across examples within Figure 8.10. Some details of the target returns have been learned in the test and training sets, while the background level was not frequently robustly copied and learned. For several drone examples in Fig. 8.10, the positioning of many HERM lines were nicely captured in the reconstructions, however the variation of signal level not mapped well in the latent representation. This provides some confidence that the learned representation has captured important target features while largely ignoring the global radar features, while only using two latent variables. Later, analysing the reconstructions with a greater latent dimension size that would in theory allow more space for the latent representation to understand the targets, will be presented.

Figure 8.11 shows the decoder sampled along each latent axis as shown in Fig. 8.11a where samples are generated along the (two) perpendicular axes crossing through the latent origin. In this case with 2 latent variables, the '0' and '1' indicates the first and second latent axis, which corresponds to the x and y axis highlighted in Fig. 8.11a[2]. These plots show the robustness of the decoder by testing if realistic samples are produced in all regions of latent space. Fig. 8.11b shows the decoder constructions out to $z_d = \pm 30$. From the latent representations shown previously, the top right cluster existed at about (28, 14) in latent coordinates. It is not clear that any of the samples produced are realistic or even plausible as target samples. In comparison, Fig. 8.11c shows that the decoder has learned a sense of the stationary clutter and effect of the low pass filter, that differed across samples based on the clutter properties at the location and height of flight. Fig. 8.11d looking near to the origin possibly has the most realistic target samples, showing returns similar to bird flocks in the negative direction and a semblance of drone μDoppler in the positive direction. This result with some promise, motivates efforts to induce more robust

[2]See Appendix A.2 for the same procedure applied to *Simple*CAE with up to 32 latent variables.

learning to create more continuous latent representations, where a sampling procedure like described here may showcase a more interpretable gradient of radar return features.

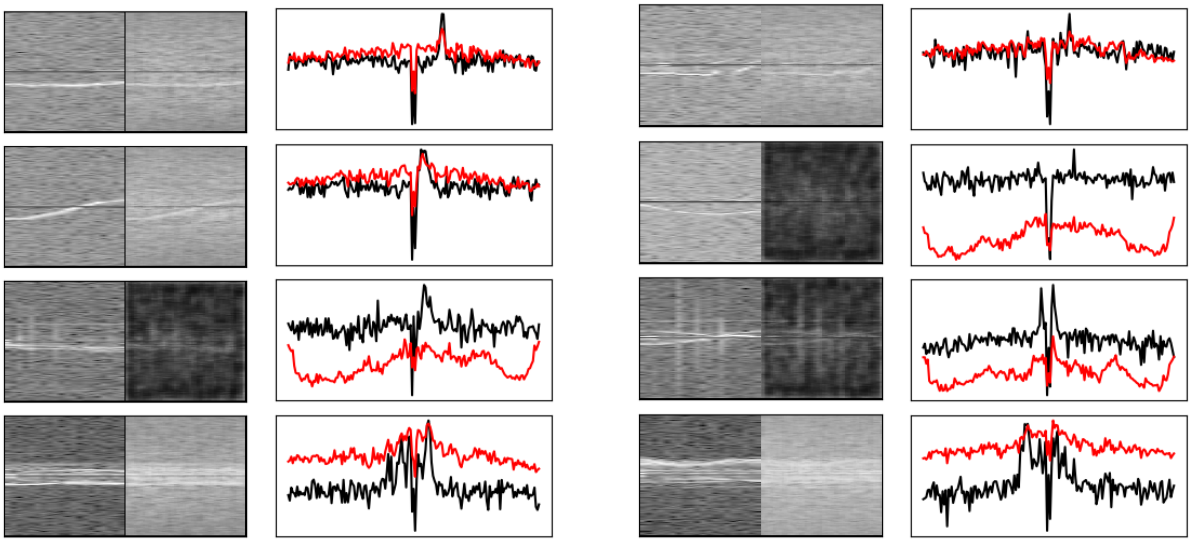


(a) Train Drone Spectrogram

(b) Centre Spectrum of (a)

(c) Test Drone Spectrogram

(d) Centre Spectrum of (c)



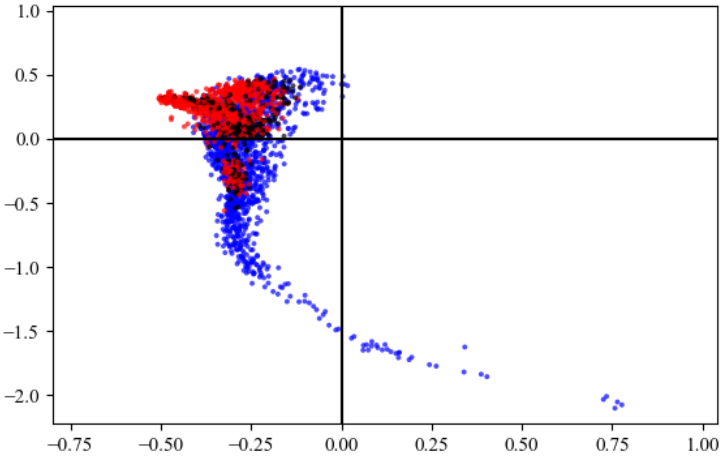
(e) Train Bird Spectrogram

(f) Centre Spectrum of (e)

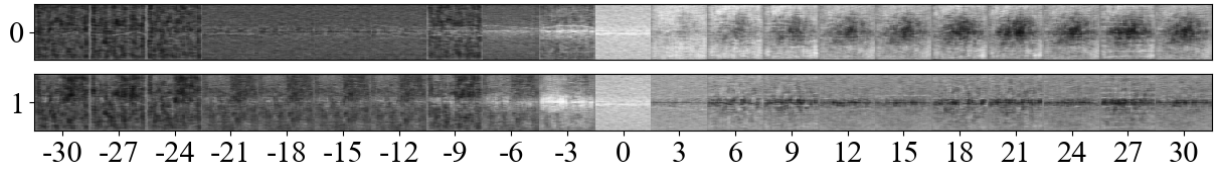
(g) Bird Test Spectrogram

(h) Centre Spectrum of (g)

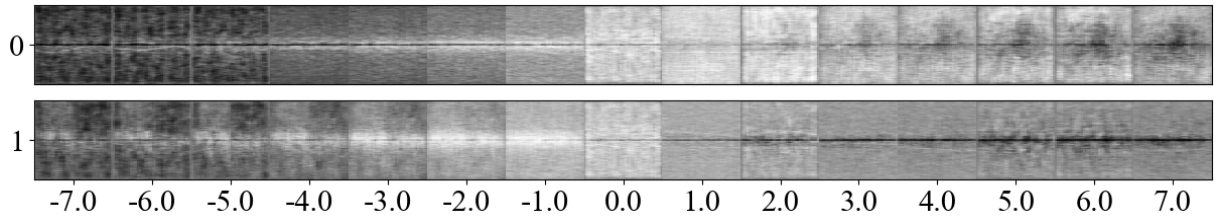
Figure 8.10: Reconstructions of spectogram and central spectra with LDB dataset. Originals are left and black, with *Simple*CAE$_2$ reconstructions on right and in red.



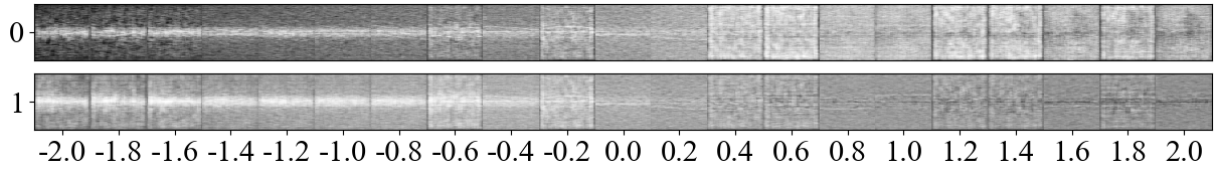
(a) *Simple*CAE$_2$ with latent axes highlighted.



(b) Between ± 30 in steps of 3



(c) Between ± 7 in steps of 1



(d) Between ± 2 in steps of 0.2

Figure 8.11: LDB trained decoder uniformly sampled along each latent dimensional axis.

It could have been already reasoned from observing the latent representation that the decoder would not produce realistic samples, in Figure 8.8, which shows that the two clusters of data occurred near the origin and in an extended zone at (28,14). This means that at, for example, (-28,-14) we would not expect any realistic samples to be produced. An ideal learned representation could produce original, realistic samples up to $\pm\infty$, but in practice this is not likely to be realised. Then, also considering that the reconstructed samples were weakly matching the original input,

showing that although learning had taken place, it was not sufficiently robust to also predict other features of the samples, such as background power. This shows that the latent representation is serviceable as it was able to cluster our samples fairly effectively, but could be improved to overcome these demonstrated limitations. Extensions to the basic CAE approach such as the *Simple*CAE used here are known to improve random sampling of the decoder by ensuring the decoder is regularised to learn a complimentary mapping. That said, the fact that unseen test samples were well reconstructed to the same level of fidelity to the training samples shows that the exists a reasonable representation of the finite real dataset, but it has not yet learned a fully robust representation of all possible bird and drone manifestations.

These downsides of the results presented could also be postulated to be a symptom of attempting to do too much compression of our dataset into just two latent variables, *Simple*CAE$_2$. Results with different latent variables presented in the same format as above for *Simple*CAE$_2$ are shown in Appendix A.2. On these latent variables, classification can be conducted directly using an algorithm such as kNN, k-Means or Guassian Mixture Models [291]. We will use a kNN here to assess the self-separation of labels of the LDB test dataset. A simple k-Nearest Nearest Neighbours Self-Similarity (kSS) algorithm was defined to give a basic assessment of how well separated the labels were amongst the latent variables:

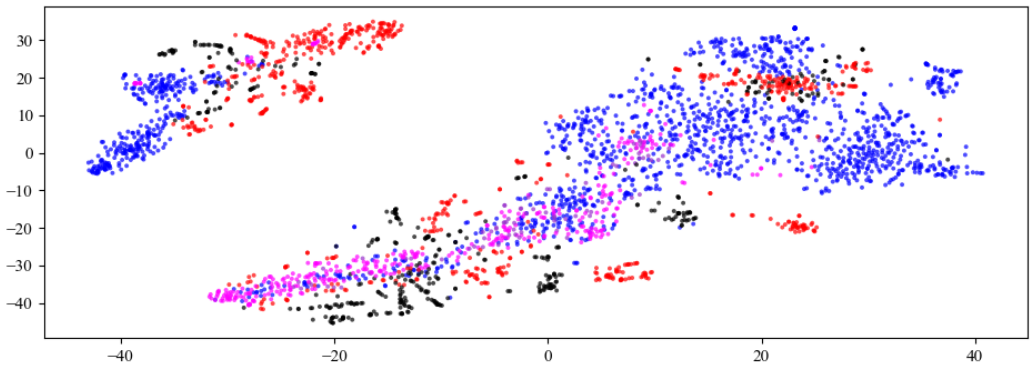
1. Using an input of two sets (e.g. Train and Test, or both could be the same set),
2. Calculate the kNN classification result on the latent variables for each in the first (test) set against the second (train) set.
3. Repeat for k = 11, 31, 51, 71 and 91,

where in this case of using the same set for the kNN test and training data, this performs an evaluation of the separation of labels amongst the latent variables of that set. Later in Sec. 8.2.3, the two input sets will be the train and test sets which will check how similar the training and test sets' latent distributions are, which along with loss or accuracy is possibly a method of determining when the model is optimally trained.

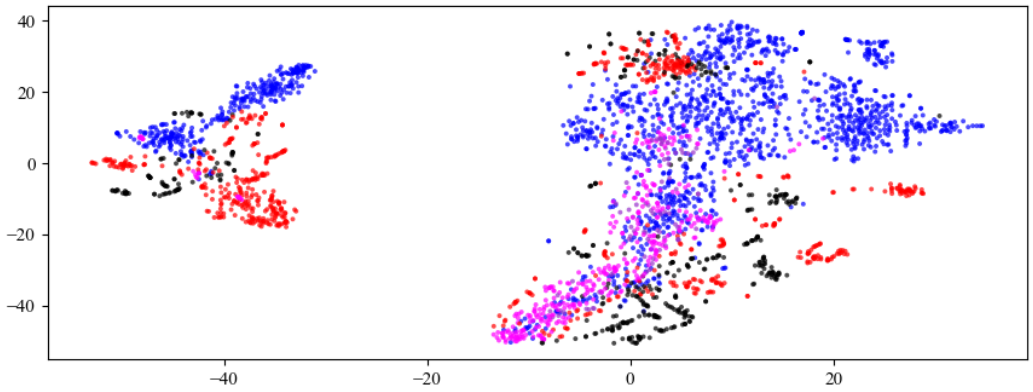
Table 8.5: Table of results with LDB dataset comparing unsupervised kSS results

kNN input	Accuracy
*Simple*CAE$_2$	80.9
*Simple*CAE$_4$	87.2
*Simple*CAE$_8$	89.6
*Simple*CAE$_{16}$	88.6
*Simple*CAE$_{32}$	88.6
28561D Full Dataset	88.4
Full Dataset + tSNE to 2D	93.5

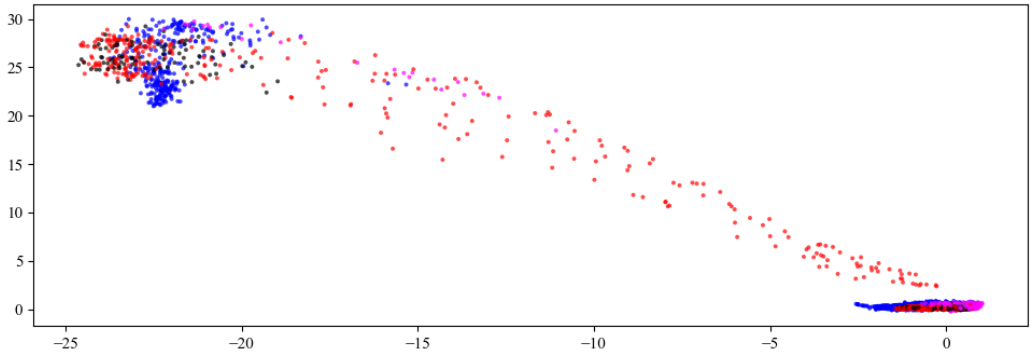
The kSS results using only the test set for the CAE models are shown in Table 8.5, along with the results of the tSNE dataset decomposition and the dataset itself. The kNN of the CAE results were applied directly to the latent distribution and not the 2D visualisation. It appears that the separation of classes increases by 7% going from 2 to 16 latent dimensions. The overall structure of the latent space was similar between the tSNE and CAE+tSNE with more than 2 latent dimensions. For the CAE with 2 latent dimensions, the structure of the representation was much simpler, having experienced much more compression which resulted in the lowest kNN separation metric. The full dataset decomposed with tSNE and kNN applied to the 2D output was 93.5% while the kNN on the full 169×169=28561 dimensional data input was 88.4%. The deficit of the CAE separation only by scrutinising its latent variables, is not of great concern for classification, as it was shown that supervised fine-tuning on the similar *Alexnet-like* $\mathcal{R}$CNN$_3$ from Sec. 8.2.1 was able to achieve over 98% on this same dataset. The *Simple*CAE from this section has been shown to produce latent representations that possess interpretable features with meaningful separation of classes, while also showing a degree of invarience to artefactual radar features. The following and final Sec. 8.2.3 aims to investigate whether separation of classes amongst the latent variables is conducive to achieving better classification after supervised fine-tuning for drone-vs-bird classification.



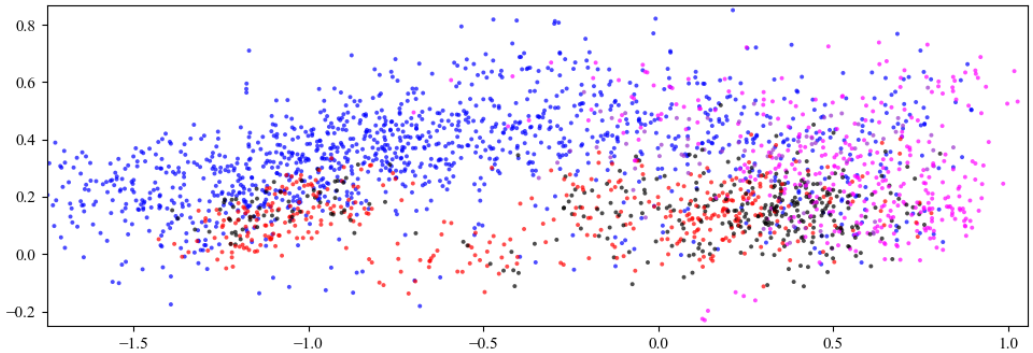
(a) Dataset+UMAP



(b) CAE+UMAP with 32 Latent Variables.



(c) CAE with 2 Latent variables.



(d) Zoom into right cluster in (c).

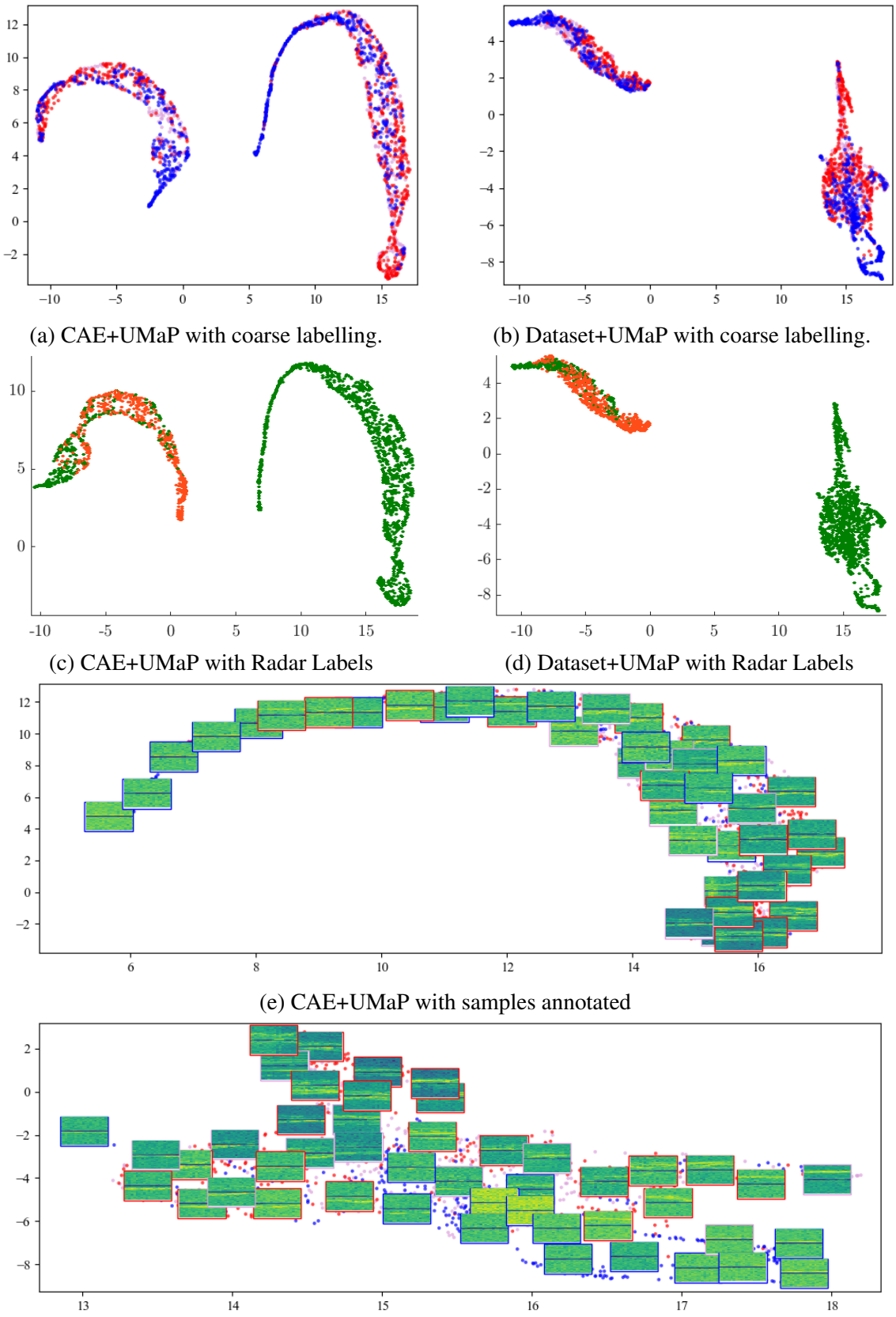
Figure 8.12: ADB labelled latent scatter plots of test set at epoch 30. *Mini* (pink), *Inspire* (red) *Matrice* (black) and birds (blue).

