DEEP LEARNING FOR RADAR CLASSIFICATION OF DRONES

by

Holly Dale

A thesis submitted to the University of Birmingham for the degree of DOCTOR OF PHILOSOPHY

Department of Electronic, Electrical and Systems Engineering
School of Engineering
College of Engineering and Physical Sciences
University of Birmingham
September 2022

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This unpublished thesis/dissertation is copyright of the author and/or third parties.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.

Abstract

Proliferation in drone numbers requires the development of a surveillance system for monitoring low-altitude airspace occupancy and ensuring safety in the skies. Radar offers a method of achieving 24-hour surveillance in all weather conditions, making it an appropriate solution for this challenge. In order to be effective, a radar solution needs to be able to discriminate between birds and drones for the removal of false alarms.

This thesis investigates the use of convolutional neural networks (CNNs) for drone classification. A CNN is trained for drone-bird discrimination on radar spectrograms obtained using an L-band staring radar, and performance is assessed and compared with machine learning benchmarks. The CNN is validated on an extensive data set, and the classifier's ability to generalise against new models of drone and unseen clutter environments is investigated. Dynamic classifier selection is used to make an intelligent choice of CNN classifier based on features extracted from the tracker and the spectrogram, leading to a reduction in false positives whilst maintaining a high true positive rate at low signal to noise ratio (SNR).

It is desirable to classify drones at longer ranges (therefore with lower SNR) to maximise the time available to deploy drone countermeasures. CNN classification performance is therefore investigated as a function of SNR, firstly through the addition of Gaussian noise to an experimentally obtained dataset of radar spectrograms, and secondly through data collected in increasingly demanding environments. Whilst classification accuracy falls with decreasing SNR, the augmentation of training data is shown to enhance performance at low SNR by 14%. Bayesian optimisation is used to maximise performance through the optimisation of augmentation hyperparameters, leading to an improvement in performance of more than 10%.

Finally, CNNs are used to perform drone vs drone classification. Performance is established for drone group (fixed wing vs rotary wing, F1 score 0.97), drone sub-

group (F1 score 0.95) and drone model (F1 score 0.67) classification using CNNs, across 9 different drone models measured at Ku-band. Multiple classification strategies are considered (single stage and multi-stage), and the relationship between dwell-time and performance is explored, forming a comprehensive investigation of CNN use for drone-to-drone classification. The feature space of drone classification is investigated, and decision tree classifiers are used to explore the relationship between processing length, coherency, and performance. Training parallel classifiers on various processing lengths and fusing the results leads to an improvement in performance of 26%.

Dedication

For Beth, For her support, encouragement, and endless patience.

Acknowledgements

I would like to give my warmest thanks to everybody who has supported me throughout my research.

I'm extremely grateful to my supervisors, Dr. Mike Antoniou and Prof. Chris Baker for your indispensable support and guidance throughout each stage of my research. Thank you to Dr. Mohammed Jahangir for your invaluable insights.

Thank you to the Engineering and Physical Sciences Research Council (EPSRC) for the studentship which has allowed me to conduct this research.

Many thanks to Dr. Stephen Harman, Dr. Aled Catherall, and all at Aveillant and PlexTek for provisioning the data and making this work possible.

Thank you to all of my fellow students in the Microwave Integrated Systems Laboratory (MISL).

I'd like to acknowledge my colleagues in the Data Lab for your encouragement, friendship, and snacks.

I would like to express my deepest gratitude to my fantastic family: the Dales, Smiths, and Baxters. Particular thanks to my Mum, Dad, and sister Anna for your unwavering love and support, for teaching me the importance of hard work, and for always believing in me. Thank you sweet Evie for making me smile, always.

Beth, if I attempted to list the reasons that I was grateful for you, I would reach the word limit before I'd even began. Thank you. I will now tidy my desk.

Contents

List of Figures **ix**

List of Tables **xxvi**

List of Abbreviations **xxx**

1 Introduction **1**

1.1 Background 1

1.1.1 Motivation 4

1.2 Aims and Objectives 5

1.2.1 Contributions 6

1.2.2 Publications 9

1.3 Thesis Outline 10

2 Foundations of Radar **12**

2.1 Introduction 12

2.2 Radar Fundamentals 13

2.2.1 Measuring Range 13

2.2.2 The Radar Range Equation 15

2.2.3 Azimuth and Elevation 18

2.3 Doppler and Micro-Doppler 20

2.3.1 The Doppler Effect 20

2.3.2 The Micro-Doppler Effect 22

2.4 Fourier Analysis . . . 25
2.5 Frequency Modulated Continuous Wave Radar . . . 27
2.6 Staring Radar . . . 30
2.7 Spectrogram Formation . . . 32

3 Machine Learning Foundations **36**
3.1 Introduction . . . 36
3.2 Machine Learning Concepts . . . 36
3.2.1 Sources of Error, Bias, and Variance . . . 37
3.3 Machine Learning Algorithms . . . 38
3.3.1 Naïve Bayes . . . 38
3.3.2 Discriminant Analysis . . . 39
3.3.3 Support Vector Machine . . . 40
3.3.4 k-Nearest Neighbours . . . 41
3.3.5 Decision Tree . . . 42
3.3.6 Random Forest . . . 42
3.4 Deep Learning Concepts . . . 43
3.4.1 Artificial Neural Networks . . . 44
3.4.2 Training a Neural Network . . . 46
3.4.3 Convolutional Neural Networks . . . 49
3.5 Performance Evaluation . . . 55

4 Radar-Based Drone Classification Review **60**
4.1 Introduction . . . 60
4.2 Rule-based Approaches . . . 61
4.3 Machine Learning Approaches to Drone vs Non-Drone Classification . . . 63
4.4 Machine Learning Approaches to Drone vs Drone Classification . . . 66
4.5 Deep Learning Approaches to Drone vs Non-Drone Classification . . . 68
4.6 Deep Learning Approaches to Drone vs Drone Classification . . . 71

4.7 Summary 73

5 Establishing CNNs For Drone Classification 79

5.1 Introduction 79

5.2 Data Collection 81

5.2.1 Ground Truth 82

5.2.2 Data Set A 84

5.2.3 Bird and Drone Spectrograms 87

5.3 Decision Tree Benchmark Performance 89

5.3.1 Classification Features 89

5.3.2 Decision Tree Performance 92

5.4 Convolutional Neural Network Performance 99

5.4.1 Comparison With Decision Tree Benchmark 101

5.4.2 Other Machine Learning Algorithms 102

5.5 Low SNR classification 104

5.5.1 Methodology 104

5.5.2 Testing on Low SNR Data 107

5.5.3 Network Architecture 109

5.5.4 Training and Testing on Similar SNR Data 112

5.5.5 Augmentation for Robust Classification 113

5.6 Chapter Conclusions and Future Work 116

5.6.1 Conclusion 116

5.6.2 Recommendations 117

5.6.3 Future Work 118

6 Drone vs Drone Classification 120

6.1 Introduction 120

6.2 Methodology 123

6.2.1 System and Processing Parameters 123

6.2.2 SNR Definition . . . 124
6.2.3 Data Coverage . . . 126
6.2.4 Example Spectrograms . . . 127
6.2.5 Classification Approach . . . 129
6.3 Drone Group Classification . . . 133
6.3.1 Baseline Case . . . 133
6.3.2 Low SNR and All Data Cases . . . 136
6.4 Drone Subgroup Classification . . . 139
6.4.1 Classification Based on Number of Rotors . . . 139
6.4.2 Classification Based on Size . . . 142
6.5 Drone Model Classification . . . 145
6.5.1 Single-stage Classifier . . . 146
6.5.2 Multi-stage Classification . . . 153
6.6 Effect of Processing Interval on Drone Classification Performance . . . 157
6.6.1 Feature Extraction . . . 158
6.6.2 Feature Analysis . . . 159
6.6.3 Classification Performance as a Function of Integration Length 163
6.7 Chapter Conclusions and Future Work . . . 167
6.7.1 Conclusion . . . 167
6.7.2 Recommendations . . . 168
6.7.3 Future Work . . . 169

7 Classification Under Real and Dynamic Conditions 171

7.1 Introduction . . . 171
7.2 Methodology . . . 174
7.2.1 Spectrogram Parameters . . . 174
7.2.2 Data Set B Coverage . . . 176
7.2.3 Example Spectrograms . . . 179
7.2.4 Implementation Details . . . 182

7.3 Control Classifier Results . . . 182
7.3.1 Training the Control Classifier . . . 182
7.3.2 Testing the Control Classifier on Data Set A . . . 183
7.3.3 Testing the Control Classifier on Data Set B . . . 188
7.4 Retrained Classifier Results . . . 196
7.4.1 Training the Retrained Classifier . . . 196
7.4.2 Testing the Retrained Classifier on Data Set A . . . 197
7.4.3 Testing the Retrained Classifier on Data Set B . . . 198
7.4.4 Evaluating the False Alarm Rate of the CNN Classifiers . . . 202
7.5 Fused Classifier Approach to Reducing the False Alarm Rate . . . 204
7.5.1 Fused Classifier Methodology . . . 206
7.5.2 Decision Tree for Classifier Fusion . . . 209
7.5.3 Dynamic Classifier Selection . . . 213
7.6 Drone-vs-Drone Classification . . . 214
7.7 Chapter Conclusions and Future Work . . . 219
7.7.1 Conclusion . . . 219
7.7.2 Recommendations . . . 221
7.7.3 Future Work . . . 221

8 Conclusion and Future Work **224**
8.1 Summary . . . 224
8.2 Future Work . . . 228
8.3 Conclusion . . . 230

A Drone Classification Per-file Confusion Matrices **231**

B Drone Classification F1 Box Plots **239**

C Drone Classification Performance Summary **244**

Bibliography **246**

List of Figures

1.1 Relative interest over time for the search term "C-UAS" (counter unmanned aerial system) on Google [1]. 3

2.1 The radiation pattern of an antenna consisting of a linear array of elements, as a function of angle from the boresight, Theta [2]. 19

2.2 A diagram showing the spectrogram of a single rotor drone with two blades [3]. 23

2.3 A simulated drone spectrogram where the rotor blade accelerates and the Fourier Transform window is fixed. As the rotation speed increases, helicopter rotor modulation (HERM) lines appear [4]. 24

2.4 The discrete-time Fourier transform (DTFT) of a rectangular pulse a) With no window function b) Using a Hamming window function [5]. 26

2.5 Frequency as a function of time for a transmitted wave with a sawtooth modulation pattern. 28

2.6 Transmitted and received waves when Frequency Modulated Continuous Wave (FMCW) radar is used to detect a moving target, a) Sawtooth modulation b) Triangular modulation. 29

2.7 A diagram showing the transmit (red) and receive (blue) beams of a staring radar [6]. 31

2.8 A diagram showing the preprocessing steps of the L-band pulsed staring radar. A 4-D data matrix is formed, from which targets are detected and tracked. The tracker output is then used to identify the resolution cell containing a particular target at each time step, and a time series vector is extracted and used to plot a spectrogram. 33

2.9 A diagram showing the process of producing a spectrogram from time series data. 33

2.10 A diagram illustrating the concept of overlap in spectrogram generation. 34

2.11 An example drone spectrogram produced using a 4096-point Fast Fourier Transform, with 50% overlap between frames and a Blackman-Harris window applied. 35

3.1 A training data set for a two class problem (red and blue), with features A and B sampled from normal distributions. 39

3.2 A training data set for a two class problem (red, blue), with two feature vectors, showing the linear (left) and quadratic (right) boundary between the classes when a discriminant analysis model is used. . . . 40

3.3 A training data set for a two class problem (red, blue), with two feature vectors, showing the hyperplane that maximises the margin between the two classes. 40

3.4 A training data set for a three class problem (red, blue, and magenta), with a test sample (black '+') that would be assigned to the blue class using a k-nearest neighbours model with $k = 3$. 41

3.5 An example decision tree trained to classify a target into category 'red' or 'blue', using two features x1 and x2 a) Training data feature space b) Trained decision tree. 43

3.6 An example of a neuron with three inputs x_i, weights w_i, bias b, and an output that is determined by the activation function σ. 45

3.7 Sigmoid, tanh and rectified linear unit (ReLU) activation functions. . 46

3.8 The typical structure of a convolutional neural network with a single convolutional layer. 51

3.9 A convolution between an image and a learned filter. The dot product between the sampled image and the kernel produces a single element within the feature map [7]. 51

3.10 AlexNet architecture [8]. 52

3.11 SqueezeNet fire module [9]. 53

3.12 Inception module [10]. 54

3.13 Residual block [11]. 54

3.14 An annotated confusion matrix. Recall and precision are calculated using the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) metrics. 57

3.15 Example receiver operating characteristic (ROC) curves for a) A random classifier b) A good classifier c) A perfect classifier. 58

5.1 The GPS track (white) of an I1-D drone flight in Google Earth [12], overlaid onto the radar field of view (purple). The radar position is denoted as GK. 82

5.2 An opportune bird target photograph overlaid with metadata recorded using Theodolite [13]. 83

5.3 Radar tracks in Google Earth [12] a) All tracks recorded during a single trial (purple) b) A single track (purple) and vectors along which field observations were recorded (white). 84

5.4 a) Deenethorpe (UK) site map, b) Antwerp (Belgium) site map. Displayed in Google Earth [12]. 84

5.5 Images showing a) I1-D, b) I2-D. 85

5.6 Box plots showing the coverage of data set A in a) Range, b) Height relative to the radar. 86

5.7 Example spectrogram showing the Doppler profile of a) an I1-D rotary wing drone b) a Bird c) an H-D rotary wing drone. 88

5.8 A cluster plot showing the number of Doppler components, velocity, and echo amplitudes of the bird and drone data frames. 90

5.9 A cluster plot showing the velocity, acceleration, and jolt of the bird and drone data frames. 91

5.10 A cluster plot showing the velocity, height, and amplitude of the bird, drone, car and aircraft data frames 92

5.11 Bar chart showing the ranking of decision tree predictor importance. . 93

5.12 Confusion matrix displaying results of a decision tree classifier tested on bird and drone data frames. 93

5.13 Receiver operating characteristic (ROC) curve displaying the relationship between the true positive rate and false positive rate of a decision tree classifier tested on bird and drone data frames. 94

5.14 A confusion matrix displaying the results of a decision tree classifier tested on bird, drone, car, and aircraft data frames. 94

5.15 The accuracy of a decision tree classifier where X% of frames are corrupted, resulting in zero Doppler components measured, emulating performance on low signal to noise ratio test data. 95

5.16 A confusion matrix displaying the results of a decision tree classifier tested on bird and drone data frames, where 60% of the non-zero number of Doppler component values were set to zero, to emulate performance on low signal to noise ratio test data. 96

5.17 A bar chart showing the ranking of decision tree predictor importance, where the number of Doppler components feature has been corrupted prior to training. 97

5.18 A confusion matrix displaying the results of a decision tree classifier tested on bird and drone data frames, where the micro-Doppler feature was not used. 98

5.19 A bar chart showing the ranking of decision tree predictor importance, where the micro-Doppler feature was not used. 98

5.20 A confusion matrix displaying the results of a convolutional neural network (CNN) classifier tested on bird and drone spectrograms. . . . 99

5.21 A ROC curve displaying the results of a convolutional neural network (CNN) classifier tested on bird, drone, car, and aircraft spectrograms. 100

5.22 A confusion matrix displaying the results of a convolutional neural network (CNN) classifier tested on bird, drone, car, and aircraft spectrograms. 100

5.23 A confusion matrix displaying the results of a decision tree classifier tested on bird and drone frames, using a majority-rules approach across 40 data frames. 101

5.24 A confusion matrix displaying the results of a decision tree classifier tested on bird, drone, car, and aircraft frames, using a majority-rules approach across 40 data frames. 102

5.25 Doppler profile showing the power, target return, and background bins used for calculating the signal to noise ratio. 105

5.26 Target power, background power, and signal to noise ratio (SNR) over a drone track. 106

5.27 Drone recall (orange), bird recall (blue), and accuracy (grey) of a convolutional neural network (CNN) trained on high signal to noise ratio (SNR) data, as the SNR of the test data is decreased in increments of 3 dB [7]. 107

5.28 Maximum activations in the first convolutional layer when testing a convolutional neural network (CNN) on a drone spectrogram, where white indicates areas of high activation. (a) 46 dB signal to noise ratio (SNR), (b) 40 dB SNR, (c) 34 dB SNR, (d) 26 dB SNR and (e) 20 dB SNR [7]. 108

5.29 Accuracy against prediction time for the pretrained image classifiers available in MATLAB, taken from [14]. The area of each marker represents network size. 110

5.30 Classification accuracy as a function of mean test data signal to noise ratio (SNR) for six network architectures [7]. 111

5.31 Classification performance when training and testing AlexNet on data of a similar signal to noise ratio (SNR) [7]. 113

5.32 Convolutional neural network (CNN) classification performance showing drone recall (orange), bird recall (blue) and accuracy (grey) as a function of test signal to noise ratio (SNR), when each SNR bin is equally represented in the training data [7]. 114

5.33 Convolutional neural network (CNN) classification accuracy showing drone recall (orange), bird recall (blue) and accuracy (grey) as a function of test signal to noise ratio (SNR), when the hyperparameters for training data augmentation have been found using Bayesian optimisation [7]. 116

6.1 Example DJI Matrice 210 spectrograms, with a Fast Fourier Transform (FFT) length ranging from 256 pulses to 4096 pulses. 125

6.2 Example drone spectrograms a) High signal to noise ratio (SNR) DJI Matrice M210 b) High SNR FMS Easy Trainer c) High SNR Yuneec H920 d) Low SNR DJI Matrice M210 e) Low SNR FMS Easy Trainer f) Low SNR Yuneec H920. 128

6.3 The number of spectrograms recorded with a signal to noise ratio (SNR) above and below the baseline threshold of 30 dB, for each model of drone. 131

6.4 Drone group classification, tested on the baseline data set a) Confusion matrix, b) F1 scores, c) Classification outcomes per drone model. 134

6.5 Drone group classification, tested on the baseline data set, where classification has been performed on a per-file basis a) Confusion matrix, b) F1 scores. 135

6.6 Drone group classification receiver operating characteristic curves, tested on the baseline data set a) Fixed wing drone group, b) Rotary wing drone group. 136

6.7 Confusion matrices displaying the result of drone group classification a) Tested on the low signal to noise ratio (SNR) data set b) Trained and tested on the full data set. 137