So far in this section, the latent space separability, sample reconstructions and decoder sampling have all been assessed using the real data LDB dataset. Now a selection of the previous analysis is repeated on the All Drones and Birds (ADB) dataset in order to consider the addition of μDoppler-less drone targets in the labelled latent scatter plots. The addition of these targets reduced supervised classification accuracies by about 8% for the AlexNet (pre-trained) Convolutional Neural Network Classifier ($\mathcal{A}$CNN) (Sec. 5.2.5), and by 6% for the *AlexNet-like* Radar (pre-trained) Convolutional Neural Network Classifier ($\mathcal{R}$CNN) (Sec. 8.2.1). Figure 8.12 shows the latent variable representations for the following cases:

- Fig. 8.12a of the full ADB dataset decomposed with UMAP
- Fig. 8.12b using *Simple*CAE$_{32}$ and UMAP
- Fig. 8.12c using *Simple*CAE$_2$
- Fig. 8.12d using *Simple*CAE$_2$ zoomed into main cluster

In each, the *Mini* samples (pink) are fairly well clustered and unsurprisingly found scattered amongst bird data. It is additionally observed in each case there are dedicated regions of bird data where pink *Mini* data exists extending out of the main cluster, that are mixed with bird data that is also adjacent to large drones with μDoppler. The tSNE embedding used in Figs. 8.12a and 8.12b makes the clustering very easy to visualise. The same is true for Figs. 8.12c and 8.12d, but the embedding without a visualisation aid might make it harder to see at first glance. The essential similarity of the three outputs shows that all *Simple*CAEs, even the *Simple*CAE$_2$, was able to perform dimensionality reduction to contain understanding of the targets' features that were learned without labelled data.

Thus far, only datasets from featuring drones and birds have been analysed. It has been shown that the variety of bird signatures in the collected dataset is large, so some insight can be gained by looking at the a bird dataset in isolation. Efforts to better understand these returns can be achieved with autoencoders by clustering of similar targets without laborious labelling. Figure 8.13 shows the latent variables that were coarsely labelled by visual inspection on a per flight basis, as a 'flock' (red), a 'single' (blue), or 'unspecified' (pink) that meant it was not a clear example of either. Figs. 8.13a, 8.13c and 8.13e show results for the test set applied to *Simple*CAE$_8$ at the 10th epoch of unsupervised training visualised with UMAP, along with Figs. 8.13a, 8.13c and 8.13e that show the result for the entire test set decomposed with the same UMAP parameters. Again, the overall separation of labels are similar in Figs 8.13a and 8.13b, while in Figs 8.13c and 8.13d labelled by radar of source shows that their global representation is similar to each other, with one cluster containing exclusively Radar#1 targets with the other featuring a mix. For both the Radar#1 clusters, the CAE and UMAP visualisations are annotated with some spectrogram samples in Figs. 8.13e and 8.13f, to illustrate the mapping of features achieved with both the UMAP decomposition with and without the *Simple*CAE. Both approaches produce a meaningful mapping of samples that is possibly much more accurate than the colour labels assigned at the track level by the human labelling procedure. The bottom right of the *Simple*CAE$_8$+UMAP results (Fig. 8.13e) and the top of the Dataset+UMAP result (Fig. 8.13f) both clearly display the success of the unsupervised approach by showing a region where extracted features conducive to flocking activity were detected and thus assigned.



(a) CAE+UMaP with coarse labelling.

(b) Dataset+UMaP with coarse labelling.

(c) CAE+UMaP with Radar Labels

(d) Dataset+UMaP with Radar Labels

(e) CAE+UMaP with samples annotated

(f) Dataset+UMaP

Figure 8.13: Comparison of *Simple*CAE$_8$+UMaP and UMaP with opportune bird test set at the 70th epoch of training. 'Flock' (red), 'Single' (blue) and 'Unspecified' (pink) labels as described in Section 4.3.4.

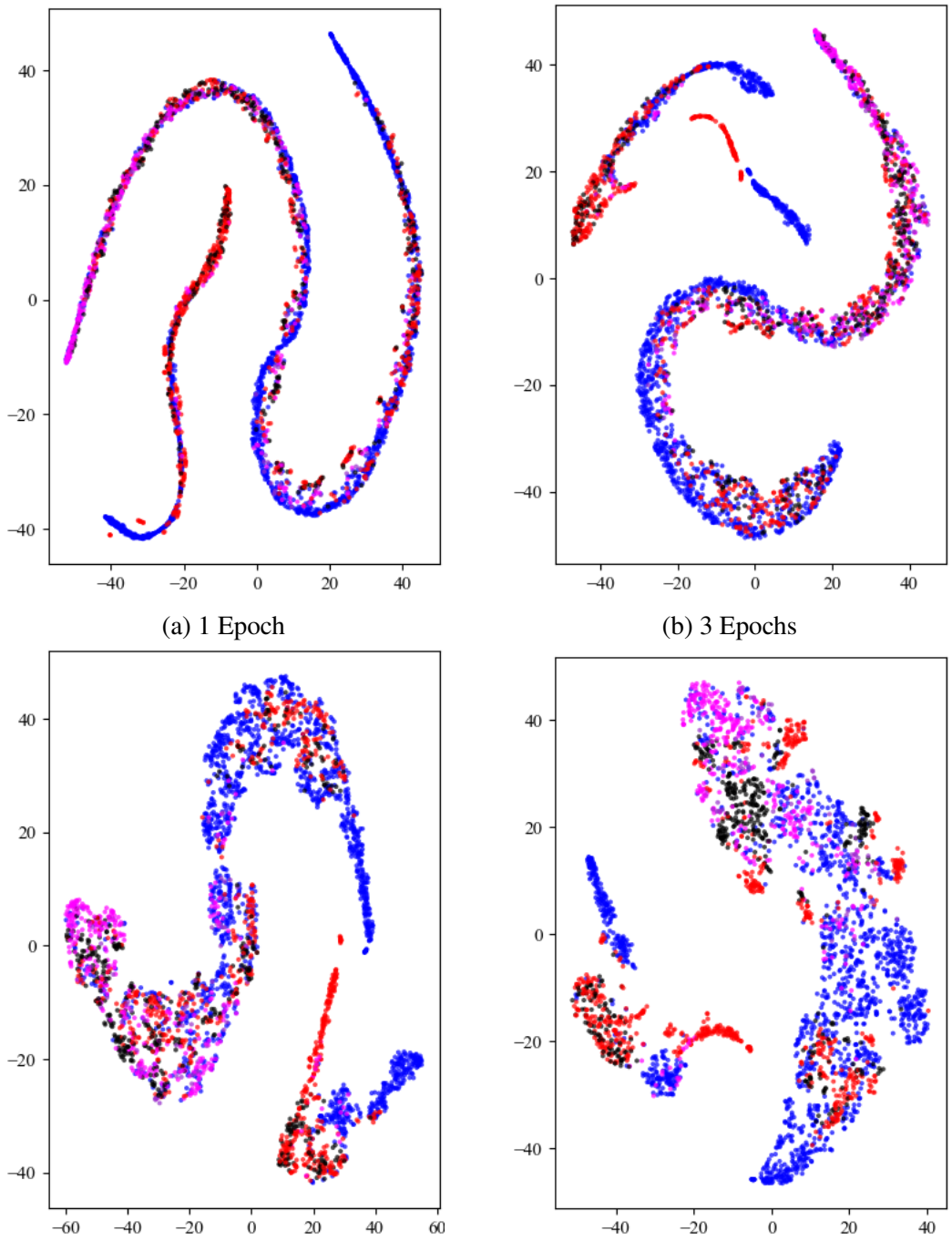
8.2.3 Latent Representation vs Classification Performance

So far we have used the learning rate 10^{-5} in all training of the 17-9-3 *Simple*CAE for the latent representations, and shown that features are captured by the CAE to successfully cluster similar data together in the encoder, and at times produce reasonable reconstructions with the decoder. Extensions to the autoencoder may improve these latent representation in a number of ways as discussed. Sec. 8.2.1 showed that a basic CAE model pre-trained on a dataset of ~5000 real radar samples with supervised fine-tuning for classification (*AlexNet-like* $\mathcal{R}$CNN) was shown to outperform the optical pre-trained AlexNet CNN ($\mathcal{A}$CNN). Additionally, it was shown that there are configurations of a basic CAE model that can learn somewhat interpretable latent representations of the dataset. In this final section we address the question how interpretabilty of a latent space is related to the classification performance observed after the final supervised fine-tuning step to create a $\mathcal{R}$CNN.

At first, the selection of epoch for assessing each of the trained autoencoders had to be decided. Each were trained for a fixed value of 31 epochs. As before the kSS was used to gather classification results directly from the latent variables used previously to populate Table 8.5. The kSS was used on the datasets three times, with the unseen test set evaluated against itself, the train set evaluated against itself, and the unseen test set evaluated against the training set - which factors in if the learned representations are being applied similarly to previous seen and unseen samples. The mean of these three results was determined to be the best trained model, and was used as the value of assessment of unsupervised performance. The epoch of this result, was then used as the starting point for supervised fine-tuning of the $\mathcal{R}$CNN with the training set, tested on the test set. .

The same 17-9-3 *Simple*CAE model was trained, sweeping the learning rate in powers of 10 from 10^{-8} to 10^{-1} and the size of latent dimension in powers of 2 from 2 to 32. These were then fine-tuned in a supervised manner identically as described in Section 8.2.1, here using the All Drones and Birds (ADB) dataset. Initially, how the *Simple*CAE$_8$ that has 8 latent dimensions performs as a function of epoch is illustrated via tSNE at 1, 3, 9 and 27 epochs (see Figure 8.14), while

the corresponding reconstructions and decoder latent sampling results are shown in Appendix A.3, in Figure A.6 and A.7. These figures show that as the model learns with increasing epochs, so a finer structure is possessed within the clusters of the latent space visualisation. As it is a possibility that autoencoder features become over-tuned to the training set, the kSS evaluation used here also considers similarity of mappings across and between the training and test sets. The test and training sets were used equally in the unsupervised performance evaluation, while only the test set was used in the supervised evaluation, with a separate validation set used as standard to combat overfitting retaining blindness to the test set.



(a) 1 Epoch

(b) 3 Epochs

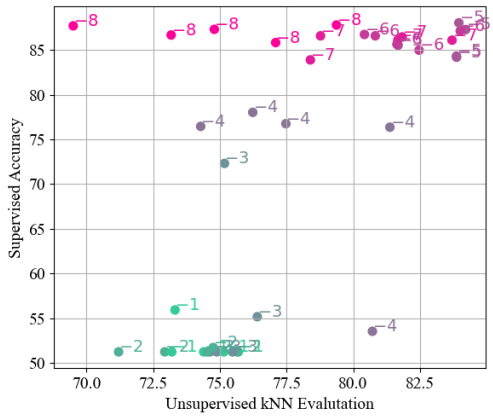
(c) 9 Epochs

(d) 27 Epochs

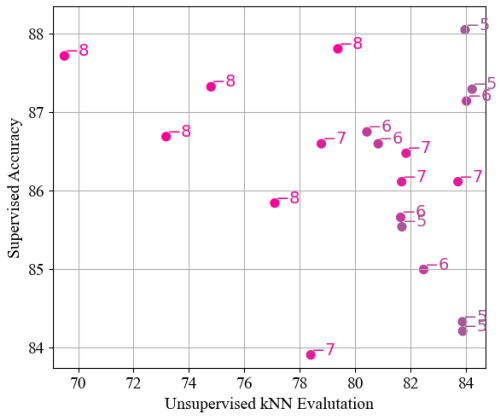
Figure 8.14: Labelled Latent Variables for ADB after a number of epochs of training.*Mini* (pink), *Inspire* (red) *Matrice* (black) and birds (blue).

Next, the assessment was repeated for a number of CAE configurations. A total of forty versions of the *Simple*CAE were trained in a 5-by-8 grid of 2, 4, 8, 16 and 32 latent dimensions and learning rates of $10^{-1,-2,-3,\ldots,-8}$, and the epoch that scored the highest in the mean of kNN evaluation

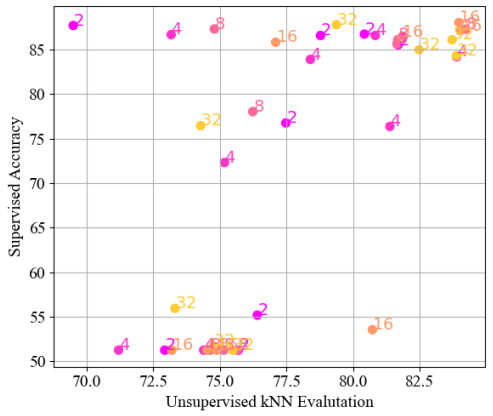
was then fine-tuned with the same training set in a supervised fashion into a $\mathcal{R}$CNN identically as described in Sec. 8.2.1. Figure 8.15 shows the best mean unsupervised kSS evaluation on the x-axis with the, classification accuracy from the subsequent supervised fine-tuned $\mathcal{R}$CNN along the y-axis. It is quite straight-forward from this plot to scrutinise if better latent representation allows a good choice of model for the downstream task of classification. The figure shows the results annotated with the CAE training configuration used, with Figs. 8.15a and 8.15b highlighting the learning rate and Figs 8.15c and 8.15d showing the latent dimensionality.



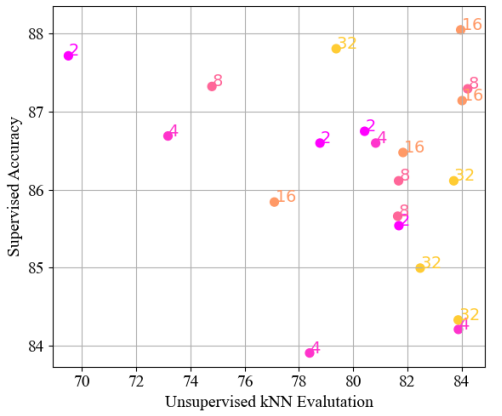
(a) Annotated with Learning Rate ($\times 10^x$)



(b) zoom of (a) to best performing



(c) Annotated with Latent Dimension Size



(d) zoom of (c) to best performing

Figure 8.15: Mean kSS Evaluation of CAE pre-training plotted against supervised fine-tuning classification.

The comparison of the supervised and unsupervised metrics shows two main clusters of performance. The first is the top right hand part of this graph. This is showing that high accuracy (i.e

>85%) with supervised classification correlated with good latent representation that sat around 80-84%. The other cluster to note are where latent representations near 75% unsupervised kNN score, produced very poor supervised classification results just above 50%. So these plots show that better latent separation can in the main lead to better classification accuracy. However, there is one result in the top left where a unsupervised score of <70% still resulted in very good classification accuracy.

When observing the pattern of the colours and annotations, the origins of the general trend with respect to the CAE training parameters is more clear. It can be seen comparing Figs. 8.15a and 8.15c that the choice of learning rate was much more significant than the choice of latent dimension. Beyond 10^{-5}, the supervised accuracy falls to 50% random guessing with the kNN result fixed around 75%. Slightly below the 10^{-5} sweet spot, kNN evaluation scores are above 80% which are the best of the experiment. All of these performed well after supervised fine-tuning, but they did not consistently perform the absolute best after supervised learning. The learning rate of 10^{-5} trained the CAE with 16 latent dimensions however did perform the best in both the supervised, and amongst the best in the unsupervised metrics.

Interestingly and constrastingly, the lowest learning rate 10^{-8} held the 2^{nd}-4^{th} best position for supervised accuracy as an $\mathcal{R}$CNN, despite performing the absolute *worst* in the the kNN evaluation of the unsupervised CAE. In fact, the lowest learning rate of 10^{-8} on average had the best supervised classification performance. This shows that the CAE is an excellent initialisation method for a supervised CNN, despite its latent representation not showing as strong label separability as compared to the other choices. As the learning rate increased to 10^{-5}, the kNN evaluation is increased while the supervised results perform slightly poorer and more inconsistently, with the sole exception of the best overall supervised accuracy at 10^{-5}.

In summary, using very low learning rates for CAE training without consideration of the latent representation was shown to consistently achieve near-saturated classification results with supervised fine-tuning. However, the overall best supervised performance came from a CAE that had a good latent representation achieved by using a higher learning rate. Thus, training a CAE

optimising for the quality of latent representation can produce better results with more effort. For practical applications, it may be advantageous to perform a low learning rate training of a CAE on great quantities of varied synthetic data, before fine-tuning on a more modest amount of real data that was available to the operator, or alternatively, to maximise the quality of the latent representation before robustly optimising the fine-tuning process. Both approaches have been shown viable over the course of this Section 8.2, and demonstrations in future work will be useful for better understanding the balance between performance, robustness and interpretability of the risks and merits to the approaches described in this chapter.

8.3 Chapter Summary

This chapter explored the CAE for classification and latent analysis of the bird-drone L-band radar data. The aspects of its operation that were explored are summarised in the following list.

- Section 8.1 at first, explained the basic operation of a CAE, and reviewed four common extensions to modify the unsupervised loss function and their use in existing radar literature.

- Sec. 8.2.1 compared results from the optically pre-trained, transfer learned AlexNet ($\mathcal{A}$CNN), with a CNN classifier pre-trained as an *Alexnet-like* CAE on real radar data before supervised fine-tuning on the same radar dataset ($\mathcal{R}$CNN). Only two epochs of unsupervised training as the CAE occurred before 30 epochs of supervised fine-tuning. The radar data only, CAE pre-training approach was able to outperform the optical transfer learned comparison, for both the LDB and ADB datasets, albeit by less than 1% (see Table 8.3).

- Afterwards, Sec. 8.2.1 again repeated the classification experiment using the 3 convolutional layer CAE that consistently performed the best, but using synthetic data for pre-training and fine-tuning along with real data. The optical benchmark was again surpassed by the CAE approach, regardless of whether pre-training was conducted with synthetic

or real data (see Table 8.4). Compared to the $\mathcal{A}$CNN, pre-training and fine-tuning with synthetic data was within 0.6% of the benchmark for the ADB dataset, and 5.4% for the LDB case. These results, published in [4], demonstrates near state-of-the-art performance using a modest amount of real and synthetic radar data, which was a key goal at the outset of this thesis and assists to lift the burden of data collection towards the enablement of robustly trained deep learning algorithms for radar-based drone surveillance.

- Sec. 8.2.2 presented an overview of the latent representations learned by a simple CAE of the L-band bird and drone spectrogram collection. A 3-convolutional layer CAE, different to before called the *Simple*CAE, was used which was better tailored to produce meaningful latent representations faster, with unsupervised training on a greater number of epochs of either 30 or greater. Labelled latent scatter plots showed the latent representation to possess structure with separation of dataset classes and features. Training and test samples were reproduced at the output of the CAE showing that the representation preserved the input data. Arbitrary sampling of the decoder did not consistently generate realistic spectrogram samples, but target and background features were present.

- Sec. 8.2.3 also using the *Simple*CAE, compared the class separation of the latent variables learned by the unsupervised CAE with the supervised accuracy after fine-tuning on real data. Generally, it was shown that better unsupervised separation of classes led to a better supervised accuracy. It was shown that training with a very low learning rate will not likely lead to good separation of labels with an unsupervised metric, but after fine-tuning, a classifier is able to get results comparable to those achieved with more interpretable latent representations.

Chapter 9

Conclusions and Future Work

In this final chapter, the original findings shared in this thesis will be reviewed in Section 9.1. Directions of future work where they may complement and extend the techniques developed, are suggested in Section 9.2.

9.1 Conclusions

This work aimed to assist in the subject of radar-based drone detection. The chief issue existing in this domain is the differentiation of detected, small airborne targets as to whether they contain drone presence or not. In the Introduction (Chapter 1), radar systems were shown to be a powerful and flexible drone surveillance choice amongst a variety of different options. It was argued that radar is possibly the most robust of methods for such a surveillance application, as their operation does not require ambient visible light, can be weather-proof, and can operate over substantial ranges. The classification of such targets in radar is imperative to allowing such systems to robustly defend and warn against an active drone threat, and not be confused with the detection of ambient bird targets. Chapter 2 reviewed the relevant fundamental radar techniques, with a focus on drone targets and Doppler processing. Chapter 3 reviewed a selection of common machine learning methods and mechanisms, with a survey of these methods as applied in radar

studies.

Chapter 4 subsequently detailed the collection of a large amount of real, urban bird and drone data that were to be used for benchmarking an existing classifier approach, as well as to be used later to develop novel classifier training and architecture methods. The collected dataset is uniquely challenging with comparison to existing radar literature due to the following.

1. Using L-band staring radar 4 km from the centre of a major city is a particularly challenging. The high levels of ground and building clutter in the radars field of view, combined with the 90° fan transmit, and smaller but still substantial, receive beams together mean measurements contain a lot of stationary returns, as well as returns from multiple targets.

2. The use of two L-band staring radar systems in the urban environment that are of the same make and model but with different gain, power and phase noise profiles both in the urban environment provided a degree of signature variation that would be far greater than that expected from a single system.

3. Finally, the dataset features 4 different drones that were measured by the radar at ranges up to 4 km, and birds were gathered in their natural environment up to ranges of 5 km and beyond shown a wide variety of natural returns featuring many species and flocking occurrences.

Chapter 5 used a transfer learned Convolutional Neural Network (CNN) to gauge the classification performance on different configurations of drone targets. The AlexNet CNN was used for fine-tuning with the collected radar drone and bird dataset. On the LDB dataset that featured only large drone targets where the rotor-blade μDoppler was visible to the L-band radar, the CNN was able to classify bird and drone targets to above 98.5% accuracy similarly to other works (Sec. 5.2.3). Two smaller drones that did not produce detectable μDoppler due to their plastic rotor blades being too short posed a more challenging classification task (SDB). The μDoppler-less drone targets were classified to 89.2% alongside more similar bird targets. Using a mix of large, μDoppler present and small, μDoppler absent drone targets (ADB) forced the

classifier to learn features for drone classification in excess of the μDoppler. This classifier performed with 90.7% overall accuracy, with 94.0% large drone accuracy and 73.3% small drone accuracy. This chapter thus reinforced that the μDoppler is very important for drone classification if it is detected, and targets without μDoppler may still be classified with slightly less reliability. The challenge to classify small drone targets was made difficult by using a short dwell of 5 seconds for classification samples, alongside a diverse collection of bird signatures that at L-band frequently did not display significant μDoppler but did had a lot of signature variation.