6.8 Drone group classification tested on the full test set a) signal to noise ratio (SNR) of correctly and incorrectly classified targets b) F1 score as a function of number of frames for drone group classification. Note that the peak corresponds to the mean track length of this data set. . 138

6.9 Drone subgroup classification, tested on the baseline data set a) Confusion matrix, b) Classification outcomes for each model of drone. . . 140

6.10 Drone subgroup classification receiver operating characteristic (ROC) curves, tested on the baseline data set. 141

6.11 Drone rotor subgroup classification results a) Tested on the low signal to noise ratio (SNR) test data set b) Tested on the full test set. . . . 142

6.12 Drone size subgroup classification results a) Baseline case b) Low signal to noise ratio (SNR) case c) All data case. 143

6.13 Feature plot showing the drone body signal to background ratio, strongest helicopter rotor modulation (HERM) line power to background ratio, and the maximum number of peaks in each spectrogram a) Rotor-based subgroups, b) Size-based subgroups. 144

6.14 Drone size subgroup classification receiver operating characteristic (ROC) curves, tested on the baseline data set. 145

6.15 Single-stage drone model classification confusion matrix, tested on the baseline data set. 147

6.16 Drone model classification receiver operating characteristic (ROC) curves, tested on the baseline data set. 147

6.17 Single-stage drone model classification results where the DJI Mavic Pro and the Ares Gamma Pro V2 classes have been a) Combined with the classes they are most commonly mistaken for, b) Removed. 149

6.18 Single-stage drone model classification results, tested on the low signal to noise ratio (SNR) data set. 150

6.19 Single-stage drone model classification results, tested on the full test data set a) Confusion matrix, b) Mean F1 score vs target signal to noise ratio (SNR). 151

6.20 Drone model classification F1 score as a function of the number of frames, tested on the full test data set. Note that the peak corresponds to the mean track length of this data set. 152

6.21 A feature plot showing the body signal to background ratio, strongest helicopter rotor modulation (HERM) line power to background ratio, and the maximum number of peaks in each spectrogram a) Drone groups, b) Drone models. 153

6.22 The classification hierarchies tested for drone model classification. . . 154

6.23 2-stage drone model classification confusion matrices: a) Group classification followed by model classification, b) Subgroup classification followed by model classification. 155

6.24 3-stage drone model classification confusion matrix. 156

6.25 a) Peaks extracted from a single spectra, after taking the signal to background ratio (SBR) b) Corresponding DJI Mavic Air 2 spectrogram. 159

6.26 a) Gain in signal to background ratio with increasing integration length, b) Gain in helicopter rotor modulation (HERM) line to background ratio with increasing integration length. 160

6.27 a) Spread in the helicopter rotor modulation (HERM) line to background ratio (HBR), for each drone model as the processing length is increased b) Spread in the mean spacing between Doppler peaks for each drone model as the processing length is increased. 161

6.28 Feature plot showing the body signal to background ratio, strongest helicopter rotor modulation (HERM) line power to background ratio, and the mean spacing between Doppler peaks in each spectrogram a) Integrating over 128 pulses, b) Integrating over 512 pulses, c) Integrating over 2048 pulses. 162

6.29 F1 score as a function of processing length, in terms of number of samples a) Drone group classification b) Drone subgroup classification (size-based), c) Drone model classification. 164

6.30 F1 score where six decision tree classifiers trained on features derived using various processing lengths are used, and the average score over all six classifiers is used to output the final label a) Drone group classification b) Drone subgroup classification (size-based), c) Drone model classification. 165

6.31 F1 score where six decision tree classifiers trained on features derived using various processing lengths are used, and a majority rules approach is used to output the final label a) Drone group classification b) Drone subgroup classification (size-based), c) Drone model classification. 166

7.1 Example I1-D spectrograms, each 20 frames long a) The first spectrogram in a track b) The second spectrogram in a track where a 20 frame update rate is used c) The second spectrogram in a track where a single frame update rate is used. 175

7.2 Coverage of the bird data in data set B in a) Range, b) Height relative to the radar. 179

7.3 Example spectrograms a) Z1-F (946 m) b) S9-D (753 m) c) I1-D (553 m) d) P1-D (2.9 km) e) I1-D (3.3 km) f) Opportune bird spectrogram. 181

7.4 Performance on data set A as a function of spectrogram length. . . . 183

7.5 Performance of the unweighted control classifier on data set A, using a 20-frame update rate. 184

7.6 Input spectrograms and their corresponding maximum activations a) Correctly classified drone spectrogram b) Misclassified drone spectrogram c) Correctly classified bird spectrogram d) Misclassified bird spectrogram. 186

7.7 Performance of the control classifier on data set A, using a single frame update rate a) Unweighted loss function b) Weighted loss function. . 187

7.8 Performance of the unweighted control classifier on test samples from data set A and data set B, using a 20-frame update rate. 189

7.9 Performance results for the control classifier on each drone model in data set B (recall 42.8%). 190

7.10 Control classifier performance as a function of range for the a) Small drones in data set B b) Medium drones in data set B c) Large drones in data set B. 192

7.11 Control classifier performance as a function of signal to background ratio (SBR) for the a) Small drones in data set B b) Medium drones in data set B c) Large drones in data set B. 194

7.12 Control classifier scores a) The spectrogram is a correctly classified 'Bird' spectrogram b) The spectrogram is a 'Bird' spectrogram misclassified as 'Drone' c) The spectrogram is a correctly classified 'Drone' spectrogram d) The spectrogram is a 'Drone' spectrogram misclassified as 'Bird'. 196

7.13 Performance of the retrained classifier on data set A, and data set B a) Balanced bird and drone classes b) Unbalanced bird and drone classes. 198

7.14 Retrained classifier performance as a function of signal to background ratio (SBR) for the a) Small drones in data set B b) Medium drones in data set B c) Large drones in data set B. 200

7.15 Retrained classifier scores a) The spectrogram is a correctly classified 'Bird' spectrogram b) The spectrogram is a 'Bird' spectrogram misclassified as 'Drone' c) The spectrogram is a correctly classified 'Drone' spectrogram d) The spectrogram is a 'Drone' spectrogram misclassified as 'Bird'. 201

7.16 Histogram showing the a) False alarm rate and b) The number of bird spectrograms misclassified per minute by the control classifier (red) and retrained classifier (blue). 202

7.17 a) The mean false alarm rate of the retrained classifier, as a function of the number of spectrograms the output is integrated over b) The mean number of false alarms per minute of the retrained classifier, as a function of the number of spectrograms the output is integrated over. 204

7.18 A flowchart outlining the fused classifier approach, where a machine learning model is used to form a weighted output from two convolutional neural network (CNN) classifiers. 205

7.19 Performance of the classifiers used in a fused classification scheme a) Control classifier b) Retrained classifier c) Theoretical optimal classifier. 208

7.20 A plot showing the optimal convolutional neural network (CNN) classifier, as a function of mean signal to background ratio (SBR) and SBR variance across a spectrogram. 209

7.21 Performance of the decision tree used for classifier fusion. 210

7.22 Estimates of predictor importance for the decision tree used in the fused classifier scheme. 210

7.23 A histogram of the weightings on the output of the control classifier, set by a decision tree model. 211

7.24 The scores output by the control classifier (green), retrained classifier (blue), and the overall weighted score (red) where the weighting was set by a decision tree model. The model was tested on a) Bird spectrograms b) Drone spectrograms. 212

7.25 Performance of the decision tree enabled bird-vs-drone convolutional neural network (CNN) classifier, where a fusion approach is used. . . 212

7.26 Performance of the decision tree enabled bird-vs-drone convolutional neural network (CNN) classifier, where a dynamic classifier selection method is used. 213

7.27 Receiver operating characteristic (ROC) curve for a drone size classifier, showing true positive rate vs false positive rate. 215

7.28 Drone size based classifier results a) 4 classes b) Including an unknown class. 217

A.1 Drone group classification performance where classification has been performed across whole files a) Low signal to noise ratio case, b) All data case. 231

A.2 Confusion matrices showing drone subgroup classification performance where classification has been performed across whole files a) Baseline case, b) Low signal to noise ratio case, c) All data case. 232

A.3 Confusion matrices showing drone size classification performance where classification has been performed across whole files a) Baseline case, b) Low signal to noise ratio case, c) All data case. 233

A.4 Confusion matrices showing single-stage drone model classification performance where classification has been performed across whole files a) b) Baseline case, 9 classes b) Baseline case, 7 classes (weaker classes combined) c) Baseline case, 7 classes (weaker classes removed) d) Low signal to noise ratio (SNR) case, 9 classes, e) All SNR case, 9 classes. 236

A.5 Confusion matrices showing multi-stage drone model classification performance where classification has been performed across whole files a) 2-stages: Group classification followed by model classification, b) 2-stages: Subgroup classification followed by model classification, c) 3-stage classifier. 238

B.1 Drone group classification F1 score median and interquartile range a) Low signal to noise ratio case, b) All data case. 239

B.2 Drone subgroup classification F1 score median and interquartile range a) Baseline case b) Low signal to noise ratio case, c) All data case. . . 240

B.3 Drone size classification F1 score median and interquartile range a) Baseline case b) Low signal to noise ratio case, c) All data case. . . . 241

B.4 Drone model single-stage F1 score median and interquartile range a) Baseline case, 9 classes, b) Baseline case, 7 classes (weaker classes combined) c) Baseline case, 7 classes (weaker classes removed) d) Low signal to noise ratio case, 9 classes e) All data case, 9 classes. 242

B.5 Drone model multi-stage classification F1 score median and range a) 2-stages: Group classification followed by model classification, b) 2-stages: Subgroup classification followed by model classification, c) 3-stage classifier. 243

List of Tables

1.1 An overview of unmanned aerial vehicle (UAV) surveillance methods [15][16] 4

2.1 The operating parameters of the Aveillant Gamekeeper 16U staring radar used for data collection. 32

4.1 An overview of the existing research into drone classification using radar. 75

5.1 Summary of data set A. 85

5.2 A table comparing the performance of six machine learning algorithms (with and without micro-Doppler features) and a CNN, for bird-drone classification. 103

5.3 Comparison of the depth, number of parameters, and training times of six different CNN architectures [7]. 111

5.4 The optimal percentage split of the training data across various signal to noise ratio (SNR) bins for training a robust convolutional neural network (CNN), found using Bayesian optimisation. 115

6.1 An overview of the models of drone for which data was obtained at Ku-band. 126

6.2 A summary of the data set that was obtained at Ku-band. 127

6.3 The number of frames used for training, testing and validation for drone group classification, where testing is performed on the baseline data set. 134

6.4 A summary of the handcrafted features extracted from the drone spectrograms. 158

7.1 Summary of the drone models included in data set B, their diameter, the amount of data recorded for each model, and the approximate span of ranges at which data was collected. 177

7.2 Size groupings of the drone models represented in data set B, based on diameter. 178

7.3 A summary of the data used for training the control classifier. 184

7.4 Summary of data used for training the retrained classifier, across data sets A and B. 197

7.5 Performance comparison of the control and retrained classifiers, grouped by size of drone. 199

7.6 The subsets used for training and testing four convolutional neural networks for a fusion model. 207

7.7 Summarised results of the fused classification scheme approaches considered. 214

C.1 Summary of the F1 scores achieved by the drone group classifiers. . . 244

C.2 Summary of the F1 scores achieved by the drone subgroup classifiers, where the rotary wing drones are grouped by number of rotors. 245

C.3 Summary of the F1 scores achieved by the drone size subgroup classifiers. 245

C.4 Summary of the F1 scores achieved by the drone model classifiers. . . 245

List of Abbreviations

C-UAS	Counter Unmanned Aerial System.
CAA	Civil Aviation Authority.
CART	Classification And Regression Tree.
CNN	Convolutional Neural Network.
COOL	Competitive Over-complete Output Layer.
CPI	Coherent Processing Interval.
CVD	Cadence Velocity Diagram.
CW	Continuous Wave.
DA	Discriminant Analysis.
DBN	Deep Belief Network.
DFT	Discrete Fourier Transform.
DT	Decision Tree.
DTFT	Discrete-Time Fourier Transform.
EM	Electromagnetic.
EMD	Empirical Mode Decomposition.
FCN	Fully Convolutional Network.
FFT	Fast Fourier Transform.
FMCW	Frequency Modulated Continuous Wave.
FT	Fourier Transform.
FTCN	Fully Temporal Convolutional Network.

GDI	Gini Diversity Index.
GPS	Global Positioning System.
HBR	HERM line to Background Ratio.
HERM	HElicopter Rotor Modulation.
kNN	k-Nearest Neighbours.
LDA	Linear Discriminant Analysis.
LSTM	Long Short Term Memory.
MIMO	Multiple-In Multiple-Out.
MLP	Multi Layer Perceptron.
NB	Naive Bayes.
PPI	Plan Position Indicator.
PRF	Pulse Repetition Frequency.
PRT	Pulse Repetition Time.
r-D	range-Doppler.
RADAR	RAdio Detection And Ranging.
RCS	Radar Cross Section.
ReLU	Rectified Linear Unit.
RF	Radio Frequency.
ROC	Receiver Operating Characteristic.
SAR	Synthetic Aperture Radar.
SBR	Signal to Background Ratio.
SCF	Spectral Correlation Function.
SNR	Signal to Noise Ratio.
STFT	Short Time Fourier Transform.
SVD	Singular Value Decomposition.
SVM	Support Vector Machine.
UAS	Unmanned Aerial System.
UAV	Unmanned Aerial Vehicle.

Chapter 1

Introduction

1.1 Background

Low altitude airspace is undergoing a global transformation due to the increasing popularity of unmanned aerial vehicles (UAVs). Recent years have seen a proliferation in drone numbers, as hobbyist drones become cheaper and easier to operate due to advances in technology. It is predicted that by 2030 there will be 76,000 drones in use in the United Kingdom [17]. The normal operating altitude of small drones is <1700 m [16] and UK legislation states that recreational drones must be flown no more than 400 ft (122 m) above the ground [18], signalling a fundamental shift in the use of low altitude airspace.

UAVs are estimated to lead to a £42 billion uplift in UK GDP by 2030 where over a third of drones will be used within the public sector to improve national security, enhance defence capabilities, and aid the emergency services [17]. Many private sector industries will also benefit from the ability to gather data remotely through the miniaturisation of camera technology, with the agriculture, construc-

tion, and transport industries expected to be front-runners in the uptake of UAV technology [19]. The number of potential applications will be increased as technology advances, allowing drones to carry heavier payloads, fly over greater distances, and even operate autonomously [20].

Whilst the use of UAVs can greatly benefit the UK economy, the rapid development of UAV technology poses inherent risks [21][22]. As UAVs occupy low-altitude airspace, there is risk of collision with manned aircraft, particularly during the take-off and landing phases. Experiments demonstrate that a 2 kg drone could cause critical damage to the windscreen of a passenger jet, whilst a 400 g drone could shatter the windscreen of a helicopter [23]. The UK Airprox Board (comprising of aviation experts from across civil and military organisations) provides safety recommendations and risk assessments on 'near-miss' incidents with manned aircraft. The increasing number of drones in an already congested airspace increases the risk of collisions with manned aircraft, with the UK Airprox Board reporting 564 UAV incidents from 2014 to 2020 [24]. Whilst unregulated UAV usage could have catastrophic consequences, there are also economic risks: Gatwick Airport was closed for 33 hours in December 2018 due to safety concerns following reports of a drone sighting leading to an estimated cost of over £50 million [25][26].

To avoid accidental collisions and facilitate the safe integration of UAV technology into society, UAV usage must be regulated [21]. The UK government have released a guide to the responsible and lawful operation of UAVs, outlining the regulations laid out by the Civil Aviation Authority (CAA) [18]. However, this does not prevent the intentional misuse of UAV technology in order to damage critical infrastructure [27], conduct illegal surveillance [28], or threaten life [29], making the safe use of UAVs an issue of public interest [30]. The ability to carry payloads via drone poses further security concerns. In 2017 a programme was formed by the UK government to mitigate security threats posed by drones in prisons [31], whilst a

Guernsey prison introduced an anti-drone fence to prevent the smuggling of contraband into the prison [32]. Further, there are concerns around the use of drones to incite terror, with the potential to fly over crowded areas and release chemical or biological agents from a distance [33]. Small UAVs are also being used increasingly in war zones (such as the current conflict in Ukraine [34]) to survey the battlefield, gather intelligence, and guide explosives [35].

It is imperative to develop techniques for intercepting malicious drones both in military and civilian environments, making counter-drone (C-UAS) technology a topic of global importance. C-UAS strategies include electronic countermeasures (such as RF jamming or GPS spoofing) or interception using physical methods (such as nets, lasers, ballistics, and even trained raptors [16][36].) Fig. 1.1 shows the trend in Google searches including the term 'C-UAS' over time, demonstrating growing interest in this area. In 2018 there were at least 235 C-UAS products on the market or in development across 33 countries, illustrating a global interest in counter-drone technology in both academia and industry [35].

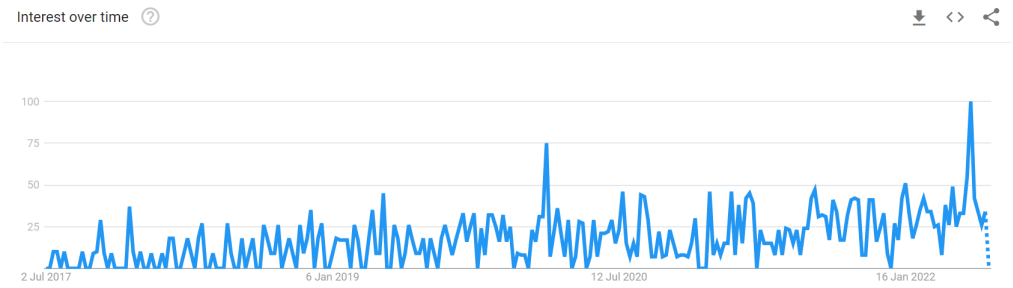


Figure 1.1: Relative interest over time for the search term "C-UAS" (counter unmanned aerial system) on Google [1].

The detection, tracking, and non-cooperative identification of drones is a prerequisite to interception. Common methods of detecting drones are outlined in Table 1.1, along with their relative advantages and disadvantages [15][16]. Long range surveillance is desirable to maximise the time available to assess and respond to threats, and surveillance methods need to be capable of operating 24 hours a day

in all weather conditions and environments, making radar the ideal solution.