To ease the reliance of deep learning classifiers on available, collected real data, the creation of synthetic data was investigated and reported as a major contribution of this thesis. Chapter 6 described an existing helicopter timeseries signal approximation, which was adapted for drones in a number of more recent works by considering more than one rotating motor structure. The derivation of this signal was shown *ab initio* and was used throughout in the creation of synthetic data. Work prior to this failed to fully consider or implement the reality that the motor speeds of a flying drone are highly fluctuating, and so the simulation parameters of the drone's (four) motor speeds were given close attention. By extracting real motor speeds from a drone flight, constructing realistic simulated data from the flight logs and the injecting into a real recorded radar background, some highly convincing synthetic spectrograms have been produced and the understanding of the technique improved.

In addition, in Chapter 7, synthetic drone data with different resampling options of real recorded motor speeds was investigated for its efficacy when used as synthetic training data for classification real spectrograms. While increased amounts of training data is one of the key motivations of synthetic data creation, instead the approach adopted was to use the real-trained benchmark established in Chapter 5 with swapping of the real drones producing μDoppler with equivalent synthetic drones. Thus, a synthetic dataset was constructed that was an attempted replication of the real benchmark dataset, enabling the performance to be evaluated as a function of the synthetic data fidelity. The highest fidelity synthetic μDoppler data that was introduced in Chapter

7 (Multi-rotor, Sub-CPI sampling) performed within 3% of the real benchmark, while mixing real and synthetic drone data showed a 1.5% improvement over the wholly real bench-marked data. Investigations on lower fidelity synthetic data with a single motor and fixed motor speeds emphatically showed that if given the choice between the use of multiple rotors or a single motor with real speeds, the realistic speeds are the more important aspect to get correct when testing on real data. The approach is instantly flexible to all radar systems so this area is ripe for future work understanding the effect of different implementations of the proposed approach.

Chapter 8 presented an exploration of unsupervised learning to support drone-bird radar classification with an assessment of the Convolutional Autoencoder (CAE). Section 8.2 used the gathered bird-drone dataset with an elementary CAE model to rigorously assess its general utility, as well as if it can provide improvements to classification over the more established methods. Sec. 8.2.1 showed a CAE pre-trained on a small (~7,000 training size), real radar dataset and fine-tuned on the same real data ($\mathcal{R}$CNN) was able to out-perform the results from the well established optically transfer learned AlexNet ($\mathcal{A}$CNN) by 0.11% for the LDB dataset, and by 0.67% for the ADB dataset. The synthetic dataset from Chapter 7 used for pre-training, fine-tuned on real data was also able to marginally outperform the $\mathcal{A}$CNN again by 0.11% for LDB, and by 0.45% for ADB. Sec. 8.2.2 explored the latent representations produced by a similar CAE, by assessing the outputs of its component parts, prior to supervised fine-tuning, to numerically and qualitatively assess the performance of the unsupervised results. For the real dataset of drones featuring μDoppler and opportune birds, the representations learned showed a natural separation of target labels of unseen, test spectrograms, which shows the promise for enabling explainability to the deep learning approach. Finally, Sec. 8.2.3 showed CAE performance as a function of quality of latent representation. We showed that to achieve optimal supervised fine-tuned performance: either training the CAE with a very low learning rate gains reliable results; or alternatively, by optimising the learning rate for the quality of latent representation at the unsupervised stage which can better the performance, but is harder to optimise in the supervised fine tuning stage. Overall, and in particular with the result of synthetic CAE pre-training being effectively identical to that of real data, the approaches shared in the final Chapters 6-8

will allow radar engineers to develop classifiers for bird and drone data that are robustly data driven, that are not hamstrung by an impassable data collection burden.

Table 9.1 lists the main novel results of this thesis with reference to their presentation.

9.2 Future Work

We will finish with the following discussion of possible future work that would complement the novel findings of this work, some of which is already underway at UoB and possibly independently elsewhere:

1. *Extracting Real Motor Speeds for All Drones*

   In Section 6.3, motor speeds from real DJI drone flights were extracted and were essential for the production of the results that followed. It should be noted that the smaller *Mini* and *Mavic* drones did not allow the extraction of their motor speeds, as their SDK is not compatible with the current version of CsvView [274]. This was not an issue for this work at L-band, but it will be at higher frequencies where smaller drones will produce strong μDoppler. One solution is to approximate motor speeds of measured drones, using a drone that has motor speed truth data available for the development of an algorithm. One such approach has been published, co-authored by the author [6], that use a peak detector and knowledge of the harmonic grouping of the sidebands. Otherwise, simulation of motor speeds from a real flow trajectory as was done in [90] is a valid approach, as more generally would be to simulate the drone and/or flight itself using Paparazzi [272] or MAVERIC [273]. The further development and validation of simulations made with either of these approaches will benefit greatly with the use of real recorded motor speeds performed in this work.

2. *Further Exploitation of Real Motor Speeds*

   With the link between the drones' motor speeds and their appearance in radar spectrograms shown in this and other work, much further exploitation can be done using true motor

Table 9.1: Summary of key results presented in the thesis.

Section	Result	Reference
Sec. 5.2	The classification benchmarking of urban, bird and drone L-band staring radar spectrograms using the AlexNet Convolutional Neural Network.	Table 5.9 Fig. 5.3, 5.4, 5.5
Sec. 6.1.3	The consideration of rotor speeds for a multi-rotor μDoppler drone simulation.	Equation 6.9 Fig. 6.4
Sec. 6.3.2	Exploration of the resampling of a recorded motor speed signal to bring improved fidelity of μDoppler signatures.	Fig. 6.7 and 6.8
Sec. 7.2	Synthetic training using entirely simulated drone μDoppler to fall 3.1% below the real baseline of Sec. 5.2.	Table 7.3 Figure 7.4
Sec. 7.2	Synthetic training using a hybrid dataset of real and simulated drone μDoppler, showed a 0.9% improvement over the real baseline of Sec. 5.	Table 7.5 Figure 7.5
Sec. 8.2.1	The training of a CAE similar to the structure of AlexNet (*Alexnet-like* CAE_3) with fine-tuning on the same radar dataset, outperforming AlexNet transfer learned using the same radar data by 0.11% and 0.67%.	Table 8.3 Fig. 8.4
Sec. 8.2.1	The pretraining of *Alexnet-like* CAE_3 with only synthetic μDoppler and with fine-tuning using equivalent real data, performing equivalently or better to AlexNet transfer learned using equivalent real data by 0.45% and 0%.	Table 8.4 Fig. 8.5
Sec. 8.2.2	Analysis of the latent space of an elementary CAE trained with bird and drone radar data, showcasing labelled latent scatter plots, test and training sample reconstructions, and decoder sampling.	Fig. 8.6 - 8.13
Sec. 8.2.3	Assessment of latent variable separability via kNN compared against subsequent supervised fine-tuning classification performance.	Fig. 8.15

speed values. At first, synthetic data based classification could be further improved with a more refined simulation approach than the Sub- and Per-CPI sampling used in Chapter 7. Additionally, as the drone's motor speeds are also a function of it's operating objective, plus accounting for environmental factors such as pressure and wind, it is conceivable to map the features of the drone's motor speeds to their operation in some way, such as, determining whether the drone is controlled by joy-stick or by pre-programmed way-points. The separating features between these two classes would be very subtle, but the ability to test this approach with realistic synthetic data, perhaps with deep learning approaches, will be an asset to its development.

3. *Further Showcasing of Multi-Rotor Synthetic Data Advantage*

   As already stated, the synthetic dataset used in Chapters 7 and 8 was the same size and with similar level of signature variation to the real data used for comparisons. Two natural questions of how synthetic data can be used to benefit classification of real data are:

   - *Further Variation of μDoppler signatures*

     We phenomenologically attenuated the sidebands in the simulation to decay uniformly and symmetrically. Real drone data is rarely perfectly symmetrical due to shielding and other factors. A stochastic, asymmetrical treatment of sideband power will likely increase the robustness of features learned from synthetic data.

   - *Further Variation of Real Background*

     The noise and clutter backgrounds used in the synthetic data were the same as their real equivalent flight. More widely varying the backgrounds used is an easy and effective way to ensure robust classification within the operational setting of the data. In the urban environment, we were restricted to where we could safely fly. Synthetic data could be injected into radar returns from tall buildings or along roads or other dangerous places of operation, that are the harder places to detect drones in the first instance.

- *Extending the Quantities of Data*

  In deep learning, more data is generally better. However, it is possible that using massive amounts of synthetic data could lead to overfitting in an un-predictable way, reducing performance on unseen test data. A survey of the risk and merits to using increased quantities of synthetic data would be interesting, especially with the use of real data amongst synthetic data.

4. *Explainable Classification using Unsupervised Representations*

   The learned representations of bird and drone data in Section 8.2.2 showed an interpretable gradation of features in the labelled latent scatter plots. This could be used to assist and bring interpretability for classification, for example, a small drone is flagged as a possible threat, but it is also accompanied by a large bird flock, the system could provide more intelligent situational awareness alongside a probability distribution of likely targets.

5. *Improved Representations of Radar Returns*

   The representations learned by the basic CAE shown in Section 8.2.2 are an example of what autoencoders can achieve. However, their distribution in the latent space could be made more uniform, and the reconstructions and generated samples could also be improved greatly. The autoencoder extensions detailed in Section 8.1.2 require more data and more steps in essential training algorithm, possibly enabled with increased quantities with synthetic data. If truly robust latent representations of radar target data are achieved, then so arises the end-user challenge of communicating the gathered information across to the security staff operators.

References

[1] D. White, M. Jahangir, M. Antoniou, *et al.*, "Multi-rotor Drone Micro-Doppler Simulation Incorporating Genuine Motor Speeds and Validation with L-band Staring Radar," in *2022 IEEE Radar Conference (RadarConf22)*, IEEE, Mar. 2022, pp. 1–6, ISBN: 978-1-7281-5368-1. DOI: 10.1109/RadarConf2248738.2022.9764352.

[2] D. White, M. Jahangir, J. P. Wayman, *et al.*, "Bird and Micro-Drone Doppler Spectral Width and Classification," in *2023 24th International Radar Symposium (IRS)*, vol. 2023-May, IEEE, May 2023, pp. 1–10, ISBN: 9783944976341. DOI: 10.23919/IRS57608.2023.10172408.

[3] D. White, M. Jahangir, C. J. Baker, and M. Antoniou, "Urban Bird-Drone Classification With Synthetic Micro-Doppler Spectrograms," *IEEE Transactions on Radar Systems*, vol. 2, pp. 167–179, 2024, ISSN: 2832-7357. DOI: 10.1109/TRS.2023.3326317.

[4] D. White, M. Jahangir, C. J. Baker, M. Antoniou, and J. Thiyagalingam, "Comparing optical transfer learning and autoencoder pretraining for radar bird-drone discrimination," Unpublished - To appear at 2024 IEEE Radar Conference (RadarConf24), Denver, Colarado, USA, 2024.

[5] G. Atkinson, M. Jahangir, D. White, *et al.*, "The Role of Target Signatures in Bird-Drone Classification," in *2023 IEEE Radar Conference (RadarConf23)*, San Antonio: IEEE, May 2023, pp. 1–6, ISBN: 978-1-6654-3669-4. DOI: 10.1109/RadarConf2351548.2023.10149665.

[6] X. Ren, M. Jahangir, D. White, G. M. Atkinson, C. J. Baker, and M. Antoniou, "Estimating physical parameters from multi-rotor drone spectrograms," in *International Conference on Radar Systems (RADAR 2022)*, Edinburgh: Institution of Engineering and Technology, 2023, pp. 20–25, ISBN: 978-1-83953-777-6. DOI: 10.1049/icp.2022.2285.

[7] M. Jahangir, G. M. Atkinson, D. White, *et al.*, "Networked staring radar testbed for urban surveillance: status and preliminary results," pp. 471–476, 2023. DOI: 10.1049/icp.2022.2363.

[8] M. Jahangir and D. White, "Good practices and approaches for counter UAV system developments – an industrial perspective," in *Radar Countermeasures for Unmanned Aerial Vehicles*, Institution of Engineering and Technology, Oct. 2021, pp. 363–383. DOI: 10.1049/SBRA543E{_}ch12. [Online]. Available: https://digital-library.theiet.org/content/books/10.1049/sbra543e_ch12.

[9] M. Jahangir, D. Griffiths, D. White, *et al.*, "Development of a networked photonic-enabled staring radar testbed for urban surveillance," *IET Radar, Sonar and Navigation*, no. October, 2023, ISSN: 17518792. DOI: 10.1049/rsn2.12524.

[10] D. Griffiths, M. Jahangir, D. White, *et al.*, "Urban Clutter Analysis for Drone Detection using L-band Staring Radar," *2023 IEEE International Radar Conference (RADAR)*, pp. 1–6, 2023. DOI: 10.1109/radar54928.2023.10371017.

[11] J. P. Wayman, G. Atkinson, M. Jahangir, *et al.*, "L-band radar quantifies major disturbance of birds by fireworks in an urban area," *Scientific Reports*, vol. 13, no. 1, pp. 1–11, 2023, ISSN: 20452322. DOI: 10.1038/s41598-023-39223-1.

[12] *Drone regulations map of the world*, https://www.droneregulations.info/.

[13] D. Kunertova, "Drones have boots: Learning from Russia's war in Ukraine," *Contemporary Security Policy*, vol. 44, no. 4, pp. 576–591, 2023, ISSN: 17438764. DOI: 10.1080/13523260.2023.2262792. [Online]. Available: https://doi.org/10.1080/13523260.2023.2262792.

[14] J. Wang, Y. Liu, and H. Song, "Counter-Unmanned Aircraft System(s) (C-UAS): State of the Art, Challenges, and Future Trends," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 3, pp. 4–29, Mar. 2021, ISSN: 1557959X. DOI: 10.1109/MAES.2020.3015537. [Online]. Available: https://ieeexplore.ieee.org/document/9374654/.

[15] R. N. Kashi, A. Prashanth, S. R. Kashi, and G. Prabhakara, "A survey and analysis of drone detection systems using a systems approach superposed on scenarios," *Systems Engineering*, vol. 27, no. 3, pp. 598–636, 2024, ISSN: 15206858. DOI: 10.1002/sys.21735.

[16] J. Gong, J. Yan, D. Li, and R. Chen, "Using Radar Signatures to Classify Bird Flight Modes between Flapping and Gliding," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 9, pp. 1518–1522, 2020, ISSN: 15580571. DOI: 10.1109/LGRS.2019.2949027.

[17] J. Spruyt and P. van Dorp, "Detection of Birds by Radar," TNO Physics and Electronics Laboratory, The Hague, Tech. Rep., 1996, pp. 5–40.

[18] J. Antonucci, "a Statistical Model of Radar Bird Clutter At the Dew Line," Rome Laboratory, Air Force Systems Command, Tech. Rep. May, 1991.

[19] J. Yan, H. Hu, J. Gong, D. Kong, and D. Li, "Exploring Radar Micro-Doppler Signatures for Recognition of Drone Types," *Drones*, vol. 7, no. 4, 2023, ISSN: 2504446X. DOI: 10.3390/drones7040280.

[20] V. Semkin, J. Haarla, T. Pairon, *et al.*, "Analyzing Radar Cross Section Signatures of Diverse Drone Models at mmWave Frequencies," *IEEE Access*, vol. 8, pp. 48 958–48 969, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2979339.

[21] Z. Zhang and L. Zhu, "A Review on Unmanned Aerial Vehicle Remote Sensing: Platforms, Sensors, Data Processing Methods, and Applications," *Drones*, vol. 7, no. 6, 2023, ISSN: 2504446X. DOI: 10.3390/drones7060398.

[22] A. Bekar, M. Antoniou, and C. J. Baker, "Low-Cost, High-Resolution, Drone-Borne SAR Imaging," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022, ISSN: 15580644. DOI: 10.1109/TGRS.2021.3085235.

[23] L. Earl, J. Vovrosh, M. Wright, *et al.*, "Demonstration of a Compact Magneto-Optical Trap on an Unstaffed Aerial Vehicle," *Atoms*, vol. 10, no. 1, pp. 1–8, 2022, ISSN: 22182004. DOI: 10.3390/atoms10010032.

[24] N. Amarasingam, A. S. Ashan Salgadoe, K. Powell, L. F. Gonzalez, and S. Natarajan, "A review of UAV platforms, sensors, and applications for monitoring of sugarcane crops," *Remote Sensing Applications: Society and Environment*, vol. 26, no. September 2021, p. 100 712, 2022, ISSN: 23529385. DOI: 10.1016/j.rsase.2022.100712. [Online]. Available: https://doi.org/10.1016/j.rsase.2022.100712.

[25] N. Schiller, M. Chlosta, M. Schloegel, *et al.*, "Drone Security and the Mysterious Case of DJI's DroneID," in *30th Annual Network and Distributed System Security Symposium, NDSS 2023*, 2023, ISBN: 1891562835. DOI: 10.14722/ndss.2023.24217.

[26] F. M. Alotaibi, A. Al-Dhaqm, Y. D. Al-Otaibi, and A. A. Alsewari, "A Comprehensive Collection and Analysis Model for the Drone Forensics Field," *Sensors*, vol. 22, no. 17, p. 6486, Aug. 2022, ISSN: 1424-8220. DOI: 10.3390/s22176486. [Online]. Available: https://www.mdpi.com/1424-8220/22/17/6486.

[27] J. S. Patel, F. Fioranelli, and D. Anderson, "Review of radar classification and RCS characterisation techniques for small UAVs or drones," *IET Radar, Sonar and Navigation*, vol. 12, no. 9, pp. 911–919, 2018, ISSN: 17518784. DOI: 10.1049/iet-rsn.2018.0020.

[28] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019, ISSN: 1573756X. DOI: 10.1007/s10618-019-00619-1.

[29] M. Inggs, H. Griffiths, F. Fioranelli, M. Ritchie, and K. Woodbridge, "Multistatic radar: System requirements and experimental validation," *2014 International Radar Conference, Radar 2014*, 2014. DOI: 10.1109/RADAR.2014.7060435.

[30] D. Griffiths, M. Jahangir, J. Kannanthara, C. J. Baker, M. Antoniou, and Y. Singh, "Direct signal synchronization for staring passive bistatic radar," in *International Conference on Radar Systems (RADAR 2022)*, Edinburgh: Institution of Engineering and Technology,

2022, pp. 220–225, ISBN: 978-1-83953-777-6. DOI: 10.1049/icp.2022.2319. [Online]. Available: https://digital-library.theiet.org/content/conferences/10.1049/icp.2022.2319.

[31] C. J. Crosby, "Lidar and Google Earth: Simplifying access to high-resolution topography data," in *Google Earth and Virtual Visualizations in Geoscience Education and Research*, Geological Society of America, 2012. DOI: 10.1130/2012.2492(03). [Online]. Available: https://pubs.geoscienceworld.org/books/book/658/chapter/3806949.

[32] M. Manduhu, A. Dow, P. Trslic, G. Dooly, B. Blanck, and J. Riordan, "Airborne Sense and Detect of Drones using LiDAR and adapted PointPillars DNN," vol. 14, no. 8, pp. 1–14, 2023. [Online]. Available: http://arxiv.org/abs/2310.09589.

[33] X. Shi, Z. Huang, D. Li, *et al.*, "FlowFormer++: Masked Cost Volume Autoencoding for Pretraining Optical Flow Estimation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2023, pp. 1599–1610. DOI: 10.1109/CVPR52729.2023.00160. [Online]. Available: https://ieeexplore.ieee.org/document/10204766/.

[34] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," May 2023. [Online]. Available: http://arxiv.org/abs/2305.09972.

[35] A. Ouaknine, A. Newson, P. Pérez, F. Tupin, and J. Rebut, "Multi-View Radar Semantic Segmentation," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 15 651–15 660, 2021, ISSN: 15505499. DOI: 10.1109/ICCV48922.2021.01538.

[36] J. Liu, W. Xiong, L. Bai, *et al.*, "Deep Instance Segmentation With Automotive Radar Detection Points," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 84–94, 2023, ISSN: 23798858. DOI: 10.1109/TIV.2022.3168899.

[37] J. Domhof, K. F. Julian, and K. M. Dariu, "An extrinsic calibration tool for radar, camera and lidar," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 8107–8113, 2019, ISSN: 10504729. DOI: 10.1109/ICRA.2019.8794186.

[38] S. Al-Emadi, A. Al-Ali, and A. Al-Ali, "Audio-based drone detection and identification using deep learning techniques with dataset enhancement through generative adversarial networks†," *Sensors*, vol. 21, no. 15, 2021, ISSN: 14248220. DOI: 10.3390/s21154953.

[39] A. Shelley, "Drone Registration Will Not Prevent Another Gatwick," *SSRN Electronic Journal*, no. March 2018, pp. 1–14, 2019, ISSN: 1556-5068. DOI: 10.2139/ssrn.3378277. [Online]. Available: https://www.ssrn.com/abstract=3378277.

[40] H. M. Swartz, B. B. Williams, B. I. Zaki, *et al.*, "Clinical EPR," *Academic Radiology*, vol. 21, no. 2, pp. 197–206, Feb. 2014, ISSN: 10766332. DOI: 10.1016/j.acra.2013.10.011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1076633213004996.

[41] Steffen Heuel and Roland Minihold, "Coexistence Test of LTE and Radar Systems Application Note," *Rohde & Schwarz*, 2014. [Online]. Available: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1ma211/1MA211_0e_Coexistence_Test_of_LTE_and_Radar.pdf.