Table 1.1: An overview of unmanned aerial vehicle (UAV) surveillance methods [15][16]

Method	Advantages	Disadvantages
Radar	Long range (>5km) All weather, 24 hour surveillance Capable of detecting and tracking multiple UAVs simultaneously	Expensive Requires line of sight Susceptible to false positives Requires frequency licensing Requires trained operators
Lidar	High resolution Effective UAV detection	Low range Expensive Requires line of sight High attenuation in poor weather
Cameras (visual and/or infra-red)	Drones can be distinguished from birds Images/videos can recorded as forensic evidence Capable of UAV classification Can be combined with neural networks	Only effective at short ranges Requires line of sight Poor in low light (optical) Low resolution (thermal)
Audio	Can extend beyond line of sight Low cost Capable of UAV classification using library of sounds	Low range (<300m) Returns can be forged Ineffective in noisy environments Library of sounds will not include all models of drone
Radio Frequency (RF)	Can locate both UAV and controller All weather Can provide early warning before drone is even launched	Low range (<600m) Requires line of sight Less effective in RF-heavy environments Cannot detect autonomous drones Can only detect a small number of UAVs at a time Only operates across certain frequency bands

1.1.1 Motivation

Whilst radar possesses several characteristics that make it suitable for UAV surveillance, there are also a number of challenges (Table 1.1). This thesis seeks to explore methods of drone classification, contributing to radar-based UAV surveillance through a reduction in the number of false positives, reducing the reliance upon highly trained radar operators.

Drones have a small radar cross section (RCS), and their slow velocities and low flight altitudes make them difficult to distinguish from clutter (particularly in urban environments), meaning that a sensitive radar is required for detection [37]. As a consequence, other slow-moving targets with a low RCS (such as birds [38][39][40])

are also detected, presenting as false alarms and reducing the effectiveness of drone surveillance methods. Automating the filtering of false positives from true UAV sightings will lower the reliance on highly trained radar operators. Consequently, a high performing classification algorithm is needed such that only detections that truly belong to drones are reported, avoiding the unnecessary deployment of costly countermeasures and maintaining confidence in the system. Whilst a low false positive rate is desirable, a low false negative rate is of utmost importance from a drone operator's standpoint, as the consequences of failing to detect malicious drones are potentially life threatening [35].

1.2 Aims and Objectives

The development of novel approaches to radar classification of UAVs using machine and deep learning techniques forms the basis of this thesis. Deep learning techniques can be used to extract complex classification features directly from the data, providing an advantage over more traditional classification methods. Convolutional neural networks (CNNs) use local receptive fields to extract spatial relationships from the data, and are robust to translations, making them highly suitable for image classification. This research will seek to answer whether CNNs are suitable for distinguishing drones from birds in the radar domain. This work uses data from two radar systems: an L-band staring radar and a Ku-band FMCW staring radar prototype.

Drones and birds produce characteristic micro-Doppler signatures due to their differing flight mechanisms, therefore radar spectrograms are chosen as the input to CNNs as they provide a representation of the signal in the frequency domain. Spectrograms show the distribution of frequencies over time, allowing the classifier to learn kinematic features (from the frequency and shape of the main body return)

as well as power-based features and micro-Doppler features. Further, as CNNs are capable of extracting spatial information, it is hypothesised that they may be able to learn relationships between different frequency components within the spectrogram, facilitating classification of drone model.

Radar for C-UAS is a rapidly developing area of research, and this thesis aims to build on the existing knowledge base by considering UAV classification as a continuum of problems and achieving classification under increasingly difficult conditions. It is not sufficient to develop a C-UAS solution that is only effective in a constrained environment, therefore this thesis looks to place limits on the ranges at which reliable classification can be achieved, and on the size of drone that can be classified.

As well as simply confirming the presence of a drone, a further aim is to develop a methodology for identifying drone type. This research will aim to investigate drone model classification, from coarse group classification down to model-level recognition.

1.2.1 Contributions

To the author's knowledge, this is the first comprehensive investigation of CNN use for long range (up to 5 km) drone-bird discrimination using radar spectrograms obtained under real operational conditions. The data used was challenging from a classification perspective, contributing to the community's understanding of performance limitations in increasingly demanding environments [41]. Data consisted of a wide variety of drone models (20 models, including small drones), extensive range coverage, and both static and dynamic clutter in urban, industrial, and rural environments. The main contributions of this work are as follows:

- **Establishing CNNs for drone-bird classification [42].**

  The suitability of CNNs for this purpose was demonstrated via performance comparison with a number of traditional machine learning classifiers.

- **Performance evaluation as a function of signal to background ratio, assessing the limitations of the classifier using both simulated reduced SNR [7][43] and real low SNR data [41].**

  The data contain many instances where the drone HElicopter Rotor Modulation (HERM) lines are not visible above the noise floor or background, with significant implications for the maximum distances at which varying sizes of drone could be reliably classified.

- **Novel methodologies for training classifiers that are robust to low SNR test data.**

  A recommendation arising from this work is to train on a range of SNRs either through direct measurement or by deliberately degrading training data acquired at a high SNR, making the training data more representative of the data that the classifier will be exposed to during deployment.

- **Bayesian optimisation for maximising classifier robustness through the augmentation of training data [7].**

  Bayesian optimisation is used to determine the augmentation parameters that maximise classifier performance, reducing the effect of low SNR upon accurate classification. This methodology could be used when designing bespoke training strategies.

- **Dynamic classifier selection for making an intelligent choice of classifier based on the features present in the spectrogram.**

  Following the development of a CNN suited to high SNR test data, and a second CNN optimised for long-range classification, both hard and soft fusion

approaches are considered, optimising performance and laying the foundations for a cognitive approach to classification.

- **A comprehensive investigation of CNN use for drone-to-drone classification in the HERM line regime, under real operational conditions [44].**
  Results demonstrate that CNNs are highly capable of handling classification based on drone group (fixed wing and rotary wing drones), and subgroup (hexacopters and quadcopters). Systematic drone model level classification is performed, illustrating the challenge of distinguishing between similar models of drone.

- **Performance evaluation as a function of the time on target.**
  There is a trade-off between accuracy and the time on target, with implications for the time taken to respond to threats.

- **Comparison of multiple classification strategies.**
  It is shown that grouping drones based on their weight leads to improved subgroup separation compared to grouping them based on the number of rotors. Both single stage and multi-stage classification strategies are explored for drone model classification.

- **Exploration of features for drone model classification.**
  Handcrafted features are extracted from the spectrograms, and this feature space is explored across various drone models.

- **Systematic assessment of the effect of processing interval on classification performance.**
  The relationship between Fourier Transform window length, coherency, and performance is investigated. It is shown that classification performance could be significantly improved by training decision tree classifiers on various window lengths in parallel and fusing the results.

1.2.2 Publications

The novel contributions of this thesis led to the following publications:

1. Dale, H., Antoniou, M., Baker, C.J., et al. (2022) “Convolutional Neural Networks for Drone Model Classification.” In 2021 18th European Radar Conference (EuRAD). April 2022. pp. 361–364

2. Dale, H., Jahangir, M., Baker, C.J., et al. (2022) “Convolutional Neural Networks for Robust Classification of Drones.” In 2022 IEEE Radar Conference (RadarConf22). March 2022. pp. 1–6

3. Dale, H., Baker, C., Antoniou, M., et al. (2022) SNR-dependent drone classification using convolutional neural networks. IET Radar, Sonar & Navigation, 16 (1): pp. 22–33

4. Dale, H., Baker, C., Antoniou, M., et al. (2021) “A Comparison of Convolutional Neural Networks for Low SNR Radar Classification of Drones.” In 2021 IEEE Radar Conference (RadarConf21). Atlanta, GA, USA, 7 May 2021. IEEE. pp. 1–5

5. Dale, H., Baker, C., Antoniou, M., et al. (2020) “An Initial Investigation into Using Convolutional Neural Networks for Classification of Drones.” In 2020 IEEE International Radar Conference (RADAR). Washington, DC, USA, April 2020. IEEE. pp. 618–623

6. Sim, J., Jahangir, M., Fioranelli, F., Baker, C., Dale, H. (2019) “Effective Ground-Truthing of Supervised Machine Learning for Drone Classification.” In 2019 International Radar Conference (RADAR). Toulon, France, September 2019. IEEE. pp. 1–5

1.3 Thesis Outline

This thesis investigates machine learning techniques for the classification of drones based on their radar signatures. Chapter 2 presents the radar theory necessary for a contextual understanding of the concepts explored in later chapters. In Chapter 3, the machine learning techniques that underpin the thesis are introduced.

The existing research into the classification of drones using radar is reviewed in Chapter 4. Both drone-bird discrimination and drone model classification are considered, providing an overview of the current research landscape that this thesis builds upon. Knowledge gaps are identified, providing context for the novel contributions presented in later chapters of this thesis.

It is necessary to first establish an appropriate model for bird-drone classification. Chapter 5 presents an investigation into the unique features that may be used to distinguish between birds and drones at L-band, and traditional machine learning approaches are used to set a performance benchmark against which convolutional neural network (CNN) performance is compared. An artificial reduction in spectrogram SNR then allows for model performance to be explored under a range of conditions, and an exploration into methods of improving classifier robustness is presented.