[42] D. S. Zrnic, V. M. Melnikov, and J. K. Carter, "Calibrating differential reflectivity on the WSR-88D," *Bulletin of the American Meteorological Society*, vol. 87, no. 10, pp. 1316–1317, 2006, ISSN: 00030007.

[43] A. Nekrasov, A. Khachaturian, J. Labun, P. Kurdel, and M. Bogachev, "Towards the Sea Ice and Wind Measurement by a C-Band Scatterometer at Dual VV/HH Polarization: A Prospective Appraisal," *Remote Sensing*, vol. 12, no. 20, p. 3382, Oct. 2020, ISSN: 2072-4292. DOI: 10.3390/rs12203382. [Online]. Available: https://www.mdpi.com/2072-4292/12/20/3382.

[44] Abuthahir. C, Karthikeyan. B, Udayakumar. R, and Silamboli. J, "Design of S-band and C-band Modified Plus Shaped Slotted PIFA Antenna for Radar Application," *International Journal of Engineering Research and*, vol. V4, no. 03, pp. 317–321, Mar. 2015, ISSN: 2278-0181. DOI: 10.17577/IJERTV4IS030423. [Online]. Available: http://www.ijert.org/view-pdf/12596/design-of-s-band-and-c-band-modified-plus-shaped-slotted-pifa-antenna-for-radar-application.

[45] H. Vettikalladi, W. T. Sethi, A. F. B. Abas, W. Ko, M. A. Alkanhal, and M. Himdi, "Sub-THz Antenna for High-Speed Wireless Communication Systems," *International Journal of Antennas and Propagation*, vol. 2019, 2019, ISSN: 16875877. DOI: 10.1155/2019/9573647.

[46] E. Marchetti, A. G. Stove, E. G. Hoare, M. Cherniakov, D. Blacknell, and M. Gashinova, "Space-Based Sub-THz ISAR for Space Situational Awareness—Concept and Design," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 1558–1573, Jun. 2022, ISSN: 0018-9251. DOI: 10.1109/TAES.2021.3126375. [Online]. Available: https://ieeexplore.ieee.org/document/9622879/.

[47] B. I. Ahmad and S. Harman, "Tracking of target body and micro-Doppler components in drone surveillance radar," pp. 260–265, 2023. DOI: 10.1049/icp.2022.2327.

[48] F. Ahmad and M. G. Amin, "Coarray-domain RELAX algorithm with source number estimation for direction finding," pp. 154–159, 2023. DOI: 10.1049/icp.2022.2308.

[49] P. Larrenie, C. L. Buron, and F. Barbaresco, "Tracking mulitple targets with multiple radars using Distributed Auctions," *Proceedings International Radar Symposium*, vol. 2023-May, pp. 1–10, 2023, ISSN: 21555753. DOI: 10.23919/IRS57608.2023.10172431.

[50] F. Hoffmann, M. Ritchie, F. Fioranelli, A. Charlish, and H. Griffiths, "Micro-Doppler based detection and tracking of UAVs with multistatic radar," in *2016 IEEE Radar Conference (RadarConf)*, IEEE, May 2016, pp. 1–6, ISBN: 978-1-5090-0863-6. DOI: 10.1109/RADAR.2016.7485236. [Online]. Available: https://ieeexplore.ieee.org/document/7485236/.

[51] A. Charlish, "Autonomous agents for multi-function radar resource management," *Doctoral thesis, UCL (University College London).*, 2011. [Online]. Available: http://discovery.ucl.ac.uk/1334115/.

[52] F. Goodyer, B. I. Ahmad, and S. Godsill, "GaPP: Multi-Target Tracking with Gaussian Processes," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal*

Processing - Proceedings, vol. 2023-June, pp. 1–5, 2023. DOI: 10.1109/ICASSP49357.2023.10095337.

[53] F. Goodyer, B. I. Ahmad, and S. Godsill, "Flexible Multi-Target Tracking with Track Management Using Dirichlet and Gaussian Processes," in *2023 26th International Conference on Information Fusion (FUSION)*, IEEE, Jun. 2023, pp. 1–8. DOI: 10.23919/FUSION52260.2023.10224170. [Online]. Available: https://ieeexplore.ieee.org/document/10224170/.

[54] T. Martelli, F. Colone, and R. Cardinali, "DVB-T based passive radar for simultaneous counter-drone operations and civil air traffic surveillance," *IET Radar, Sonar and Navigation*, vol. 14, no. 4, pp. 505–515, 2020, ISSN: 17518784. DOI: 10.1049/iet-rsn.2019.0309.

[55] Y. Dan, J. Yi, X. Wan, Y. Rao, and B. Wang, "LTE-based passive radar for drone detection and its experimental results," *The Journal of Engineering*, vol. 2019, no. 20, pp. 6910–6913, 2019, ISSN: 2051-3305. DOI: 10.1049/joe.2019.0583.

[56] R. Maksymiuk, M. Plotka, K. Abratkiewicz, and P. Samczynski, "5G Network-Based Passive Radar for Drone Detection," *Proceedings International Radar Symposium*, vol. 2023-May, pp. 1–10, 2023, ISSN: 21555753. DOI: 10.23919/IRS57608.2023.10172437.

[57] D. H. Park, G. H. Park, J. H. Park, J. H. Bang, D. Kim, and H. N. Kim, "Interference Suppression for an FM-Radio-Based Passive Radar via Deep Convolutional Autoencoder," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 1, pp. 106–118, 2024, ISSN: 15579603. DOI: 10.1109/TAES.2023.3240668.

[58] J. C. Maxwell, "A dynamical theory of the electromagnetic field," *Philosophical Transactions of the Royal Society of London*, vol. 155, pp. 459–512, 1965.

[59] P. W. Van Der Walt and W. Steyn, "Characterizing Phase Noise in Self-Referencing Radar," *TechRxiv*, vol. 38, no. 12, pp. 4–13, 2023, ISSN: 1557959X. DOI: 10.1109/MAES.2023.3323919.

[60] A. W. Doerry, "Comments on radar interference sources and mitigation techniques," *Radar Sensor Technology XIX; and Active and Passive Signatures VI*, vol. 9461, p. 94611X, 2015, ISSN: 1996756X. DOI: 10.1117/12.2075743.

[61] F. Uysal and S. Sanka, "Mitigation of automotive radar interference," *2018 IEEE Radar Conference, RadarConf 2018*, pp. 405–410, 2018. DOI: 10.1109/RADAR.2018.8378593.

[62] N. R. Rypkema, "A Straightforward Derivation of the Matched Filter," no. 1, pp. 1–7, 2019. DOI: https://doi.org/10.31219/osf.io/gsjx3.

[63] J. N. Reddy, *Introduction to the Finite Element Method*, 2nd, J. J. Corrigan and J. M. Morris, Eds. McGraw-Hill Education, 2013, ISBN: 9781848211551. DOI: 10.1002/9781118557877.ch7. [Online]. Available: https://www.accessengineeringlibrary.com/content/book/9781259861901.

[64] J. Xu and G. Xie, "A Novel Hybrid Method of Spatially Filtered FDTD and Subgridding Technique," *IEEE Access*, vol. 7, pp. 85 622–85 626, 2019, ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2925835.

[65] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of modern radar: Basic principles*. 2010, vol. I, pp. 1–925, ISBN: 9781613531488. DOI: 10.1049/sbra021e.

[66] R. C. Johnson and H. Jasik, *Antenna Engineering Handbook*, 1st ed. McGraw-Hill, 1961, vol. 23, pp. 37–37, ISBN: 0-07-032381-X.

[67] IEEE Aerospace & Electronic Systems Society, "IEEE Standard Letter Designations for Radar-Frequency Bands," IEEE, Tech. Rep., 2003.

[68] R. Sun, K. Suzuki, Y. Owada, *et al.*, "A millimeter-wave automotive radar with high angular resolution for identification of closely spaced on-road obstacles," *Scientific Reports*, vol. 13, no. 1, pp. 1–15, 2023, ISSN: 20452322. DOI: 10.1038/s41598-023-30406-4. [Online]. Available: https://doi.org/10.1038/s41598-023-30406-4.

[69] C. A. Balanis, *Antenna theory: analysis and design*. John wiley & sons, 2016.

[70] M. A. Richards, *Fundamentals of radar signal processing*, 3rd. McGraw-Hill, 2008, ISBN: 9780071798327. [Online]. Available: https://www.accessengineeringlibrary.com/content/book/9780071798327%0Ahttps://www.accessengineeringlibrary.com/content/book/9780071798327.abstract.

[71] "Hilbert Transforms," in *Transforms and Applications Handbook*, A. D. Poularikas, Ed., CRC Press, Sep. 2018, ch. 7, p. 100, ISBN: 9781315218915. DOI: 10.1201/9781315218915. [Online]. Available: https://www.taylorfrancis.com/books/9781420066531.

[72] C. M. Rader, "A Simple Method For Sampling In-Phase And Quadrature Components," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 6, pp. 821–824, Nov. 1984, ISSN: 00189251. DOI: 10.1109/TAES.1984.310466. [Online]. Available: http://ieeexplore.ieee.org/document/4103989/.

[73] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965, ISSN: 0025-5718. DOI: 10.1090/S0025-5718-1965-0178586-1. [Online]. Available: https://www.ams.org/mcom/1965-19-090/S0025-5718-1965-0178586-1/.

[74] S. Harman, "A comparison of staring radars with scanning radars for UAV detection: Introducing the Alarm™ staring radar," *2015 European Radar Conference, EuRAD 2015 - Proceedings*, pp. 185–188, 2015. DOI: 10.1109/EuRAD.2015.7346268.

[75] QinetiQ, *Obsidian™ counter drone system*, https://www.qinetiq.com/en/what-we-do/services-and-products/counter-drone-systems.

[76] T. Dai, L. Mei, Y. Zhang, *et al.*, "UAVs and birds classification using robust coordinate attention synergy residual split-attention network based on micro-Doppler signature measurement by using L-band staring radar," *Measurement*, vol. 222, no. October, p. 113 692, Nov. 2023, ISSN: 02632241. DOI: 10.1016/j.measurement.2023.113692. [Online]. Available: https://doi.org/10.1016/j.measurement.2023.113692%20https://linkinghub.elsevier.com/retrieve/pii/S0263224123012563.

[77] P. Beasley, M. Ritchie, H. Griffiths, *et al.*, "Multistatic Radar Measurements of UAVs at X-band and L-band," *IEEE National Radar Conference - Proceedings*, vol. 2020-Septe, 2020, ISSN: 10975659. DOI: 10.1109/RadarConf2043947.2020.9266444.

[78] C. Schroeder and H. Rohling, "X-band FMCW radar system with variable chirp duration," *IEEE National Radar Conference - Proceedings*, pp. 1255–1259, 2010, ISSN: 10975659. DOI: 10.1109/RADAR.2010.5494425.

[79] S. Rahman and D. A. Robertson, "Radar micro-Doppler signatures of drones and birds at K-band and W-band," *Scientific Reports*, vol. 8, no. 1, pp. 1–11, 2018, ISSN: 20452322. DOI: 10.1038/s41598-018-35880-9. [Online]. Available: http://dx.doi.org/10.1038/s41598-018-35880-9.

[80] H. Dale, M. Antoniou, C. J. Baker, M. Jahangir, and A. Catherall, "Convolutional Neural Networks for Drone Model Classification," in *2021 18th European Radar Conference, EuRAD 2021*, London: IEEE, Apr. 2021, pp. 361–364, ISBN: 9782874870651. DOI: 10.23919/EuRAD50154.2022.9784563. [Online]. Available: https://ieeexplore.ieee.org/document/9784563/.

[81] I. Lam, S. Pant, M. Manning, *et al.*, "Time-Frequency Analysis using V-band Radar for Drone Detection and Classification," *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, vol. 2023-May, pp. 1–6, 2023, ISSN: 10915281. DOI: 10.1109/I2MTC53148.2023.10176027.

[82] R. Mamgain, R. Jain, and D. Deb, "Study and Simulation of Radar Targets' Micro-Doppler Signature," *2018 International Conference on Radar, RADAR 2018*, no. 1, pp. 1–5, 2018. DOI: 10.1109/RADAR.2018.8557264.

[83] V. Agnihotri, M. Sabharwal, and V. Goyal, "Effect of frequency on micro-doppler signatures of a helicopter," *icABCD 2019 - 2nd International Conference on Advances in Big Data, Computing and Data Communication Systems*, pp. 1–5, 2019. DOI: 10.1109/ICABCD.2019.8851024.

[84] Y. Cai, *Radar micro-doppler patterns for drone's characterisation*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:181669323.

[85] M. Ritchie, F. Fioranelli, H. Griffiths, and B. Torvik, "Micro-drone RCS analysis," *2015 IEEE Radar Conference - Proceedings*, pp. 452–456, 2015. DOI: 10.1109/RadarConf.2015.7411926.

[86] C. R. Vaughn, "Birds and Insects as Radar Targets: A Review," *Proceedings of the IEEE*, vol. 73, no. 2, pp. 205–227, 1985, ISSN: 15582256. DOI: 10.1109/PROC.1985.13134.

[87] P. Klaer, A. Huang, P. Sévigny, *et al.*, "An Investigation of Rotary Drone HERM Line Spectrum under Manoeuvering Conditions," *Sensors*, pp. 11–15, 2020. DOI: 10.3390/s20205940. [Online]. Available: https://www.mdpi.com/1424-8220/20/20/5940.

[88] J. S. Patel, C. Al-Ameri, F. Fioranelli, and D. Anderson, "Multi-time frequency analysis and classification of a micro-drone carrying payloads using multistatic radar," *The Journal of Engineering*, vol. 2019, no. 20, pp. 7047–7051, 2019, ISSN: 2051-3305. DOI: 10.1049/joe.2019.0551.

[89] J. Duan, L. Zhang, Y. Wu, *et al.*, "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, Feb. 2023, ISSN: 16711793. DOI: 10.23919/JSEE.2023.000002. [Online]. Available: https://ieeexplore.ieee.org/document/10043786/.

[90] C. Bennett, S. Harman, and I. Petrunin, "Realistic Simulation of Drone Micro-Doppler Signatures," *2021 18th European Radar Conference, EuRAD 2021*, no. April, pp. 114–117, 2021. DOI: 10.23919/EuRAD50154.2022.9784488.

[91] J. Markow and A. Balleri, "Examination of Drone Micro-Doppler and JEM/HERM Signatures," *IEEE National Radar Conference - Proceedings*, vol. 2020-Septe, no. 2, 2020, ISSN: 10975659. DOI: 10.1109/RadarConf2043947.2020.9266342.

[92] B. Choi and D. Oh, "Classification of Drone Type Using Deep Convolutional Neural Networks Based on Micro- Doppler Simulation," *ISAP 2018 - 2018 International Symposium on Antennas and Propagation*, no. Isap, pp. 403–404, 2019.

[93] R. M. Narayanan, B. Tsang, and R. Bharadwaj, "Classification and Discrimination of Birds and Small Drones Using Radar Micro-Doppler Spectrogram Images," *Signals*, vol. 4, no. 2, pp. 337–358, 2023. DOI: 10.3390/signals4020018.

[94] S. Rahman and D. A. Robertson, "Classification of drones and birds using convolutional neural networks applied to radar micro-Doppler spectrogram images," *IET Radar, Sonar and Navigation*, vol. 14, no. 5, pp. 653–661, 2020, ISSN: 17518784. DOI: 10.1049/iet-rsn.2019.0493.

[95] J. Gong, J. Yan, D. Li, D. Kong, and H. Hu, "Interference of radar detection of drones by birds," *Progress In Electromagnetics Research M*, vol. 81, no. February, pp. 1–11, 2019, ISSN: 19378726. DOI: 10.2528/pierm19020505.

[96] J. Gong, J. Yan, D. Li, and D. Kong, "Low detectable JEM signals in radar echoes from a helicopter with composite blades," *Proceedings International Radar Symposium*, vol. 2022-Septe, no. 1, pp. 235–240, 2022, ISSN: 21555753. DOI: 10.23919/irs54158.2022.9905038.

[97] A. K. Singh and Y. H. Kim, "Automatic Measurement of Blade Length and Rotation Rate of Drone Using W-Band Micro-Doppler Radar," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 1895–1902, 2018, ISSN: 1530437X. DOI: 10.1109/JSEN.2017.2785335.

[98] M. S. Seyfioglu and S. Z. Gurbuz, "Deep Neural Network Initialization Methods for Micro-Doppler Classification with Low Training Sample Support," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2462–2466, 2017, ISSN: 1545598X. DOI: 10.1109/LGRS.2017.2771405.

[99] M. S. Seyfioğlu, A. M. Özbayoğlu, and S. Z. Gürbüz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1709–1723, 2018, ISSN: 00189251. DOI: 10.1109/TAES.2018.2799758.

[100] B. Erol, S. Z. Gurbuz, and M. G. Amin, "GAN-based synthetic radar micro-doppler augmentations for improved human activity recognition," in *2019 IEEE Radar Conference,*

RadarConf 2019, IEEE, 2019, pp. 1–5, ISBN: 9781728116792. DOI: 10.1109/RADAR.2019.8835589.

[101] S. Z. Gurbuz, M. M. Rahman, E. Kurtoglu, *et al.*, "Multi-Frequency RF Sensor Fusion for Word-Level Fluent ASL Recognition," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 11 373–11 381, 2022, ISSN: 15581748. DOI: 10.1109/JSEN.2021.3078339.

[102] B. Jokanovic and M. Amin, "Fall Detection Using Deep Learning in Range-Doppler Radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 180–189, 2018, ISSN: 00189251. DOI: 10.1109/TAES.2017.2740098.

[103] H. Hameed, M. Usman, A. Tahir, *et al.*, "Recognizing British Sign Language Using Deep Learning: A Contactless and Privacy-Preserving Approach," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 2090–2098, 2023, ISSN: 2329924X. DOI: 10.1109/TCSS.2022.3210288.

[104] F. X. Hofele and A. Hanewinkel, "Automatic Radar Target Classification A New Idea for Distinguishing Drones and Birds from the Invention to Serial Production," *Proceedings International Radar Symposium*, vol. 2023-May, pp. 1–10, 2023, ISSN: 21555753. DOI: 10.23919/IRS57608.2023.10172402.

[105] R. I. Harmanny, J. J. De Wit, and G. Prémel Cabic, "Radar micro-Doppler feature extraction using the spectrogram and the cepstrogram," *European Microwave Week 2014: "Connecting the Future", EuMW 2014 - Conference Proceedings; EuRAD 2014: 11th European Radar Conference*, pp. 165–168, 2014. DOI: 10.1109/EuRAD.2014.6991233.

[106] S. Rahman and D. Robertson, "Time-Frequency Analysis of Millimeter-Wave Radar Micro-Doppler Data from Small UAVs," *2017 Sensor Signal Processing for Defence Conference, SSPD 2017*, vol. 2017-Janua, pp. 1–5, 2017. DOI: 10.1109/SSPD.2017.8233269.

[107] E. Hunter, D. Raval, B. Balaji, and A. Damini, "Convolutional neural networks and wavelets for drone classification," in *Radar Sensor Technology XXVI*, A. M. Raynal and

K. I. Ranney, Eds., Orlando: SPIE, May 2022, p. 25, ISBN: 9781510650923. DOI: 10.1117/12.2618832. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12108/2618832/Convolutional-neural-networks-and-wavelets-for-drone-classification/10.1117/12.2618832.full.

[108] H. Park and J. H. Son, "Machine learning techniques for thz imaging and time-domain spectroscopy," *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–25, 2021, ISSN: 14248220. DOI: 10.3390/s21041186.

[109] F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978, ISSN: 15582256. DOI: 10.1109/PROC.1978.10837.

[110] I. Choi, K. B. Kang, J. H. Jung, and S. H. Park, "Efficient Estimation of the Helicopter Blade Parameter by Independent Component Analysis," *IEEE Access*, vol. 8, pp. 156 889–156 899, 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3019506.

[111] D. Blacknell and H. Griffiths, *Radar Automatic Target Recognition (ATR) and Non-Cooperative Target Recognition (NCTR)*, D. Blacknell and H. Griffiths, Eds. IET, 2013, ISBN: 978-1-84919-685-7. [Online]. Available: https://shop.theiet.org/radar-auto-targ-recognit.

[112] Y. Huang, Z. Ren, Z. Guo, D. Yang, and W. Yi, "Swarm Behavior Recognition Using VGG16-based Network with Multi-dimensional Parameters," IRS, 2022, pp. 261–266, ISBN: 9788395602047.

[113] J. Yun, D. Anderson, and F. Fioranelli, "Parametric Investigation on Simulated Staring FMCW Radar for Anti-Drone Swarms," *IEEE National Radar Conference - Proceedings*, vol. 2020-Septe, no. 2, 2020, ISSN: 10975659. DOI: 10.1109/RadarConf2043947.2020.9266354.

[114] C. Conte, S. V. Supplizi, G. de Alteriis, A. Mele, G. Rufino, and D. Accardo, "Using Drone Swarms as a Countermeasure of Radar Detection," *Journal of Aerospace Information Systems*, vol. 20, no. 2, pp. 70–80, 2023, ISSN: 23273097. DOI: 10.2514/1.I011131.

[115] J. Gong, J. Yan, D. Li, and R. Chen, "Comparison of radar signatures based on flight morphology for large birds and small birds," *IET Radar, Sonar and Navigation*, vol. 14, no. 9, pp. 1365–1369, 2020, ISSN: 17518784. DOI: 10.1049/iet-rsn.2020.0064.

[116] J. Gong, J. Yan, D. Li, and R. Chen, "Using Radar Signatures to Classify Bird Flight Modes between Flapping and Gliding," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 9, pp. 1518–1522, 2020, ISSN: 15580571. DOI: 10.1109/LGRS.2019.2949027.

[117] P. Molchanov, R. I. Harmanny, J. J. De Wit, K. Egiazarian, and J. Astola, "Classification of small UAVs and birds by micro-Doppler signatures," *International Journal of Microwave and Wireless Technologies*, vol. 6, no. 3-4, pp. 435–444, 2014, ISSN: 17590795. DOI: 10.1017/S1759078714000282.

[118] D. Dhulashia, N. Peters, C. Horne, P. Beasley, and M. Ritchie, "Multi-Frequency Radar Micro-Doppler Based Classification of Micro-Drone Payload Weight," *Frontiers in Signal Processing*, vol. 1, no. December, pp. 1–14, 2021. DOI: 10.3389/frsip.2021.781777.

[119] B. I. Ahmad, J. Grey, M. Newman, and S. Harman, "Low-latency Convolutional Neural Network for Classification of Previously Unseen Drone Types," *2022 19th European Radar Conference, EuRAD 2022*, no. September, pp. 189–192, 2022. DOI: 10.23919/EuRAD54643.2022.9924966.

[120] H. Dale, C. J. Baker, M. Antoniou, M. Jahangir, and G. Atkinson, "A Comparison of Convolutional Neural Networks for Low SNR Radar Classification of Drones," *IEEE National Radar Conference - Proceedings*, vol. 2021-May, pp. 1–5, 2021, ISSN: 10975659. DOI: 10.1109/RadarConf2147009.2021.9455181.

[121] H. C. Kumawat, M. Chakraborty, A. A. Bazil Raj, and S. V. Dhavale, "DIAT-μSAT: micro-Doppler Signature Dataset of Small Unmanned Aerial Vehicle (SUAV)," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, ISSN: 15580571. DOI: 10.21227/1x2q-8v62. [Online]. Available: https://dx.doi.org/10.21227/1x2q-8v62.

[122] Y. Cai, O. Krasnov, and A. Yarovoy, "Simulation of Radar Micro-Doppler Patterns for Multi-Propeller Drones," *2019 International Radar Conference, RADAR 2019*, pp. 4–8, 2019. DOI: 10.1109/RADAR41533.2019.171372.

[123] D. Park, S. Lee, S. U. Park, and N. Kwak, "Radar-spectrogram-based UAV classification using convolutional neural networks," *Sensors (Switzerland)*, vol. 21, no. 1, pp. 1–18, 2021, ISSN: 14248220. DOI: 10.3390/s21010210.

[124] H. Dale, C. J. Baker, M. Antoniou, M. Jahangir, G. Atkinson, and S. Harman, "SNR-dependent drone classification using convolutional neural networks," *IET Radar, Sonar and Navigation*, vol. 16, no. 1, pp. 22–33, 2022, ISSN: 17518784. DOI: 10.1049/rsn2.12161. [Online]. Available: https://doi.org/10.1049/rsn2.12161.

[125] L. Lehmann and J. Dall, "Simulation-based Approach to Classification of Airborne Drones," in *2020 IEEE Radar Conference (RadarConf20)*, vol. 2020-Septe, IEEE, Sep. 2020, pp. 1–6, ISBN: 978-1-7281-8942-0. DOI: 10.1109/RadarConf2043947.2020.9266405. [Online]. Available: https://ieeexplore.ieee.org/document/9266405/.

[126] S. Yoon, S. Kim, J. Jung, *et al.*, "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, ISSN: 21693536. DOI: 10.1109/ACCESS.2022.3155776.

[127] 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, Feb. 2023, ISSN: 16711793. DOI: 10.23919/JSEE.2023.000002. [Online]. Available: https://ieeexplore.ieee.org/document/10043786/.

[128] M. Ritchie, F. Fioranelli, H. Borrion, and H. Griffiths, "Multistatic micro-Doppler radar feature extraction for classification of unloaded/loaded micro-drones," *IET Radar, Sonar and Navigation*, vol. 11, no. 1, pp. 116–124, 2017, ISSN: 17518784. DOI: 10.1049/iet-rsn.2016.0063.

[129] D. Raval, E. Hunter, S. Hudson, A. Damini, and B. Balaji, "Convolutional neural networks for classification of drones using radars," *Drones*, vol. 5, no. 4, p. 149, Dec. 2021, ISSN: 2504446X. DOI: 10.3390/drones5040149. [Online]. Available: https://www.mdpi.com/2504-446X/5/4/149.

[130] D. J. Hand, P. Christen, and N. Kirielle, "F*: an interpretable transformation of the F-measure," *Machine Learning*, vol. 110, no. 3, pp. 451–456, 2021, ISSN: 15730565. DOI: 10.1007/s10994-021-05964-1. [Online]. Available: https://doi.org/10.1007/s10994-021-05964-1.

[131] S. Mokhtari, K. K. Yen, and J. Liu, "Effectiveness of Artificial Intelligence in Stock Market Prediction based on Machine Learning," *International Journal of Computer Applications*, vol. 183, no. 7, pp. 1–8, Jun. 2021, ISSN: 09758887. DOI: 10.5120/ijca2021921347. [Online]. Available: http://www.ijcaonline.org/archives/volume183/number7/mokhtari-2021-ijca-921347.pdf.

[132] L. Y. Hu, M. W. Huang, S. W. Ke, and C. F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *SpringerPlus*, vol. 5, no. 1, 2016, ISSN: 21931801. DOI: 10.1186/s40064-016-2941-7.

[133] V. Pestov, "Is the k-NN classifier in high dimensions affected by the curse of dimensionality?" *Computers and Mathematics with Applications*, vol. 65, no. 10, pp. 1427–1437, 2013, ISSN: 08981221. DOI: 10.1016/j.camwa.2012.09.011. [Online]. Available: http://dx.doi.org/10.1016/j.camwa.2012.09.011.

[134] L. F. C. Ruiz, L. A. Guasselli, A. t. Caten, and D. C. Zanotta, "Iterative K–Nearest Neighbors Algorithm (IKNN) for submeter spatial resolution image classification obtained by Unmanned Aerial Vehicle (UAV)," *International Journal of Remote Sensing*, vol. 39, no. 15-16, pp. 5043–5058, 2018, ISSN: 13665901. DOI: 10.1080/01431161.2018.1444296. [Online]. Available: https://doi.org/10.1080/01431161.2018.1444296.

[135] D. K. Srivastava and L. Bhambhu, "Data classification using support vector machine," *Journal of Theoretical and Applied Information Technology*, vol. 12, no. 1, pp. 1–7, 2010, ISSN: 18173195.

[136] S. Ding, Z. Zhu, and X. Zhang, "An overview on semi-supervised support vector machine," *Neural Computing and Applications*, vol. 28, no. 5, pp. 969–978, 2017. DOI: 10.1007/s00521-015-2113-7.

[137] J. T. Vogelstein, E. W. Bridgeford, M. Tang, *et al.*, "Supervised dimensionality reduction for big data," *Nature Communications*, vol. 12, no. 1, pp. 1–9, 2021, ISSN: 20411723. DOI: 10.1038/s41467-021-23102-2. [Online]. Available: http://dx.doi.org/10.1038/s41467-021-23102-2.

[138] Z. Pengfei, L. Gang, H. Chaoying, and Y. Hongcheng, "Classification of drones based on micro-Doppler radar signatures using dual radar sensors," *Journal of Radars*, vol. 7, no. 5, pp. 557–564, 2018, ISSN: 2095283X. DOI: 10.12000/JR18061.

[139] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI Communications*, vol. 30, no. 2, pp. 169–190, 2017, ISSN: 09217126. DOI: 10.3233/AIC-170729.

[140] L. van der Maaten and G. E. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html.

[141] S. Arora, W. Hu, and P. K. Kothari, "An Analysis of the t-SNE Algorithm for Data Visualization," *Proceedings of Machine Learning Research*, vol. 75, no. 2002, pp. 1455–1462, 2018, ISSN: 26403498.

[142] D. M. Chan, R. Rao, F. Huang, and J. F. Canny, "GPU accelerated t-distributed stochastic neighbor embedding," *Journal of Parallel and Distributed Computing*, vol. 131, pp. 1–13, 2019, ISSN: 07437315. DOI: 10.1016/j.jpdc.2019.04.008. [Online]. Available: https://doi.org/10.1016/j.jpdc.2019.04.008.

[143] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," 2018. [Online]. Available: http://arxiv.org/abs/1802.03426.

[144] E. Becht, L. McInnes, J. Healy, *et al.*, "Dimensionality reduction for visualizing single-cell data using UMAP," *Nature Biotechnology*, vol. 37, no. 1, pp. 38–47, 2019, ISSN: 15461696. DOI: 10.1038/nbt.4314.

[145] C. J. Nolet, V. Lafargue, E. Raff, *et al.*, "Bringing UMAP Closer to the Speed of Light with GPU Acceleration," *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 1, pp. 418–426, 2021, ISSN: 2159-5399. DOI: 10.1609/aaai.v35i1.16118.

[146] G. Franch, G. Jurman, L. Coviello, M. Pendesini, and C. Furlanello, "MASS-UMAP: Fast and accurate analog ensemble search in weather radar archives," *Remote Sensing*, vol. 11, no. 24, pp. 1–20, 2019, ISSN: 20724292. DOI: 10.3390/rs11242922.

[147] X. Liang, B. Chen, W. Chen, P. Wang, and H. Liu, "Unsupervised Radar Target Detection under Complex Clutter Background Based on Mixture Variational Autoencoder," *Remote Sensing*, vol. 14, no. 18, 2022, ISSN: 20724292. DOI: 10.3390/rs14184449.

[148] K. N. Parashar, M. C. Oveneke, M. Rykunov, H. Sahli, and A. Bourdoux, "Micro-Doppler feature extraction using convolutional auto-encoders for low latency target classification," in *2017 IEEE Radar Conference (RadarConf)*, IEEE, May 2017, pp. 1739–1744, ISBN: 978-1-4673-8823-8. DOI: 10.1109/RADAR.2017.7944488. [Online]. Available: http://ieeexplore.ieee.org/document/7944488/.

[149] S. Kearney and S. Z. Gurbuz, "Influence of Radar Signal Processing on Deep Learning-based Classification," in *Proceedings of the IEEE Radar Conference*, vol. 2023-May, IEEE, 2023, pp. 1–5, ISBN: 9781665436694. DOI: 10.1109/RadarConf2351548.2023.10149612.

[150] Y. Karayaneva, S. Sharifzadeh, W. Li, Y. Jing, and B. Tan, "Unsupervised Doppler Radar Based Activity Recognition for e-Healthcare," *IEEE Access*, vol. 9, pp. 62 984–63 001, 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3074088.

[151] J. Guo, L. Wang, D. Zhu, and C. Hu, "Compact convolutional autoencoder for SAR target recognition," *IET Radar, Sonar and Navigation*, vol. 14, no. 7, pp. 967–972, 2020, ISSN: 17518784. DOI: 10.1049/iet-rsn.2019.0447.

[152] Y. Gao, Y. Zhou, Y. Wang, and Z. Zhuo, "Narrowband Radar Automatic Target Recognition Based on a Hierarchical Fusing Network with Multidomain Features," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 6, pp. 1039–1043, 2021, ISSN: 15580571. DOI: 10.1109/LGRS.2020.2993039.

[153] L. Rokach and O. Maimon, "Chapter 9: Decision Trees," in *Data Mining and Knowledge Discovery Handbook*, Figure 1, 2005, pp. 165–192, ISBN: 9784431538592. DOI: 10.1007/0-387-25465-X{_}9. [Online]. Available: https://www.researchgate.net/publication/225237661_Decision_Trees.

[154] L. E. Raileanu and K. Stoffel, "Theoretical Comparison between the Gini Index and Information Gain Criteria," *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77–93, May 2004, ISSN: 1012-2443. DOI: 10.1023/B:AMAI.0000018580.96245.c6. [Online]. Available: http://link.springer.com/10.1023/B:AMAI.0000018580.96245.c6.

[155] A. Cutler, D. R. Cutler, and J. R. Stevens, *Ensemble Machine Learning*. 2012, ISBN: 9781441993267. DOI: 10.1007/978-1-4419-9326-7.

[156] O. Steinki and Z. Mohammad, "Introduction to Ensemble Learning," *SSRN Electronic Journal*, no. July, 2015, ISSN: 1556-5068. DOI: 10.2139/ssrn.2634092. [Online]. Available: http://www.ssrn.com/abstract=2634092.

[157] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," *Journal of Machine Learning Research*, vol. 20, pp. 97–112, 2011, ISSN: 15324435.

[158] Y. Hamasuna, Y. Endo, and S. Miyamoto, "Support vector machine for data with tolerance based on hard-margin and soft-margin," *IEEE International Conference on Fuzzy Systems*, pp. 750–755, 2008, ISSN: 10987584. DOI: 10.1109/FUZZY.2008.4630454.

[159] Y. Kim and H. Ling, "Human activity classification based on micro-doppler signatures using a support vector machine," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1328–1337, 2009, ISSN: 01962892. DOI: 10.1109/TGRS.2009.2012849.

[160] M. Jahangir, B. I. Ahmad, and C. J. Baker, "Robust drone classification using two-stage decision trees and results from SESAR SAFIR trials," *2020 IEEE International Radar Conference, RADAR 2020*, pp. 636–641, 2020. DOI: 10.1109/RADAR42522.2020.9114870.

[161] C. Bennett, M. Jahangir, F. Fioranelli, B. I. Ahmad, and J. L. Kernec, "Use of Symmetrical Peak Extraction in Drone Micro-Doppler Classification for Staring Radar," in *2020 IEEE Radar Conference (RadarConf20)*, vol. 2020-Septe, IEEE, Sep. 2020, pp. 1–6, ISBN: 978-1-7281-8942-0. DOI: 10.1109/RadarConf2043947.2020.9266702. [Online]. Available: https://ieeexplore.ieee.org/document/9266702/.

[162] L. Pallotta, C. Clemente, A. Raddi, and G. Giunta, "A Feature-Based Approach for Loaded/Unloaded Drones Classification Exploiting micro-Doppler Signatures," *IEEE National Radar Conference - Proceedings*, vol. 2020-Septe, 2020, ISSN: 10975659. DOI: 10.1109/RadarConf2043947.2020.9266458.

[163] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, ISSN: 18792782. DOI: 10.1016/j.neunet.2014.09.003. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2014.09.003.

[164] S. Sathyanarayana, "A Gentle Introduction to Backpropagation," *Numeric Insight*, no. July 2014, pp. 1–15, 2014.

[165] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, 2016. [Online]. Available: http://arxiv.org/abs/1609.04747.

[166] A. Paszke, S. Gross, F. Massa, *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2019, pp. 8026–8037. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[167] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. January 1995, pp. 255–258, 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.9297&rep=rep1&type=pdf.

[168] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, *A survey of the recent architectures of deep convolutional neural networks*. Springer Netherlands, 2020, vol. 53, pp. 5455–5516, ISBN: 0123456789. DOI: 10.1007/s10462-020-09825-6. [Online]. Available: https://doi.org/10.1007/s10462-020-09825-6.

[169] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," Jul. 2016. [Online]. Available: http://arxiv.org/abs/1607.06450.

[170] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning, ICML 2015*, vol. 1, Feb. 2015, pp. 448–456, ISBN: 9781510810587. [Online]. Available: http://arxiv.org/abs/1502.03167.

[171] Z. Dai and R. Heckel, "Channel Normalization in Convolutional Neural Network avoids Vanishing Gradients," pp. 1–13, Jul. 2019. [Online]. Available: http://arxiv.org/abs/1907.09539.

[172] H. Gholamalinezhad and H. Khosravi, "Pooling Methods in Deep Neural Networks, a Review," Sep. 2020. [Online]. Available: http://arxiv.org/abs/2009.07485.

[173] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016, ISSN: 10636919. DOI: 10.1109/CVPR.2016.90.

[174] D. Brooks, O. Schwander, F. Barbaresco, *et al.*, "Temporal Deep Learning for Drone Micro-Doppler Classification To cite this version : HAL Id : hal-02290839 Temporal Deep Learning for Drone Micro-Doppler Classification HAL Id : hal-02290839," 2019.

[175] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, ISSN: 08997667. DOI: 10.1162/neco.1997.9.8.1735.

[176] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017, ISSN: 10495258.

[177] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, no. 1, pp. 111–132, 2022, ISSN: 26666510. DOI: 10.1016/j.aiopen.2022.10.001.

[178] W. X. Zhao, K. Zhou, J. Li, *et al.*, "A Survey of Large Language Models," pp. 1–124, 2023. [Online]. Available: http://arxiv.org/abs/2303.18223.

[179] S. Bubeck, V. Chandrasekaran, R. Eldan, *et al.*, "Sparks of Artificial General Intelligence: Early experiments with GPT-4," 2023. [Online]. Available: http://arxiv.org/abs/2303.12712.

[180] M. Ufer, V. Mhasawade, R. Graef, H. Appel, and T. Brosch, "Micro-Doppler based Deep Learning approaches for radar applications," in *2023 24th International Radar Symposium (IRS)*, Berlin: IEEE, May 2023, pp. 1–8, ISBN: 978-3-944976-34-1. DOI: 10.23919/IRS57608.2023.10172465. [Online]. Available: https://ieeexplore.ieee.org/document/10172465/.

[181] K. Yamazaki, V. K. Vo-Ho, D. Bulsara, and N. Le, *Spiking Neural Networks and Their Applications: A Review*, 2022. DOI: 10.3390/brainsci12070863.

[182] Z. Xia, G. Ding, H. Wang, and F. Xu, "Person Identification with Millimeter-Wave Radar in Realistic Smart Home Scenarios," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, ISSN: 15580571. DOI: 10.1109/LGRS.2021.3117001.

[183] B. Vogginger, F. Kreutz, J. López-Randulfe, *et al.*, "Automotive Radar Processing With Spiking Neural Networks: Concepts and Challenges," *Frontiers in Neuroscience*, vol. 16, no. April, pp. 1–19, Apr. 2022, ISSN: 1662-453X. DOI: 10.3389/fnins.2022.851774.

[184] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, Dec. 2013. [Online]. Available: http://arxiv.org/abs/1312.6114.

[185] W. H. Lopez Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Autoencoders," in *Machine Learning*, Elsevier, Mar. 2020, ch. 11, pp. 193–208, ISBN: 9780128157398. DOI: 10.1016/B978-0-12-815739-8.00011-0. [Online]. Available: http://arxiv.org/abs/2003.05991%20https://linkinghub.elsevier.com/retrieve/pii/B9780128157398000110.

[186] M. Meyer, S. Nekkah, G. Kuschk, and S. Tomforde, "Automotive Object Detection on Highly Compressed Range-Beam-Doppler Radar Data," *2022 19th European Radar Conference, EuRAD 2022*, no. September, pp. 141–144, 2022. DOI: 10.23919/EuRAD54643.2022.9924839.

[187] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, "Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review," *Sensors*, vol. 20, no. 15, p. 4220, Jul. 2020, ISSN: 1424-8220. DOI: 10.3390/s20154220. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4220.

[188] S. Chen and W. Guo, "Auto-Encoders in Deep Learning—A Review with New Perspectives," *Mathematics*, vol. 11, no. 8, 2023, ISSN: 22277390. DOI: 10.3390/math11081777.

[189] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO: From YOLOv1 and Beyond," pp. 1–36, 2023. [Online]. Available: http://arxiv.org/abs/2304.00501.

[190] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative Adversarial Networks," *Machine Learning*, vol. 112, no. 12, pp. 5135–5161, Jun. 2014, ISSN: 0885-6125. DOI: 10.1007/s10994-023-06367-0.

[191] A. Safa, T. Verbelen, O. Catal, *et al.*, "FMCW Radar Sensing for Indoor Drones Using Variational Auto-Encoders," *Proceedings of the IEEE Radar Conference*, vol. 2023-May, pp. 1–6, 2023, ISSN: 23755318. DOI: 10.1109/RadarConf2351548.2023.10149738.

[192] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 9, pp. 201–208, 2010, ISSN: 15324435.

[193] M. M. Rahman and S. Z. Gurbuz, "Self-Supervised Contrastive Learning for Radar-Based Human Activity Recognition," in *Proceedings of the IEEE Radar Conference*, vol. 2023-May, IEEE, May 2023, pp. 1–6, ISBN: 9781665436694. DOI: 10.1109/RadarConf2351548.2023.10149770. [Online]. Available: https://ieeexplore.ieee.org/document/10149770/.

[194] S. Wagner, W. Johannes, D. Qosja, and S. Bruggenwirth, "Small Target Detection in a Radar Surveillance System Using Contractive Autoencoders," *IEEE Transactions on Aerospace and Electronic Systems*, vol. XX, no. Xx, 2023, ISSN: 15579603. DOI: 10.1109/TAES.2023.3253469.

[195] M. Tschannen, O. Bachem, and M. Lucic, "Recent Advances in Autoencoder-Based Representation Learning," Dec. 2018. [Online]. Available: http://arxiv.org/abs/1812.05069.

[196] *Adaptive detection mode with threshold control as a function of spatially sampled-clutter-level estimates*, 1968.