In addition to distinguishing drones from other objects in airspace, similar techniques can be used to classify drones, building a more complete picture of airspace occupancy. In Chapter 6, a CNN is trained on spectrograms obtained using a Ku-band Frequency Modulated Continuous Wave (FMCW) staring radar prototype, with the aim of performing drone classification from coarse groupings down to model level recognition.

It is not enough to report high performance under idealised conditions - instead, the solution space must be pushed forwards and classification attempted under increasingly challenging conditions. In Chapter 7, a comprehensive data set representing a range of environments, target ranges, and target types is used to investigate the limitations on performance in high clutter environments, and for unseen models of drone. Performance is evaluated as a function of model and signal to background ratio (SBR).

Finally, the conclusions of the research presented in this thesis are outlined in Chapter 8. Future avenues of research are suggested, building upon the contributions arising from this thesis.

Chapter 2

Foundations of Radar

2.1 Introduction

Radio waves have been used to detect the presence of ships and aircraft since the early 1900s [45][46][47]. By World War II, radar technology had been independently developed across eight different nations [47]. Britain's "Chain Home" was a series of radar systems used in World War II to provide vital intelligence about the direction and distance of German aircraft from British shores [48]. In addition to military applications, radar is now commonly used across a wide range of fields including aviation, meteorology, and geology. Advances in signal processing and machine learning have seen a proliferation in radar applications in recent years, for example within the automotive industry for autonomous vehicles [49][50], and in healthcare for the non-invasive monitoring of patients [51][52]. This thesis aims to use radar alongside machine learning techniques to differentiate between objects in airspace such that drones can be identified.

This section covers the relevant radar theory for this thesis. The basic principles

of target detection and location are outlined, along with the Doppler and micro-Doppler effects. Factors that impact on the strength of the target signal are discussed in the context of drone classification. The concept of staring radar is introduced, and the advantages of staring radar over conventional scanning radar for the detection of drones are presented. The basic principles of the two staring radar systems used for data collection throughout the thesis are outlined: an L-band pulsed staring radar, and a Ku-band FMCW staring radar prototype. Finally, the process of spectrogram formation is demonstrated, as spectrograms are used to train CNNs in later chapters.

2.2 Radar Fundamentals

2.2.1 Measuring Range

Radar (RAdio Detection And Ranging) uses radio waves to detect and locate targets. A stable oscillator emits electromagnetic (EM) waves which are then amplified, and transmitted by the antenna. The wave propagates through space until it encounters an object with a different permittivity, then a current is induced and some of the energy is re-radiated back into space. A fraction of this energy is reflected back towards the antenna, where a signal is received, amplified, and processed.

Comparing the received signal to the transmitted signal reveals information about the location of the target. Pulsed radar systems emit series of short pulses, and comparison across many consecutive received pulses reveals changes in the position and motion of the target.

As EM waves travel at a known velocity (the speed of light, c), the time delay, t, between emitting a pulse and measuring the return depends on the range, R, of

the target,

$$R = \frac{ct}{2}. \tag{2.1}$$

The radio waves travel a round-trip from the radar to the target in time t, introducing the factor of 2.

As pulsed radars cannot receive and transmit simultaneously, there is a minimum detection range that depends on the time taken to transmit a pulse and switch to receive mode. The maximum unambiguous range, R_{max}, is half of the longest distance that a pulse could travel to in the time between consecutive pulses being transmitted. If the target exceeds this range, the received signal could have been from the last transmitted signal, or from an earlier pulse, placing an uncertainty on the range measurement. The maximum unambiguous range, R_{max}, is given by

$$R_{max} = \frac{c(T - \tau)}{2} \tag{2.2}$$

where T is the pulse repetition time (PRT) and τ is the duration of a single pulse.

The finite pulse width places limits on the resolution of range measurement. The range resolution, ΔR, is the minimum range separation between two targets at which their radar echoes can be separated from one another. Targets will not be resolved if the waves reflected from each target both reach the receiver within the time duration of a single pulse. The range resolution is therefore defined as

$$\Delta R = \frac{c\tau}{2}. \tag{2.3}$$

Decreasing pulse length improves the range resolution but limits the power contained within a single pulse, with significant implications on the maximum range at which a given target can be detected. Phase coding and frequency modulation techniques can be used to increase the range resolution and maximum unambiguous range.

2.2.2 The Radar Range Equation

The signal to noise ratio (SNR) dictates the limits of detection, as a target signal can only be detected if it is distinguishable from noise. The radar range equation describes the relationship between the radar design parameters, the target parameters, and SNR, setting an upper limit on the range at which a given target can be detected [53]. Conversely, the equation can also be used to determine the minimum 'size' of target detectable at a given range.

An isotropic source emits a pulse that spreads out spherically, therefore the signal intensity at a distance R from the source is the transmitted power divided by the surface area of a sphere of radius R. However, in radar, directional beams are used to concentrate the power in a particular direction. The transmit gain, G_t, describes the ratio of the power per unit solid angle in a particular direction, compared to the power if the source were isotropic. Therefore, at a range R from the antenna, the transmitted power density, Q_t, is given by

$$Q_t = \frac{P_t G_t}{4\pi R^2}.$$

The radar cross section (RCS) is an effective area that describes the proportion of the transmitted power that is scattered isotropically by a target. RCS depends not only on the physical size of the target, but also its materials, shape, and the aspect angle. The RCS also depends on the choice of radar frequency and polarisation [38]. The power density reflected back from a target, Q_{ref}, is given by

$$Q_{ref} = \frac{P_t G_t \sigma}{4\pi R^2},$$

where σ is the RCS.

The radiation is again spread over a sphere of radius R as the reflected signal

propagates back towards the receiver. The power received by the antenna, P_r, is scaled by the effective area of the antenna, A_e,

$$P_r = \frac{P_t G_t A_e \sigma}{(4\pi)^2 R^4}.$$

The effective area of the antenna is defined as $\frac{G_r \lambda^2}{4\pi}$, where G_r is the receiver gain and λ is the wavelength [54]. Thus the received power from a single pulse reflecting from a target with an RCS of σ at range R can be written as

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

Increasing the number of pulses increases SNR. If the returns from n consecutive pulses are integrated over coherently, then the signal power increases by a factor of n,

$$P_r = \frac{P_t G_t G_r n \lambda^2 \sigma}{(4\pi)^3 R^4}.$$

As well as the target signal, signals with random amplitude and phase are also detected as noise. Sources of noise are internal (active circuits within receiver) and external (solar and cosmic sources). Thermal noise comes from charged particles with random motion, and is uniformly distributed over all frequency bands. As only the noise power within the receiver bandwidth, B, will have an effect on performance, the thermal noise power, P_n, is given by

$$P_n = kT_sB,$$

where k is Boltzmann's constant and T_s is the system noise temperature.

The noise power can be written in terms of the standard temperature (290K)

through the introduction of the noise figure, F:

$$P_n = kT_0FB.$$

Thus, the signal to noise ratio at the receiver is given by

$$SNR = \frac{P_t G_t G_r n \lambda^2 \sigma}{(4\pi)^3 R^4 k T_0 F B L} \tag{2.4}$$

where L represents additional losses such as atmospheric attenuation, resistive losses on transmit and receive, and losses due to signal processing [55].

Eq. (2.4) demonstrates the sensitivity of SNR to range and RCS. Whilst at L-band the RCS of a hexacopter body is observed to be in the region of 0.1 m^2 [56], the RCS of the blades is expected to be considerably lower, making them significantly more difficult to detect [38]. If the blade returns are found to be a necessary feature for distinguishing between birds and drones, then the low blade RCS will limit performance. The desired false alarm rate for a particular application sets limits on the SNR (and therefore the range) at which a given target can be detected. In later chapters, performance is investigated as a function of SNR, setting limits on the target range and RCS required for classification.

SNR is dependent on the target range to the fourth power, where doubling range decreases SNR by a factor of 16. Note that doubling the integration time has the same effect upon the SNR as doubling the power, and increasing the integration time is much cheaper and easier than changing the radar hardware. Therefore increasing the integration time could theoretically extend the range at which targets can be detected and classified. However, this is only valid for the time period over which the returns remain coherent. This will be investigated further in Chapter 6.

Finally, the radar equation illustrates the importance of radar hardware choices

such as gain, antenna area, power, and choice of carrier frequency. Generally atmospheric attenuation increases with frequency, limiting the range capabilities of high frequency radar, particularly in poor weather conditions. Secondly, RCS is a function of wavelength. This thesis presents the results of drone classification at both L-band and Ku-band. At L-band, the wavelength (15-30 cm) is comparable to the length of a drone propeller, placing the scattering regime in the Mie region where RCS fluctuates with small changes in rotor length or wavelength [57][58]. At Ku-band, the wavelength (1.67-2.5 cm) is considerably less than the length of a drone propeller, placing the scattering in the optical regime, where RCS is more stable [57].

2.2.3 Azimuth and Elevation

Range only provides information about the distance of the target from the radar, and vertical and horizontal angular offsets (elevation and azimuth angles, respectively) are required in order to determine target position. The energy radiated from an antenna is not isotropic, and is highly dependent on angle. Model a linear array of antenna elements as a coherent beam of light passing through a series of slits, each separated by a distance d. An interference pattern Fig. 2.1 [2] is produced as the angle relative to the boresight is varied, due to the phase differences between waves travelling different path lengths from each antenna element. Thus, the radiation pattern consists of a main lobe where each element combines in phase, and a series of nulls where destructive interference occurs. From Young's Double-Slit Experiment [59], the lobes have an angular separation of $\frac{\lambda}{d}$ in the far-field limit, therefore the angular resolution, θ, of an antenna of aperture length L can be approximated as

$$\theta = \frac{\alpha\lambda}{L}, \tag{2.5}$$

where α is some constant. A typical measurement of beam width is the half power beam width, also known as the 3 dB beam width, which is the angle at which the power is at least 50% of the power along the boresight.