[197] Y. Wang, Y. Han, C. Wang, S. Song, Q. Tian, and G. Huang, "Computation-efficient Deep Learning for Computer Vision: A Survey," pp. 1–47, Aug. 2023. [Online]. Available: http://arxiv.org/abs/2308.13998.

[198] S. Särkkä, "Bayesian filtering and smoothing," *Bayesian Filtering and Smoothing*, pp. 1–232, 2010. DOI: 10.1017/CBO9781139344203.

[199] S. Ishtiaq, X. Wang, S. Hassan, A. Mohammad, A. A. Alahmadi, and N. Ullah, "Three-Dimensional Multi-Target Tracking Using Dual-Orthogonal Baseline Interferometric Radar," *Sensors*, vol. 22, no. 19, p. 7549, Oct. 2022, ISSN: 1424-8220. DOI: 10.3390/s22197549. [Online]. Available: https://www.mdpi.com/1424-8220/22/19/7549.

[200] N. Mohajerin, J. Histon, R. Dizaji, and S. L. Waslander, "Feature extraction and radar track classification for detecting UAVs in civillian airspace," in *2014 IEEE Radar Conference*, IEEE, May 2014, pp. 0674–0679, ISBN: 9781479920341. DOI: 10.1109/RADAR.2014.6875676. [Online]. Available: https://ieeexplore.ieee.org/document/6875676.

[201] W. S. Chen, J. Liu, and J. Li, "Classification of UAV and bird target in low-altitude airspace with surveillance radar data," *Aeronautical Journal*, vol. 123, no. 1260, pp. 191–211, 2019, ISSN: 00019240. DOI: 10.1017/aer.2018.158.

[202] J. Liu, Q. Y. Xu, and W. S. Chen, "Classification of Bird and Drone Targets Based on Motion Characteristics and Random Forest Model Using Surveillance Radar Data," *IEEE Access*, vol. 9, pp. 160 135–160 144, 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3130231.

[203] W. Zhan, J. Yi, X. Wan, and Y. Rao, "Track-Feature-Based Target Classification in Passive Radar for Low-Altitude Airspace Surveillance," *IEEE Sensors Journal*, vol. 21, no. 8, pp. 10 017–10 028, Apr. 2021, ISSN: 1530-437X. DOI: 10.1109/JSEN.2021.3057706. [Online]. Available: https://ieeexplore.ieee.org/document/9349474/.

[204] E. R. Keydel, S. W. Lee, and J. T. Moore, "MSTAR Extended Operating Conditions: A Tutorial," in *SPIE Defence, Security, and Sensing*, E. G. Zelnio and R. J. Douglass, Eds., Jun. 1996, pp. 228–242. DOI: 10.1117/12.242059. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1018994.

[205] R. Praneetha, T. Dhipu, and R. Rajesh, "SAR Image Classification using Transfer Learning," in *2021 2nd International Conference on Range Technology (ICORT)*, IEEE, Aug. 2021, pp. 1–6, ISBN: 978-1-6654-4956-4. DOI: 10.1109/ICORT52730.2021.9581609. [Online]. Available: https://ieeexplore.ieee.org/document/9581609/.

[206] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional Neural Network with Data Augmentation for SAR Target Recognition," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, pp. 364–368, 2016, ISSN: 1545598X. DOI: 10.1109/LGRS.2015.2513754.

[207] G. Priyanka, R. Gandhirai, and K. P. Soman, "Target Recognition in SAR using Deep Learning," *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2019*, pp. 1442–1447, 2019. DOI: 10.1109/ICICICT46008.2019.8993162.

[208] S. Zaied, A. Toumi, and A. Khenchaf, "Target classification using convolutional deep learning and auto-encoder models," *2018 4th International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2018*, pp. 1–6, 2018. DOI: 10.1109/ATSIP.2018.8364502.

[209] M. Tay, "Unsupervised Deep Learning Parameter Estimation for High Fidelity Synthetic Aperture Radar Super Resolution," *Proceedings International Radar Symposium*, vol. 2022-Septe, pp. 241–246, 2022, ISSN: 21555753. DOI: 10.23919/irs54158.2022.9905004.

[210] J. Do, M. Yoo, and S. Kim, "A Semi-supervised SAR Image Classification with Data Augmentation and Pseudo Labeling," in *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, IEEE, Oct. 2022, pp. 1–4, ISBN: 978-1-6654-6434-5. DOI: 10.1109/ICCE-Asia57006.2022.9954846. [Online]. Available: https://ieeexplore.ieee.org/document/9954846/.

[211] Z. Huang, Y. Liu, X. Yao, J. Ren, and J. Han, "Uncertainty Exploration: Toward Explainable SAR Target Detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–14, 2023, ISSN: 15580644. DOI: 10.1109/TGRS.2023.3247898.

[212] N. Blomerus, P. de Villiers, and W. Nel, "Improved Explainability through Uncertainty Estimation in Automatic Target Recognition of SAR Images," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*, IEEE, Nov. 2021, pp. 1–6, ISBN: 978-1-7377497-1-4. DOI: 10.23919/FUSION49465.2021.9627066. [Online]. Available: https://ieeexplore.ieee.org/document/9627066/.

[213] Mandeep, H. S. Pannu, and A. Malhi, "Deep learning-based explainable target classification for synthetic aperture radar images," in *2020 13th International Conference on*

Human System Interaction (HSI), vol. 2020-June, IEEE, Jun. 2020, pp. 34–39, ISBN: 978-1-7281-7392-4. DOI: 10.1109/HSI49210.2020.9142658. [Online]. Available: https://ieeexplore.ieee.org/document/9142658/.

[214] A. Shyam, K. Komalavally, M. Gautam, *et al.*, "An Automotive Radar Dataset For Object Classification," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Jun. 2023, pp. 1–5, ISBN: 978-1-7281-6327-7. DOI: 10.1109/ICASSP49357.2023.10097078. [Online]. Available: https://ieeexplore.ieee.org/document/10097078/.

[215] M. Meyer and G. Kuschk, "Automotive radar dataset for deep learning based 3D object detection," *EuRAD 2019 - 2019 16th European Radar Conference*, pp. 129–132, 2019.

[216] L. Zheng, Z. Ma, X. Zhu, *et al.*, "TJ4DRadSet: A 4D Radar Dataset for Autonomous Driving," pp. 2–7, Apr. 2022. DOI: 10.1109/ITSC55140.2022.9922539. [Online]. Available: https://github.com/TJRadarLab/TJ4DRadSet.%20http://arxiv.org/abs/2204.13483%20http://dx.doi.org/10.1109/ITSC55140.2022.9922539.

[217] A. Pirkani and M. Gashinova, *COSMOS 'Cross Traffic Alert (CTA) dataset' for co-existence/ interference analysis and simultaneous scene representation by automotive radar and video*, Birmingham, 2022. DOI: 10.25500/edata.bham.00000801. [Online]. Available: https://edata.bham.ac.uk/801/.

[218] C. Decourt, R. VanRullen, D. Salle, and T. Oberlin, "DAROD: A Deep Automotive Radar Object Detector on Range-Doppler maps," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, vol. 2022-June, IEEE, Jun. 2022, pp. 112–118, ISBN: 978-1-6654-8821-1. DOI: 10.1109/IV51971.2022.9827281. [Online]. Available: https://ieeexplore.ieee.org/document/9827281/.

[219] F. Jia, J. Tan, X. Lu, and J. Qian, "Radar Timing Range–Doppler Spectral Target Detection Based on Attention ConvLSTM in Traffic Scenes," *Remote Sensing*, vol. 15, no. 17, 2023, ISSN: 20724292. DOI: 10.3390/rs15174150.

[220] Y. Xiao, L. Daniel, and M. Gashinova, "Image Segmentation and Region Classification in Automotive High-Resolution Radar Imagery," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6698–6711, Mar. 2021, ISSN: 1530-437X. DOI: 10.1109/JSEN.2020.3043586. [Online]. Available: https://ieeexplore.ieee.org/document/9288850/.

[221] A. Ouaknine, A. Newson, P. Pérez, F. Tupin, and J. Rebut, "Multi-View Radar Semantic Segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, Montreal, 2021, pp. 15 651–15 660, ISBN: 9781665428125. DOI: 10.1109/ICCV48922.2021.01538. [Online]. Available: https://telecom-paris.hal.science/hal-03324900.

[222] C. Grimm, T. Fei, E. Warsitz, R. Farhoud, T. Breddermann, and R. Haeb-Umbach, "On the RCS Estimation from Camera Images," *Proceedings International Radar Symposium*, vol. 2023-May, pp. 1–12, 2023, ISSN: 21555753. DOI: 10.23919/IRS57608.2023.10172430.

[223] Y. Jin, Y. Kuang, M. Hoffmann, *et al.*, "Radar and Lidar Deep Fusion: Providing Doppler Contexts to Time-of-Flight Lidar," *IEEE Sensors Journal*, vol. 23, no. 20, pp. 25 587–25 600, 2023, ISSN: 15581748. DOI: 10.1109/JSEN.2023.3313093.

[224] L. V. D. Sloover, L. Werthen-Brabants, G. Bhavanasi, T. Dhaene, and I. Couckuyt, "Tailoring Radar-Based Patient Monitoring Models to Real-Life Needs using Utility Maximization," *2022 19th European Radar Conference, EuRAD 2022*, no. September, pp. 81–84, 2022. DOI: 10.23919/EuRAD54643.2022.9924872.

[225] M. Di Seglio, F. Filippini, C. Bongioanni, and F. Colone, "Human and Drone Surveillance via RpF-based WiFi Passive Radar: Experimental Validation," *Proceedings International Radar Symposium*, vol. 2022-Septe, pp. 402–407, 2022, ISSN: 21555753. DOI: 10.23919/irs54158.2022.9904988.

[226] A. Helen Victoria and G. Maragatham, "Activity recognition of FMCW radar human signatures using tower convolutional neural networks," *Wireless Networks*, vol. 7, 2021,

ISSN: 15728196. DOI: 10.1007/s11276-021-02670-7. [Online]. Available: https://doi.org/10.1007/s11276-021-02670-7.

[227] L. Werthen-Brabants, G. Bhavanasi, I. Couckuyt, T. Dhaene, and D. Deschrijver, "Quantifying Uncertainty in Real Time with Split BiRNN for Radar Human Activity Recognition," *2022 19th European Radar Conference, EuRAD 2022*, no. September, pp. 173–176, 2022. DOI: 10.23919/EuRAD54643.2022.9924932.

[228] M. Jia, S. Li, J. L. Kernec, S. Yang, F. Fioranelli, and O. Romain, "Human activity classification with radar signal processing and machine learning," *2020 International Conference on UK-China Emerging Technologies, UCET 2020*, no. Cvd, 2020. DOI: 10.1109/UCET51115.2020.9205461.

[229] S. Yoon, S. Ahmed, and S. H. Cho, "UWB Radar-Based Pet Monitoring on Daily Basis in an Unconstrained Living Environment," *Proceedings International Radar Symposium*, vol. 2023-May, pp. 1–7, 2023, ISSN: 21555753. DOI: 10.23919/IRS57608.2023.10172441.

[230] M. Czerkawski, C. Clemente, C. Michie, I. Andonovic, and C. Tachtatzis, "Robustness of Deep Neural Networks for Micro-Doppler Radar Classification," *Proceedings International Radar Symposium*, vol. 2022-Septe, no. September, pp. 480–485, 2022, ISSN: 21555753. DOI: 10.23919/irs54158.2022.9905017.

[231] M. Mercuri, G. Sacco, R. Hornung, *et al.*, "2-D Localization, Angular Separation and Vital Signs Monitoring Using a SISO FMCW Radar for Smart Long-Term Health Monitoring Environments," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 065–11 077, 2021, ISSN: 23274662. DOI: 10.1109/JIOT.2021.3051580.

[232] K. Han and S. Hong, "MIMO Differential Radar Using Null Point Beams for Vital Sign Detection in the Presence of Body Motions," *2022 19th European Radar Conference, EuRAD 2022*, no. September, pp. 177–180, 2022. DOI: 10.23919/EuRAD54643.2022.9924869.

[233] C. Li and J. Lin, "Random body movement cancellation in doppler radar vital sign detection," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, 2008, pp. 3143–3152. DOI: 10.1109/TMTT.2008.2007139.

[234] A. Alkasimi, T. Shepard, S. Wagner, *et al.*, "Dual-Biometric Human Identification Using Radar Deep Transfer Learning," *Sensors*, vol. 22, no. 15, 2022, ISSN: 14248220. DOI: 10.3390/s22155782.

[235] Y. Huang, E. Jiang, H. Xu, and G. Zhang, "Person identification using a new CNN-based method and radar gait micro-Doppler signatures," *Journal of Physics: Conference Series*, vol. 2258, no. 1, 2022, ISSN: 17426596. DOI: 10.1088/1742-6596/2258/1/012044.

[236] X. Huang, N. Patel, and K. P. Tsoi, "Application of mmWave Radar Sensor for People Identification and Classification," *Sensors*, vol. 23, no. 8, pp. 1–12, 2023, ISSN: 14248220. DOI: 10.3390/s23083873.

[237] P. Zhao, C. X. Lu, J. Wang, *et al.*, "MID: Tracking and identifying people with millimeter wave radar," *Proceedings - 15th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2019*, pp. 33–40, 2019. DOI: 10.1109/DCOSS.2019.00028.

[238] Z. Chen, G. Li, F. Fioranelli, and H. Griffiths, "Dynamic hand gesture classification based on multistatic radar micro-doppler signatures using convolutional neural network," *2019 IEEE Radar Conference, RadarConf 2019*, 2019. DOI: 10.1109/RADAR.2019.8835796.

[239] H. Kulhandjian, P. Sharma, M. Kulhandjian, and C. D'Amours, "Sign language gesture recognition using doppler radar and deep learning," *2019 IEEE Globecom Workshops, GC Wkshps 2019 - Proceedings*, pp. 1–6, 2019. DOI: 10.1109/GCWkshps45667.2019.9024607.

[240] E. Kurtoglu, A. C. Gurbuz, E. Malaia, D. Griffin, C. Crawford, and S. Z. Gurbuz, "RF Micro-Doppler Classification with Multiple Spectrograms from Angular Subspace Projections," *Proceedings of the IEEE Radar Conference*, 2022, ISSN: 23755318. DOI: 10.1109/RadarConf2248738.2022.9763904.

[241] M. M. Rahman, E. Kurtoglu, M. Taskin, *et al.*, "Performance Comparison of Radar and Video for American Sign Language Recognition," *Proceedings of the IEEE Radar Conference*, pp. 1–6, 2022, ISSN: 23755318. DOI: 10.1109/RadarConf2248738.2022.9764269.

[242] H. Haifawi, F. Fioranelli, A. Yarovoy, and R. Van Der Meer, "Drone Detection & Classification with Surveillance 'Radar On-The-Move' and YOLO," *Proceedings of the IEEE Radar Conference*, vol. 2023-May, pp. 1–6, 2023, ISSN: 23755318. DOI: 10.1109/RadarConf2351548.2023.10149588.

[243] E. V. Carrera, F. Lara, M. Ortiz, A. Tinoco, and R. Leon, "Target Detection using Radar Processors based on Machine Learning," in *2020 IEEE ANDESCON*, IEEE, Oct. 2020, pp. 1–5, ISBN: 978-1-7281-9365-6. DOI: 10.1109/ANDESCON50619.2020.9272173. [Online]. Available: https://ieeexplore.ieee.org/document/9272173/.

[244] D. Greig, C. Clemente, M. Antoniou, *et al.*, "Try living in the real world: the importance of experimental radar systems and data collection trials," pp. 459–464, 2023. DOI: 10.1049/icp.2022.2361.

[245] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.

[246] D. Gusland, J. M. Christiansen, B. Torvik, F. Fioranelli, S. Z. Gurbuz, and M. Ritchie, "Open Radar Initiative: Large Scale Dataset for Benchmarking of micro-Doppler Recognition Algorithms," in *2021 IEEE Radar Conference (RadarConf21)*, vol. 2021-May, IEEE, May 2021, pp. 1–6, ISBN: 978-1-7281-7609-3. DOI: 10.1109/RadarConf2147009.2021.9455239. [Online]. Available: https://github.com/openradarinitiative%20https://ieeexplore.ieee.org/document/9455239/.

[247] D. Raval, E. Hunter, I. Lam, S. Rajan, A. Damini, and B. Balaji, "Detecting drones with radars and convolutional networks based on micro-Doppler signatures," *Proceed-*

ings of the IEEE Radar Conference, pp. 1–6, 2022, ISSN: 23755318. DOI: 10.1109/RadarConf2248738.2022.9764249.

[248] R. Fu, M. A. Al-Absi, K. H. Kim, Y. S. Lee, A. A. Al-Absi, and H. J. Lee, "Deep Learning-Based Drone Classification Using Radar Cross Section Signatures at mmWave Frequencies," *IEEE Access*, vol. 9, pp. 161 431–161 444, 2021, ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3115805.

[249] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2017, ISSN: 15577317. DOI: 10.1145/3065386. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf%20https://dl.acm.org/doi/10.1145/3065386.

[250] A. Karlsson, M. Jansson, and M. Hamalainen, "Model-Aided Drone Classification Using Convolutional Neural Networks," in *Proceedings of the IEEE Radar Conference*, IEEE, Mar. 2022, pp. 1–6, ISBN: 9781728153681. DOI: 10.1109/RadarConf2248738.2022.9764194. [Online]. Available: https://ieeexplore.ieee.org/document/9764194/.

[251] H. Dale, M. Jahangir, C. J. Baker, M. Antoniou, S. Harman, and B. I. Ahmad, "Convolutional Neural Networks for Robust Classification of Drones," *Proceedings of the IEEE Radar Conference*, pp. 1–6, 2022, ISSN: 23755318. DOI: 10.1109/RadarConf2248738.2022.9764172.

[252] H. Dale, C. J. Baker, M. Antoniou, and M. Jahangir, "An initial investigation into using convolutional neural networks for classification of drones," *2020 IEEE International Radar Conference, RADAR 2020*, pp. 618–623, 2020. DOI: 10.1109/RADAR42522.2020.9114745.

[253] H. Dale, M. Jahangir, C. J. Baker, M. Antoniou, S. Harman, and B. I. Ahmad, "Convolutional Neural Networks for Robust Classification of Drones," in *2022 IEEE Radar Conference (RadarConf22)*, IEEE, Mar. 2022, pp. 1–6, ISBN: 978-1-7281-5368-1. DOI:

10.1109/RadarConf2248738.2022.9764172. [Online]. Available: https://ieeexplore.ieee.org/document/9764172/.

[254] V. Lafontaine, K. Bouchard, J. Maitre, and S. Gaboury, "Denoising UWB Radar Data for Human Activity Recognition Using Convolutional Autoencoders," *IEEE Access*, vol. 11, no. August, pp. 81 298–81 309, 2023, ISSN: 21693536. DOI: 10.1109/ACCESS.2023.3300224.

[255] N. Regev, I. Yoffe, and D. Wulich, "Classification of single and multi propelled miniature drones using multilayer perceptron artificial neural network," *IET Conference Publications*, vol. 2017, no. CP728, pp. 1–5, 2017. DOI: 10.1049/cp.2017.0378.

[256] S. Kunze and B. Saha, "Drone Classification with a Convolutional Neural Network Applied to Raw IQ Data," *2022 3rd URSI Atlantic and Asia Pacific Radio Science Meeting, AT-AP-RASC 2022*, no. June, pp. 3–6, 2022. DOI: 10.23919/AT-AP-RASC54737.2022.9814170.

[257] M. Jahangir, G. M. Atkinson, M. Antoniou, C. J. Baker, J. P. Sadler, and S. J. Reynolds, "Measurements of Birds and Drones with L-Band Staring Radar," *Proceedings International Radar Symposium*, vol. 2021-June, pp. 1–10, 2021, ISSN: 21555753. DOI: 10.23919/IRS51887.2021.9466224.

[258] M. Jahangir, J. M. Jones, J. Kannanthara, *et al.*, "Development of Quantum Enabled Staring Radar with Low Phase Noise," *2021 18th European Radar Conference, EuRAD 2021*, no. April, pp. 225–228, 2021. DOI: 10.23919/EuRAD50154.2022.9784517.

[259] *Theodolite application for ios. accessed march 2021.* https://apps.apple.com/gb/app/theodolite/id339393884.

[260] H. Dale, "Deep Learning for Radar Classification of Drones," Ph.D. dissertation, University of Birmingham, 2022.

[261] Z. Geng, H. Yan, J. Zhang, and D. Zhu, "Deep-Learning for Radar: A Survey," *IEEE Access*, vol. 9, pp. 141 800–141 818, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3119561. [Online]. Available: https://ieeexplore.ieee.org/document/9568916/.

[262] G. Olivares, T. Lacy, L. Gomez, *et al.*, "UAS airborne collision severity evaluation | Executive Summary – Structural Evaluation," U.S. Department of Transport, Wichita State University, Tech. Rep., 2017, pp. 1–27. [Online]. Available: http://www.assureuas.org/projects/deliverables/a3/Volume%20II%20-%20UAS%20Airborne%20Collision%20Severity%20Evaluation%20-%20Quadcopter.pdf.

[263] G. Habib and S. Qureshi, "Optimization and acceleration of convolutional neural networks: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4244–4268, Jul. 2022, ISSN: 13191578. DOI: 10.1016/j.jksuci.2020.10.004. [Online]. Available: https://doi.org/10.1016/j.jksuci.2020.10.004%20https://linkinghub.elsevier.com/retrieve/pii/S1319157820304845.