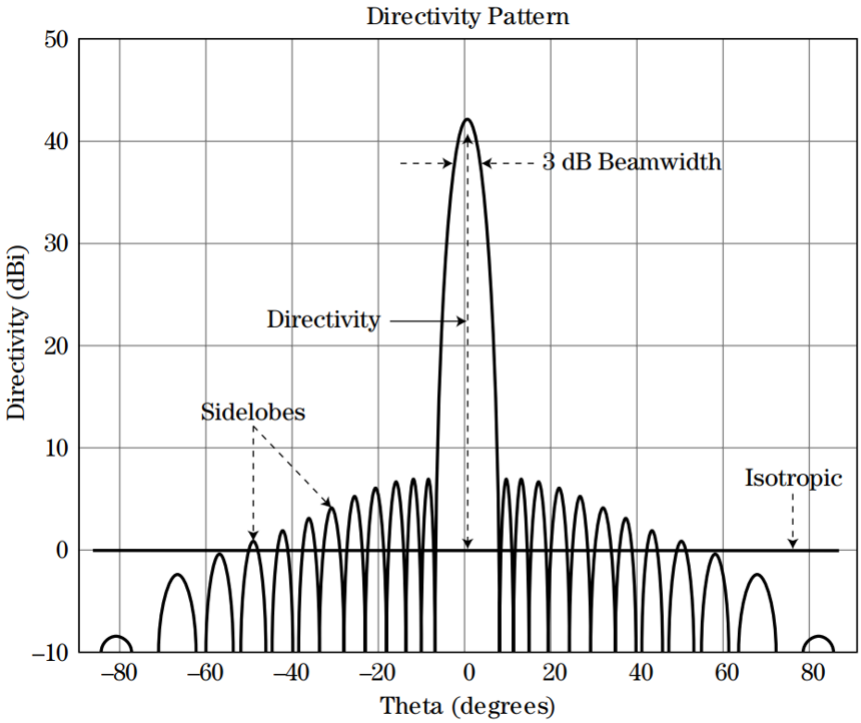


Figure 2.1: The radiation pattern of an antenna consisting of a linear array of elements, as a function of angle from the boresight, Theta [2].

In scanning radar, the beam is electronically steered by adding a phase offset to each of the antenna elements. The combination of constructive and destructive interference changes the angular offset of the main lobe, effectively steering the wave front. This technique allows for flexible manoeuvring.

When simply scanning a pencil beam across space (whether mechanically or electronically), targets can only be resolved if they are separated by an angle greater than the beam width. However, more accurate angular estimates can be made using techniques such as sequential lobing, conical scan, and monopulse [60].

2.3 Doppler and Micro-Doppler

2.3.1 The Doppler Effect

The Doppler effect is the phenomenon by which the frequency of a wave changes when there is relative motion between the source and the receiver. As a receiver moves towards the source it experiences a greater number of wave cycles per second, therefore an increase in pitch. The Doppler effect applies to all wave motion, and consequently the received signal from a target moving with velocity v relative to a radar system experiences a frequency shift f_D, where

$$f_D = \frac{2v}{\lambda} cos(\alpha), \tag{2.6}$$

and α is the angle between the direction of travel of the transmitter and the direction of travel of the target [61]. The maximum Doppler shift is experienced when a target moves radially along the radar boresight, and is zero when the target moves tangentially.

Provided the system is coherent, comparing the phase of the received wave to the phase of the internal oscillator allows for a measure of target velocity, with increasingly small changes in phase becoming apparent over time. Pulsed radar can be used to measure target velocity by comparing the phase shift over consecutive pulses, provided the phase is continuous across pulses.

Of course, the Doppler shift is much smaller than the carrier frequency, therefore the received signal is mixed to convert the signal to baseband. Upon mixing, the received wave is decomposed into two channels: an in-phase, I, component and quadrature, Q, component. Forming a complex sum of the two channels allows

independent calculation of both amplitude and phase. The result is a sinusoid with a frequency component equal to the Doppler frequency, and applying a Fourier Transform (FT) results in a peak at the Doppler frequency.

Integrating over n pulses results in a narrower frequency peak as n increases. The Doppler resolution, Δf_D, is the reciprocal of the integration period,

$$\Delta f_D = \frac{PRF}{n} \tag{2.7}$$

where PRF is the pulse repetition frequency. Increasing the processing interval increases SNR and frequency resolution provided the target remains coherent, aiding the detection of low RCS targets. However, as drones are capable of dynamic manoeuvres, the coherent processing interval (CPI) is limited [62]. High Doppler resolution allows distinction between targets moving at similar velocities, and between slow moving targets and clutter. Stationary clutter such as the ground, trees, and buildings, can be removed by the application of a high-pass filter, however, this enforces a minimum detectable velocity where slow moving targets (or targets with a low radial velocity component) will be filtered out.

A signal can only be recreated if the sampling rate is at least twice the maximum frequency present in the signal [63]. Therefore the maximum unambiguous Doppler frequency, $f_{d_{max}}$ is

$$f_{d_{max}} = \frac{2v_{max}}{\lambda} = \frac{PRF}{2}, \tag{2.8}$$

and frequencies above this will appear aliased. A high PRF increases the unambiguous Doppler frequency, but has an adverse effect on the range ambiguity – there is therefore a balance to strike between frequency and range measurements.

Phase noise occurs as the phase and frequency of the internal oscillator is unstable over time. Instability in the oscillator, internal clock, or transmitter introduces a

band of noise across all frequencies within a range gate. These effects are particularly noticeable in regions of high clutter [64], where phase noise can dominate over the thermal noise floor, hindering detection.

2.3.2 The Micro-Doppler Effect

Drones and birds have a similar RCS, fly within a similar range of velocities, and occupy similar regions of airspace [58]. Therefore, there will be considerable overlap between the two types of target if the only available classification features relate to signal strength, positional coordinates, and flight velocity. The method of flight provides a discriminating feature between a drone and a bird: drones have rapidly rotating propeller blades, whilst birds flap their wings at a much lower frequency. Fortunately, target constituents induce frequency modulations, known as micro-Doppler, which are detected on top of the bulk frequency change whenever target components vibrate or rotate [65]. The micro-Doppler effect is a phenomenon that has been observed for rotating helicopter blades [66][67][68][69], rotating jet engine turbines [66][70][71], and human limbs swinging whilst walking [72][73][74]. One expects drones to have markedly different micro-Doppler signatures to birds, providing a potentially discriminating feature for classification that is characteristic of target motion [75][76][40].

When the integration period is significantly shorter than the period of rotation, the micro-Doppler signature is characterised as a sinusoid, known as a "blade flash", as shown in Fig. 2.2[3][77]. The blade flash is maximised when a blade is moving radially towards (or away from) the radar, and is zero when the blade is moving tangentially. The period of rotation can be measured from the periodicity of the micro-Doppler returns [3][77], and the number of blades can be inferred from the symmetry of the blade flashes [78]. Finally, blade length r can be estimated from the

maximum Doppler velocity using the relationship $v = 2\pi\Omega r$, where Ω is the number of blade rotations per second [79][3].

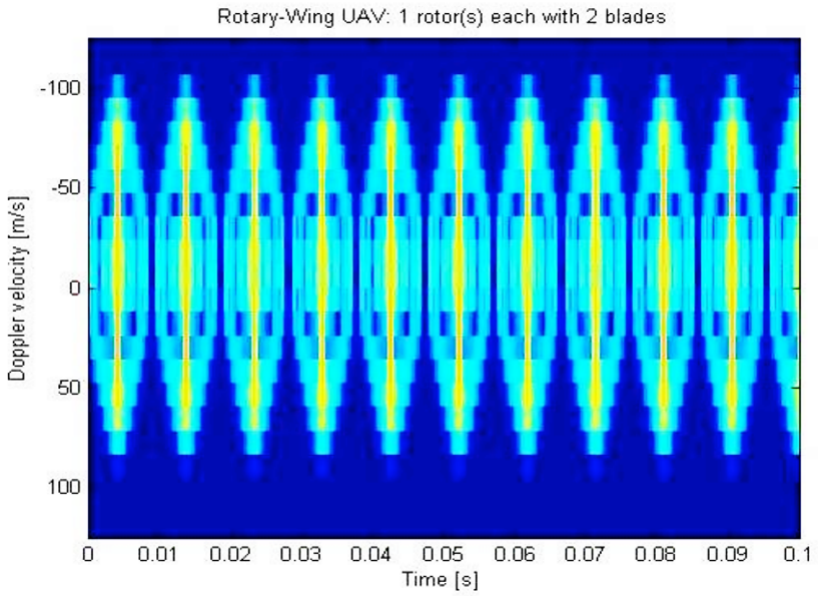


Figure 2.2: A diagram showing the spectrogram of a single rotor drone with two blades [3].