[264] K. Kandasamy, K. R. Vysyaraju, W. Neiswanger, *et al.*, "Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly," *Journal of Machine Learning Research*, vol. 21, pp. 1–27, 2020, ISSN: 15337928.

[265] P. Kulmon and P. Lukeš, "Hyperparameters selection analysis of Bayesian deghosting algorithm for MSPSR systems," pp. 714–718, 2023. DOI: 10.1049/icp.2023.1279.

[266] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019, ISSN: 1674862X. DOI: 10.11989/JEST.1674-862X.80904120. [Online]. Available: https://doi.org/10.11989/JEST.1674-862X.80904120.

[267] J. Martin and B. Mulgrew, "Analysis of Theoretical Radar Return Signal from Aircraft Propellor Blades," pp. 569–572, 1990. DOI: 10.1109/radar.1990.201091.

[268] S. Vishwakarma, W. Li, C. Tang, K. Woodbridge, R. Adve, and K. Chetty, "SimHumalator: An Open-Source End-to-End Radar Simulator for Human Activity Recognition," *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 3, pp. 6–22, 2022, ISSN: 1557959X. DOI: 10.1109/MAES.2021.3138948.

[269] M. Stolz, E. Schubert, F. Meinl, M. Kunert, and W. Menzel, "Multi-target reflection point model of cyclists for automotive radar," *European Microwave Week 2017: "A Prime Year for a Prime Event", EuMW 2017 - Conference Proceedings; 14th European Microwave Conference, EURAD 2017*, vol. 2018-Janua, pp. 94–97, 2017. DOI: 10.23919/EURAD.2017.8249155.

[270] V. C. Chen, F. Li, S. S. Ho, and H. Wechsler, "Micro-doppler effect in radar: Phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 2–21, 2006, ISSN: 00189251. DOI: 10.1109/TAES.2006.1603402.

[271] J. Gérard, J. Gerard, J. Gerard, L. Signal, and I. Université, "Drone Recognition by Deep Learning," Ph.D. dissertation, Université Paris-Saclay, 2022.

[272] D. Anderson, *Maveric: Modelling of autonomous vehicles using robust, intelligent computing*. [Online]. Available: https://www.gla.ac.uk/media/Media_480053_smxx.pdfn.

[273] PaparazziUAV, *Paparazzi: The free autopilot*. [Online]. Available: https://wiki.paparazziuav.org/wiki/Simulation.

[274] *CsvView*. [Online]. Available: https://datfile.net/index.html.

[275] Hsieh Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 508–517, Dec. 1978, ISSN: 0096-3518. DOI: 10.1109/TASSP.1978.1163154. [Online]. Available: http://ieeexplore.ieee.org/document/1163154/.

[276] B. I. Ahmad, J. Grey, M. Newman, and S. Harman, "Low-latency convolution neural network for estimating drone physical parameters with radar," in *International Conference on Radar Systems (RADAR 2022)*, Edinburgh: Institution of Engineering and Technology, 2023, pp. 1–6, ISBN: 978-1-83953-777-6. DOI: 10.1049/icp.2022.2282. [Online]. Available: https://digital-library.theiet.org/content/conferences/10.1049/icp.2022.2282.

[277] P. Molchanov, *Radar target classification by micro-Doppler contributions*. 2014, p. 208, ISBN: 16182650.

[278] M. Moore, D. A. Robertson, and S. Rahman, "Simulating UAV micro-Doppler using dynamic point clouds," *Proceedings of the IEEE Radar Conference*, 2022, ISSN: 23755318. DOI: 10.1109/RadarConf2248738.2022.9764284.

[279] S. Kaur and R. Agrawal, "Convolutional networks," *Deep Learning Approaches to Cloud Security*, pp. 109–122, 2021. DOI: 10.1002/9781119760542.ch8.

[280] V. Turchenko, E. Chalmers, and A. Luczak, "A deep convolutional auto-encoder with pooling - unpooling layers in caffe," *International Journal of Computing*, vol. 18, no. 1, pp. 8–31, 2019, ISSN: 23125381. DOI: 10.47839/ijc.18.1.1270.

[281] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2018–2025, 2011. DOI: 10.1109/ICCV.2011.6126474.

[282] W. Zhao, Z. Jia, X. Wei, and H. Wang, "An FPGA implementation of a convolutional auto-encoder," *Applied Sciences (Switzerland)*, vol. 8, no. 4, 2018, ISSN: 20763417. DOI: 10.3390/app8040504.

[283] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep Convolutional AutoEncoder-based Lossy Image Compression," *2018 Picture Coding Symposium, PCS 2018 - Proceedings*, pp. 253–257, 2018. DOI: 10.1109/PCS.2018.8456308.

[284] Y. Watkins, O. Iaroshenko, M. Sayeh, and G. Kenyon, "Image compression: Sparse coding vs. bottleneck autoencoders," *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, vol. 2018-April, pp. 17–20, 2018. DOI: 10.1109/SSIAI.2018.8470336.

[285] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High Fidelity Neural Audio Compression," no. 1990, pp. 1–20, 2022. [Online]. Available: http://arxiv.org/abs/2210.13438.

[286] T. Baltrusaitis, C. Ahuja, and L. P. Morency, *Multimodal Machine Learning: A Survey and Taxonomy*, 2019. DOI: 10.1109/TPAMI.2018.2798607.

[287] S. Tian, Y. Lin, W. Gao, H. Zhang, and C. Wang, "A multi-scale u-shaped convolution auto-encoder based on pyramid pooling module for object recognition in synthetic aperture radar images," *Sensors (Switzerland)*, vol. 20, no. 5, 2020, ISSN: 14248220. DOI: 10.3390/s20051533.

[288] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010, ISSN: 15324435.

[289] H. Steck, "Autoencoders that don\textquotesingle t overfit towards the Identity," *Advances in Neural Information Processing Systems*, vol. 33, no. NeurIPS, pp. 19 598–19 608, 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/e33d974aae13e4d877477d51d8bafdc4-Paper.pdf.

[290] S. Chen, J. Taghia, T. Fei, U. Kühnau, N. Pohl, and R. Martin, "A DNN autoencoder for automotive radar interference mitigation," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2021-June, pp. 4065–4069, 2021, ISSN: 15206149. DOI: 10.1109/ICASSP39728.2021.9413619.

[291] N. Ferreira and M. Silveira, "Ship Detection in SAR Images Using Convolutional Variational Autoencoders," in *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, Sep. 2020, pp. 2503–2506, ISBN: 978-1-7281-6374-1. DOI: 10.1109/IGARSS39084.2020.9324389. [Online]. Available: https://ieeexplore.ieee.org/document/9324389/.

[292] F. Zhao, Y. Liu, K. Huo, S. Zhang, and Z. Zhang, "Radar HRRP target recognition based on stacked autoencoder and extreme learning machine," *Sensors (Switzerland)*, vol. 18, no. 1, pp. 10–15, 2018, ISSN: 14248220. DOI: 10.3390/s18010173.

[293] J. Geng, H. Wang, J. Fan, and X. Ma, "Change detection of SAR images based on supervised contractive autoencoders and fuzzy clustering," in *RSIP 2017 - International Workshop on Remote Sensing with Intelligent Processing, Proceedings*, IEEE, May 2017,

pp. 1–3, ISBN: 9781538619902. DOI: 10.1109/RSIP.2017.7958819. [Online]. Available: http://ieeexplore.ieee.org/document/7958819/.

[294] L. Yang, Y. Wang, Z. Wang, *et al.*, "A new method based on stacked auto-encoders to identify abnormal weather radar echo images," *Eurasip Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 2020, ISSN: 16871499. DOI: 10.1186/s13638-020-01769-3.

[295] G. Czibula, A. Mihai, A. I. Albu, I. G. Czibula, S. Burcea, and A. Mezghani, "Autonowp: An approach using deep autoencoders for precipitation nowcasting based on weather radar reflectivity prediction," *Mathematics*, vol. 9, no. 14, 2021, ISSN: 22277390. DOI: 10.3390/math9141653.

[296] Y. Pu, Z. Gan, R. Henao, *et al.*, "Variational Autoencoder for Deep Learning of Images , Labels and Captions," no. Nips, 2016.

[297] D. P. Kingma, "Variational Inference & Deep Learning: A New Synthesis," *PhD Thesis*, pp. 1–162, 2017.

[298] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning - ICML '08*, vol. 311, New York, New York, USA: ACM Press, 2008, pp. 1096–1103, ISBN: 9781605582054. DOI: 10.1145/1390156.1390294. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1390156.1390294.

[299] D. Lee, S. Choi, and H.-j. Kim, "Performance evaluation of image denoising developed using convolutional denoising autoencoders in chest radiography," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 884, no. December 2017, pp. 97–104, Mar. 2018, ISSN: 01689002. DOI: 10.1016/j.nima.2017.12.050. [Online]. Available: https://doi.org/10.1016/j.nima.2017.12.050%20https://linkinghub.elsevier.com/retrieve/pii/S0168900217314560.

[300] A. Pawar, "Noise reduction in images using autoencoders," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, 2020, pp. 987–990, ISBN: 9781728170893. DOI: 10.1109/ICISS49785.2020.9315908.

[301] K. Bajaj, D. K. Singh, and M. A. Ansari, "Autoencoders Based Deep Learner for Image Denoising," in *Procedia Computer Science*, vol. 171, Elsevier B.V., 2020, pp. 1535–1541, ISBN: 9179055141. DOI: 10.1016/j.procs.2020.04.164. [Online]. Available: https://doi.org/10.1016/j.procs.2020.04.164.

[302] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Advances in Neural Information Processing Systems*, 2013, pp. 1–9.

[303] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations. BT - Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event.," *Icml*, vol. 119, no. Figure 1, pp. 1597–1607, 2020. [Online]. Available: https://proceedings.mlr.press/v119/chen20j.html%20http://proceedings.mlr.press/v119/chen20j.html.

[304] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive Representation Learning: A Framework and Review," *IEEE Access*, vol. 8, pp. 193 907–193 934, Oct. 2020, ISSN: 21693536. DOI: 10.1109/ACCESS.2020.3031549.

[305] M. C. Oveneke, M. Aliosha-Perez, Y. Zhao, D. Jiang, and H. Sahli, "Efficient Convolutional Auto-Encoding via Random Convexification and Frequency-Domain Minimization," no. Nips 2016, pp. 1–6, Nov. 2016. [Online]. Available: http://arxiv.org/abs/1611.09232.

[306] A. Jabbar, X. Li, and B. Omar, "A Survey on Generative Adversarial Networks: Variants, Applications, and Training," *ACM Computing Surveys*, vol. 54, no. 8, pp. 1–49, Nov. 2022, ISSN: 0360-0300. DOI: 10.1145/3463475. [Online]. Available: https://dl.acm.org/doi/10.1145/3463475.

[307] M. Ostovan, S. Samadi, and A. Kazemi, "Generation of Human Micro-Doppler Signature Based on Layer-Reduced Deep Convolutional Generative Adversarial Network," *Computational Intelligence and Neuroscience*, vol. 2022, no. 3, 2022, ISSN: 16875273. DOI: 10.1155/2022/7365544.

[308] F. Li, D. Wu, B. Gu, J. Dai, and W. Li, "Airborne Radar Forward-Looking Imaging Algorithm Based on Generative Adversarial Networks," *Proceedings International Radar Symposium*, vol. 2023-May, pp. 1–8, 2023, ISSN: 21555753. DOI: 10.23919/IRS57608.2023.10172417.

[309] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, "High-Resolution SAR Image Classification via Deep Convolutional Autoencoders," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, pp. 2351–2355, Nov. 2015, ISSN: 1545598X. DOI: 10.1109/LGRS.2015.2478256. [Online]. Available: https://ieeexplore.ieee.org/document/7286736/.

[310] N. Inkawhich, M. J. Inkawhich, E. K. Davis, *et al.*, "Bridging a Gap in SAR-ATR: Training on Fully Synthetic and Testing on Measured Data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2942–2955, 2021, ISSN: 21511535. DOI: 10.1109/JSTARS.2021.3059991.

[311] T. D. Ross, S. W. Worrell, V. J. Velten, J. C. Mossing, and M. L. Bryant, "Standard sar atr evaluation experiments using the mstar public release data set," in *Algorithms for Synthetic Aperture Radar Imagery V*, SPIE, vol. 3370, 1998, pp. 566–573.

[312] M. S. Seyfioglu, B. Erol, S. Z. Gurbuz, and M. G. Amin, "DNN Transfer Learning from Diversified Micro-Doppler for Motion Classification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 5, pp. 2164–2180, 2019, ISSN: 15579603. DOI: 10.1109/TAES.2018.2883847.

[313] D. Masters and C. Luschi, "Revisiting Small Batch Training for Deep Neural Networks," pp. 1–18, Apr. 2018. [Online]. Available: http://arxiv.org/abs/1804.07612.

Appendix A

Appendix

A.1 CNN Hyper-parameters

The tables list and describe the hyperparameters of the model, the training procedure and data processing used for the training of all models in the thesis, unless stated otherwise in the text. Not all hyperparameters were optimised for each experiment, but were experimented with for best classification performance by hand prior to the secondary, more focused parameter optimisation within each experiment and published result.

Feature	Examples	Effect
Kernel Sizes	(5,5)	Maximum of feature learned
Number of Channels	(In: 3, Out: 16)	Volume of channels of feature multiplication
Grouped Convolutions	(1,2,1,2,2) for AlexNet	Parallel convolutional processing streams
Strides	1, 3, 5	Feature undersampling across sample input dimension
Max-Pooling	(2,2) with (3,3) stride	Undersampling of features within each convolution output
Normalisation	Layer [169], Batch [170], Channel [171]	For dataset variation, runtime, trainability considerations respectively
Linear Layer Width	4800	For feature processing, depending on dataset size and variation
Linear Layer Depth	2-15	For feature processing, depending on dataset size and variation
Input Size	(277,277)	Data Resolution vs Size and runtime of model in memory

Table A.1: Typical CNN Hyper-Parameters.

Parameter	Examples	Effect
Loss Function	MSE or BCE Loss	How target differentiation will be calculated and backpropagated
Epochs	16 to 500	Number of views of dataset required before overfitting
Batch Size	2^n [313]	Number of samples averaged together in each training
Optimising Algorithm	SGDm or AdaM	Backpropagation gradient decent measurement determination
Learning Rate	$1e^{-4}$	Strength and pace of learning, can be decayed over time
Activation Function	Sigmoid or ReLU	Non-linear component to learn and combine abstract features
Regularisation	L1 or L2	Clipping of the learning gradient components calculated
Dropout	25%	Zeros a random crop of inputs each batch to encourage training set invariance

Table A.2: Typical Training Hyper-Parameters.

Parameter	Examples	Effect
CPI Duration	4096	Trade-off of return dynamics and resolution
CPIs per sample	1,5,20	Trade-off of latency and information present
Doppler bins displayed	300	Crop out the high frequencies to the maximum bandwidth of target returns
Clutter Suppression	High-pass Filter	Increase dynamic range of signatures with high clutter

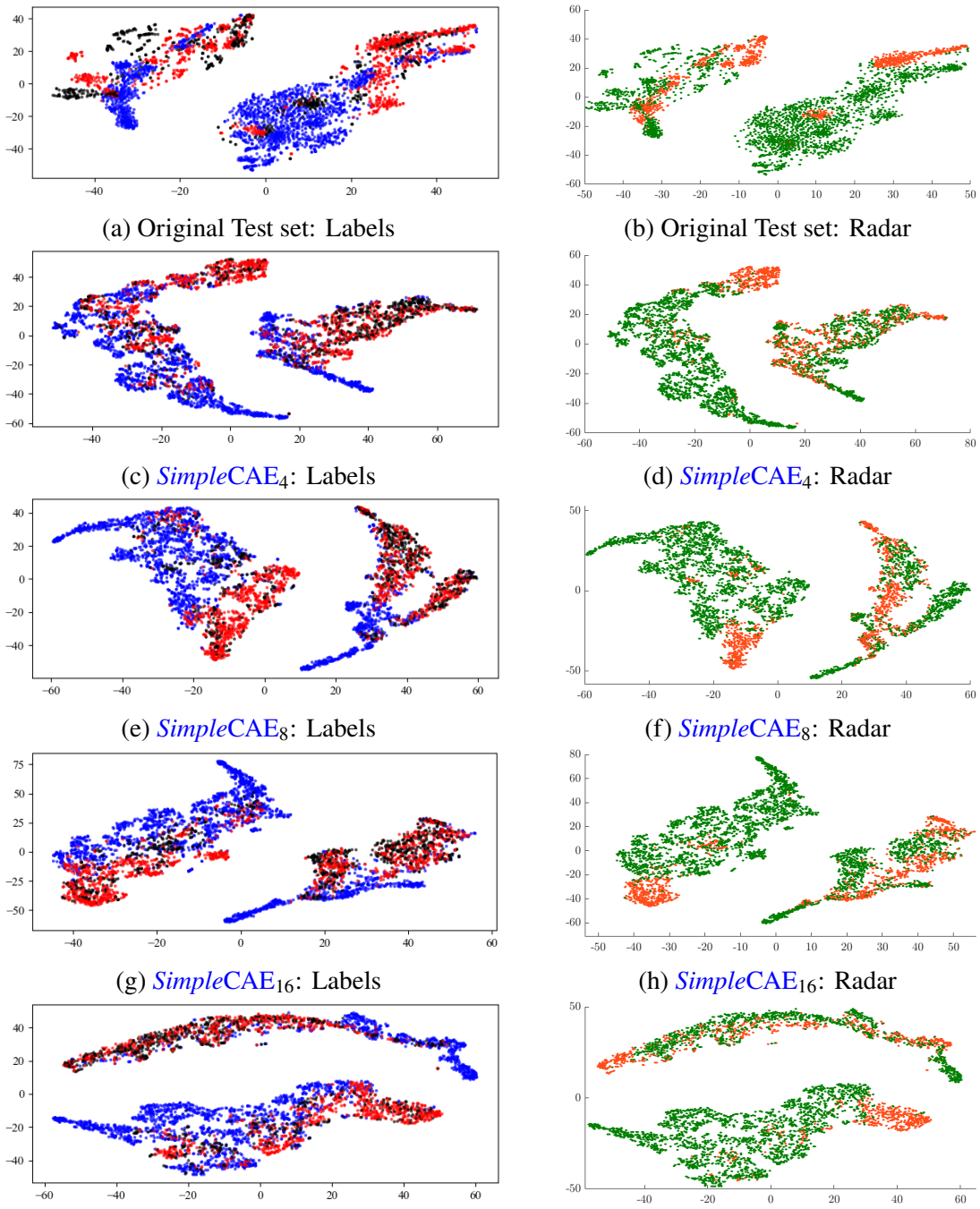
Table A.3: Spectrogram DSP options.

A.2 ADB Latent Representations with 4-32 Latent Variables

Figure A.1 shows the labelled latent scatter plots using tSNE with perplexity=50 for the same *Simple*CAE but with latent dimensionality of 4, 8, 16 and 32. This figure includes at the top the tSNE decomposition of the entire LDB test dataset. There is not a great deal of interpretable difference between any of these latent scatter plots. All decompositions, including the original dataset, both show two main clusters - with one of the clusters featuring a clear region where Radar#2 returns exist at the edge of the cluster. This cluster (on the right in Fig. A.1c/A.1d, on the left in A.1e-A.1g and below in A.1i/A.1j) in Fig. A.1b of the full dataset decomposition has a clump in the middle of that belongs to Radar#2. For autoencoder versions of this observation, the minor Radar#2 cluster is not as localised in the middle of a mass of Radar#1 returns. To differing degrees in the latent representations this sub-cluster has been spread out into the structure of the latent representation. However, overall the increasing of latent dimensionality from 4 to 32 did not have as large an impact on the representation compared to the difference between 2 and 4.

Figure A.2 and A.3 show the reconstructions of bird and drone samples. Each model from 2 to 32 latent variables learned to reconstruct unseen test images, but it can be seen that at 32 latent dimensions, row 1 and 4 of Fig. A.2d, the quality of the reconstruction is much improved where previously they were with a black background with feint signatures barely visible. These examples are of complex returns of drone with phase noise induced self-clutter lines repeated in Doppler that feature more frequently in Radar#2 signatures, suggesting possibly that larger latent dimensions can better combine features from different sources in the spectrograms. Figure A.4 and A.5 shows the decoder mapping for each of latent axes for the models with 4, 8, 16 and 32 latent variables. We can see that radar many samples are a plausible return - all showing the effect of the low pass filter, and show a different type of target return. Some latent axis show clear μDoppler returns, such as in the negative of the 4th latent dimension in Fig. A.2b for 8 latent dimensions. However on the positive side of this, the returns are highly dissimilar to any possible radar return. The latent axes sampling of 32 dimensions in Figure A.5 takes the entire page, and throughout, many axes possess realistic samples and a reasonable gradation of

features as the latent space is traversed away from the origin.



(a) Original Test set: Labels

(b) Original Test set: Radar

(c) *Simple*CAE$_4$: Labels

(d) *Simple*CAE$_4$: Radar

(e) *Simple*CAE$_8$: Labels

(f) *Simple*CAE$_8$: Radar

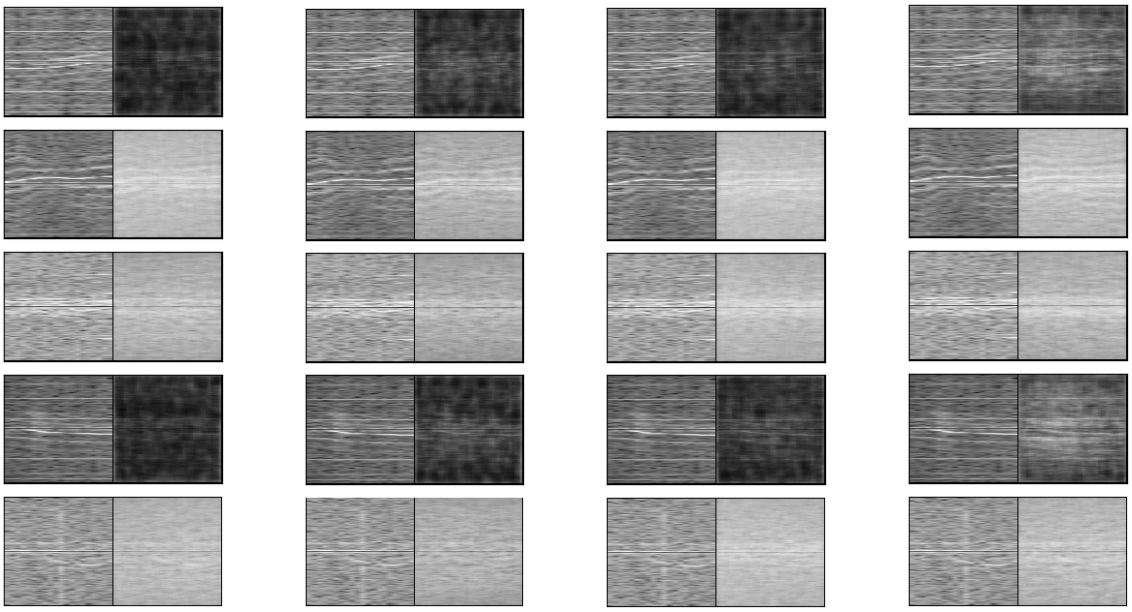
(g) *Simple*CAE$_{16}$: Labels

(h) *Simple*CAE$_{16}$: Radar

(i) *Simple*CAE$_{32}$: Labels

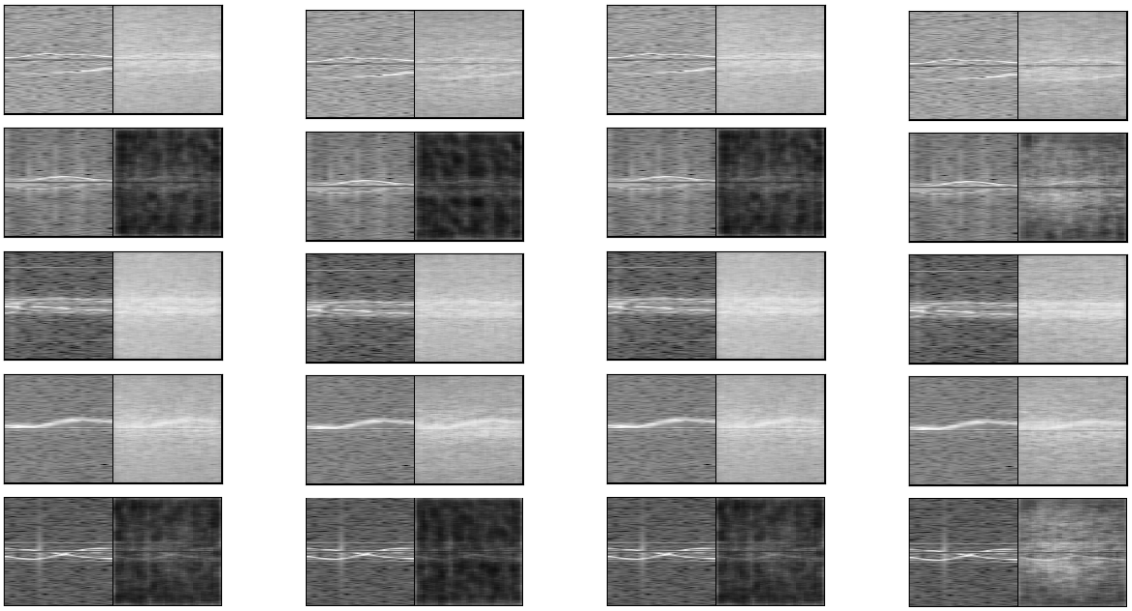
(j) *Simple*CAE$_{32}$: Radar

Figure A.1: Latent variable visualisations of LDB using tSNE. Target labelled latent scatter plots on the left and radar origin on the right.



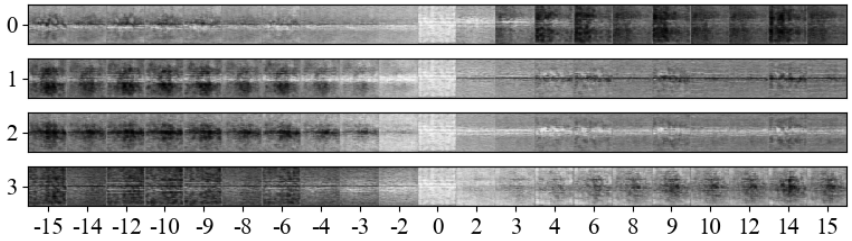
(a) 4 Latent Variables (b) 8 Latent Variables (c) 16 Latent Variables (d) 32 Latent Variables

Figure A.2: Reconstructions of Drones for 4 to 32 Latent Variables.

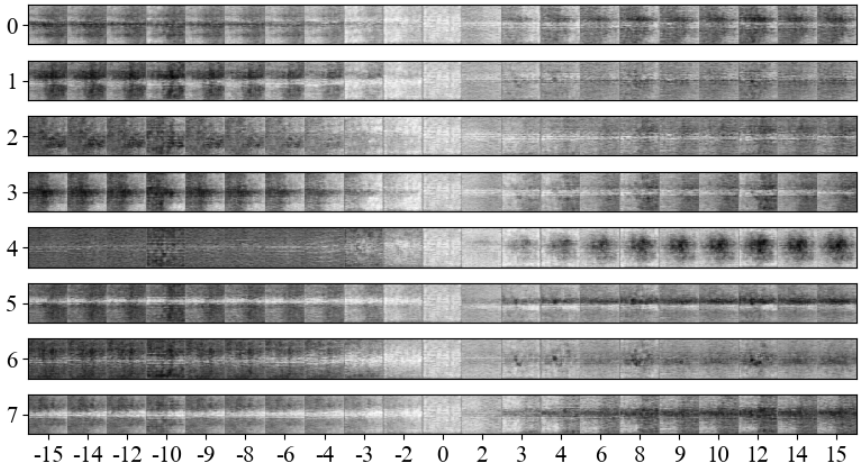


(a) 4 Latent Variables (b) 8 Latent Variables (c) 16 Latent Variables (d) 32 Latent Variables

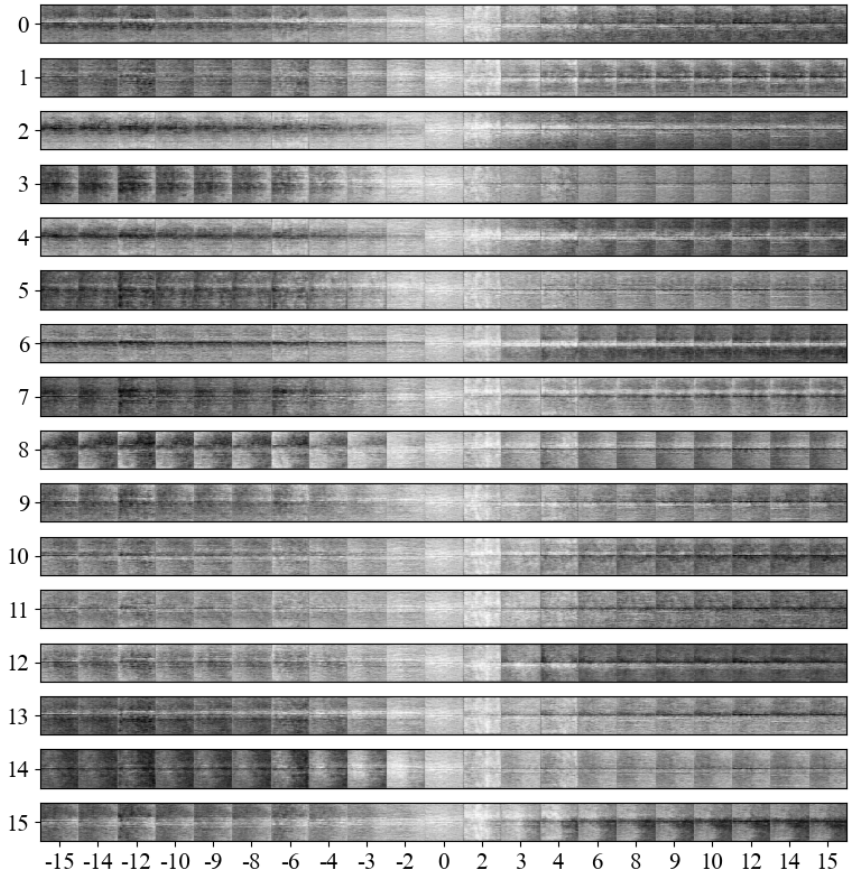
Figure A.3: Reconstructions of Birds for 4 to 32 Latent Variables.



(a) 4 Latent Variables

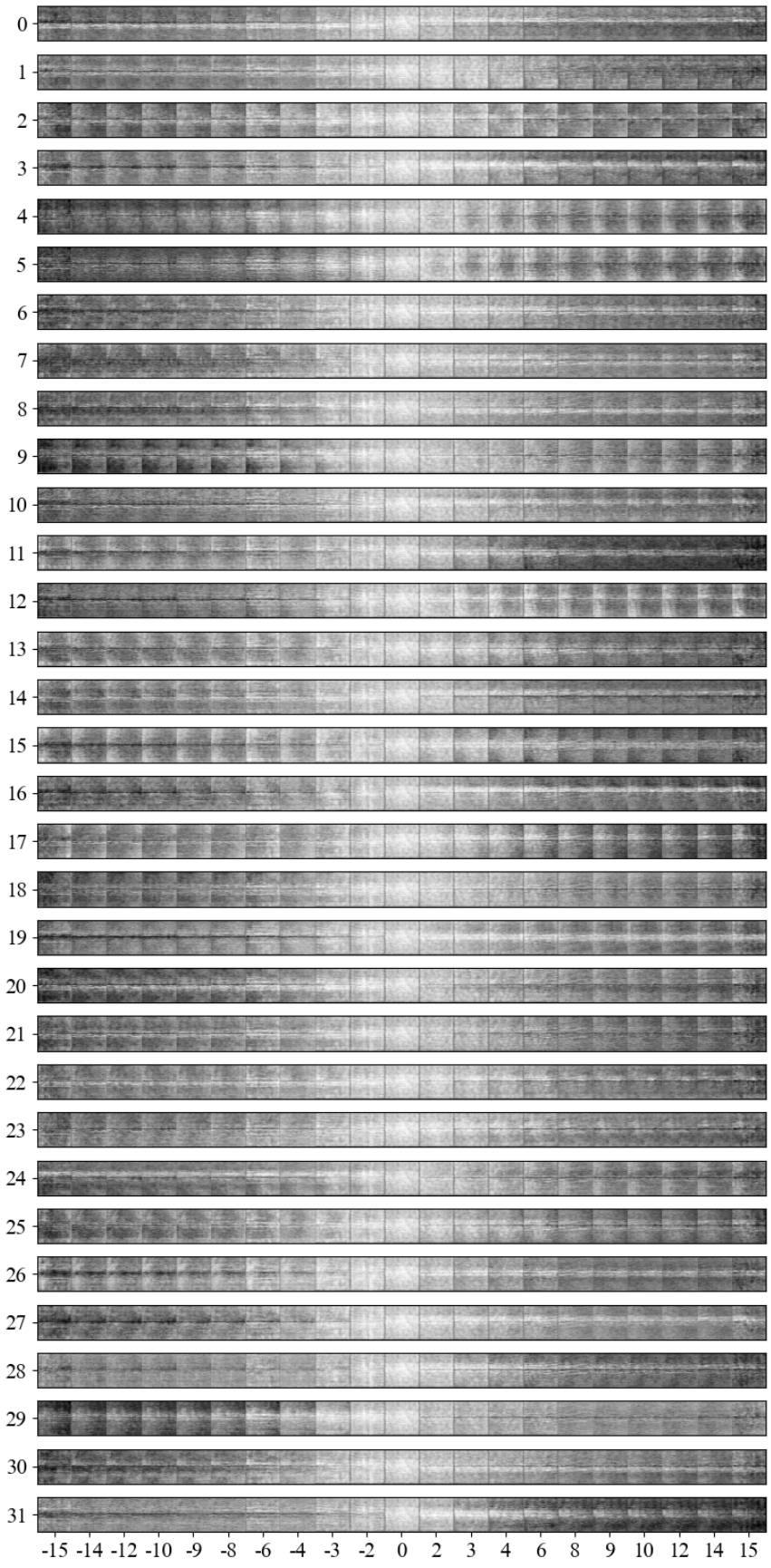


(b) 8 Latent Variables



(c) 16 Latent Variables

Figure A.4: Latent Axis Decoder Sampling of 4, 8 and 16 Latent Variables.

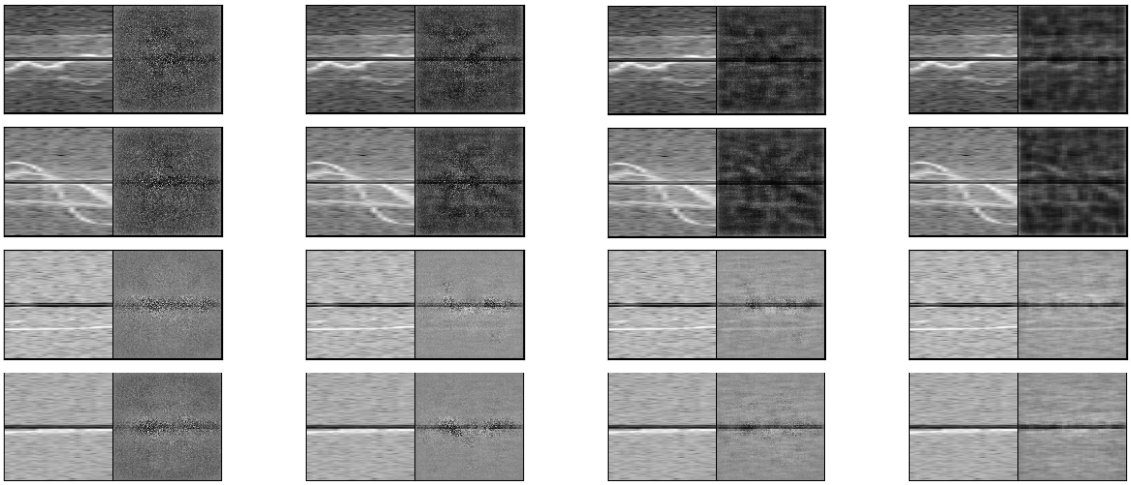


(a)

Figure A.5: Latent Axis Decoder Sampling of 32 Latent Variables.

A.3 CAE Decoder Performance on ADB Dataset

Figure A.6 contains the reconstructions of 4 samples for birds and drones from the LDB dataset with increasing training epochs, using the *Simple*CAE$_8$.

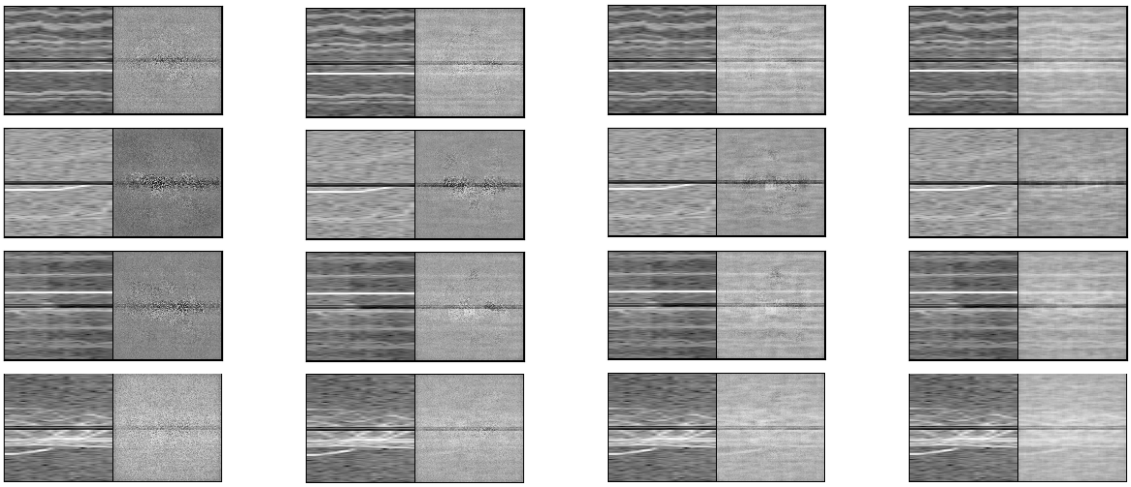


(a) Birds at 1 epoch

(b) Birds at 3 epochs

(c) Birds at 9 epochs

(d) Birds at 27 epochs



(e) Drones at 1 epoch

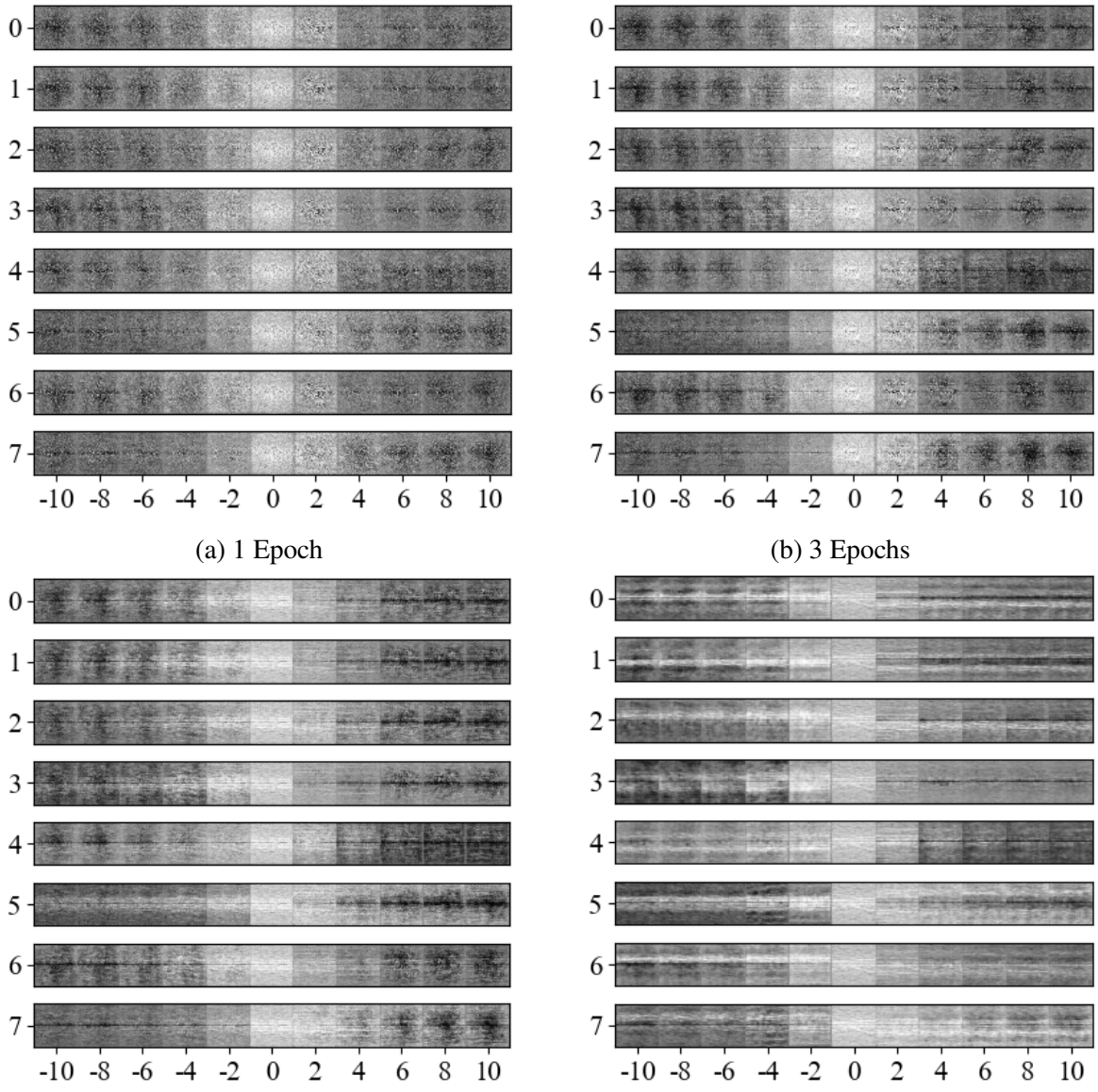
(f) Drones at 3 epochs

(g) Drones at 9 epochs

(h) Birds at 27 epochs

Figure A.6: ADB test set reconstructions against training epoch.

Figure A.6 shows the latent axis sampling of the decoder trained on the LDB dataset with increasing training epochs, also using the *Simple*CAE$_8$.



(a) 1 Epoch

(b) 3 Epochs

(c) 9 Epochs

(d) 27 Epochs

Figure A.7: Decoder sampling of each latent axis against training epoch.