A high sampling rate is required to extract blade-flashes as the blade rotation must be over-sampled, however, a high PRF means a high range ambiguity. Further, to achieve sufficient SNR for detection of micro-Doppler at longer ranges, and to resolve slow-moving targets from stationary clutter, a longer integration time is required. When the drone blade rotation frequency exceeds the PRF, the blade flashes are under-sampled resulting in characteristic frequency modulations known as HElicopter Rotor Modulation (HERM) lines [80]. In [4] and [81], UAV blade flash and HERM line regimes are explored, where it is demonstrated that the micro-Doppler regime depends on both the target parameters (rotor frequency, geometry, and rotor length) and the radar parameters (wavelength and integration time) and that altering these parameters can drastically change the appearance of the spectrogram as shown in Fig. 2.3 [4]. The PRF of the Aveillant Gamekeeper 16U is approximately 7.5 kHz, and integrating over 2048 pulses results in a sampling frequency of approximately 3.6 Hz, which is considerably lower than the frequency of

rotation of a UAV propeller ($\approx$ 100 Hz). The work presented in this thesis therefore considers classification in the HERM line regime.

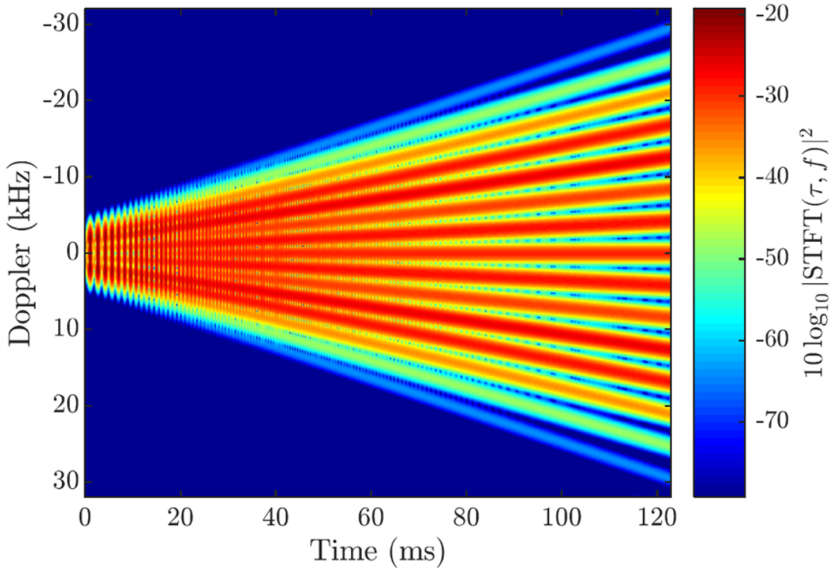


Figure 2.3: A simulated drone spectrogram where the rotor blade accelerates and the Fourier Transform window is fixed. As the rotation speed increases, helicopter rotor modulation (HERM) lines appear [4].

The HERM line spacing and shaping is characteristic of the UAV's physical properties (weight, number of rotors, number of blades, and blade length) and trajectory (velocity, pitch, and acceleration). Mechanisms for estimating the blade rotation rate from HERM lines are outlined in [82] and [83]. In [83] a methodology for estimating the number of blades is outlined, where the micro-Doppler modulations are a combination of frequency modulation from the blade motion relative to the radar, and amplitude modulation from the changing aspect angle and RCS of the blades. If this is the case then there are two frequencies: the chopping frequency due to frequency modulation, equal to the rotation rate multiplied by the number of blades, and secondly, a spool frequency from the changing RCS over a cycle [66].

In [84], Eq. (2.9) was derived to describe the signal from a drone with M rotors

with B blades of length L,

$$V(t) \propto L \sum_{m=1}^{M} \sum_{b=1}^{B} e^{-i\frac{4\pi}{\lambda}(R_0+\frac{L}{2}cos(\phi_t^{m,b}))} sinc(\frac{L}{2}\frac{4\pi}{\lambda}cos(\phi_t^{m,b})) \tag{2.9}$$

where $\phi_t^{m,b}$ is a term that encapsulates the rotation speed and angular offset of each blade. The equation was populated with rotor speeds from an experimental campaign and used to simulate a spectrogram. Comparison to the genuine spectrogram recorded using an Aveillant L-band staring radar revealed accurate HERM line appearance and behaviour. Therefore, the HERM lines can be understood as the result of applying a long windowed Fourier Transform to the summed contributions from each of the blades and rotors, where the differences in blade position leads to a path difference relative to the bulk component.

2.4 Fourier Analysis

Fourier analysis is used to decompose a signal into its frequency components. The Fourier Transform (FT) is used to analyse the frequency content f of a continuous signal $x(t)$,

$$X(f)_{FT} = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft}dt. \tag{2.10}$$

A radar signal is received over a finite interval and is sampled in time, therefore the Discrete-Time Fourier Transform (DTFT) can be used for frequency domain analysis,

$$X(f)_{DTFT} = \sum_{n=0}^{N-1} x[n]e^{-i2\pi fn}, \tag{2.11}$$

where the signal $x[n]$ is sampled over N samples.

Note that the DTFT is intended for periodic signals with an integer multiple of complete waves within the sampling period. When the signal does not meet this criterion there is a discontinuity between the start and end points, leading to

a broadening in the frequency domain. To remove this discontinuity, the signal is multiplied by a window function that tapers towards zero at each end. However, tapering introduces sidelobes as the DTFT of a rectangular pulse is a sinc function, with a first sidelobe that sits 13 dB below the main lobe.

Suppression of sidelobe amplitude comes at the cost of an amplitude reduction and a broadening of the main lobe, therefore there is a balance to consider when choosing a window function. Choice of window function is application dependent, and there are many different window functions with their own relationships between sideband attenuation and the effect on the main lobe [85]. Fig. 2.4 shows the effect of applying a Hamming window to a rectangular pulse before applying a DTFT [5].

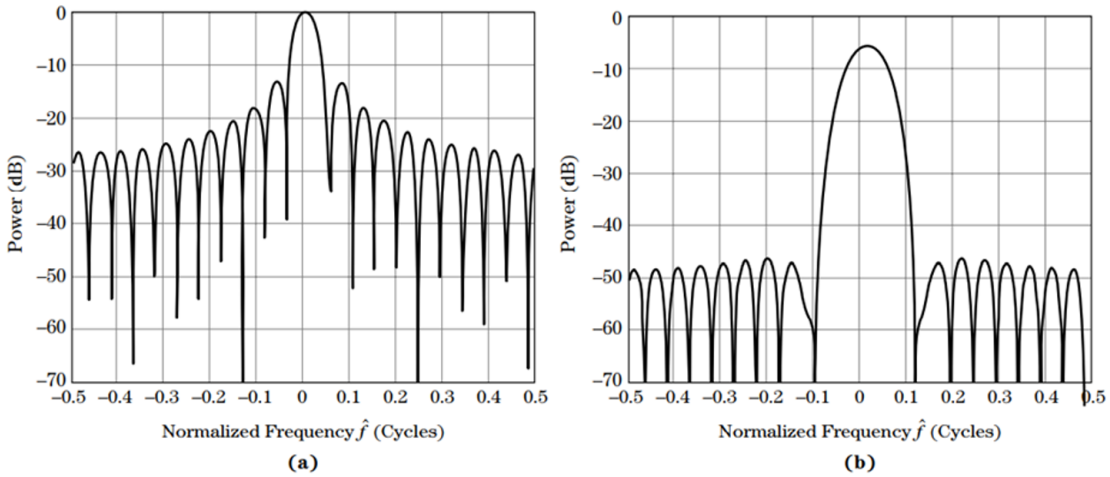


Figure 2.4: The discrete-time Fourier transform (DTFT) of a rectangular pulse a) With no window function b) Using a Hamming window function [5].

The DTFT is sampled in time but continuous in frequency, whilst the Discrete Fourier Transform (DFT) is sampled in both time and frequency. The DFT allows frequency domain analysis of a digitised signal over K samples in the frequency domain with discrete frequency k,

$$X(k)_{DFT} = \sum_{n=0}^{K-1} x[n] e^{-i\frac{2\pi}{K}nk}. \tag{2.12}$$

The Fast Fourier Transform (FFT) is an algorithm which can be used to compute the DFT of a signal quickly and efficiently by reducing the complexity of the calculation using symmetries [86][87].

2.5 Frequency Modulated Continuous Wave Radar

Continuous wave (CW) radar is an alternative to pulsed radar, whereby a high frequency signal is continuously transmitted. Doppler sensitivity is high due to the high carrier frequency, making CW radar suitable for a range of applications such as gauging speed or sensing motion. CW radar systems are easier to manufacture and operate than pulsed radar, and tend to be smaller in size, however, target distance cannot be measured due to the absence of pulses.

FMCW radar is a form of CW radar where the carrier frequency is modulated throughout the transmit period, as shown in Fig. 2.5. Provided the frequency modulation is coherent and deterministic, the change in frequency between the transmitted and received wave, Δf, allows estimation of the time delay, and consequently, the range of the target. Eq. (2.1) can be modified to provide an estimate of target range,

$$R = \frac{c|\Delta f|}{2(\frac{d(f)}{d(t)})}, \tag{2.13}$$

where $\frac{d(f)}{d(t)}$ is the frequency gradient of the modulated wave. The minimum detectable distance is comparable to a wavelength. The maximum unambiguous range is set by the period of repetition of the frequency sweep. In monostatic configurations, the maximum operational range is limited in practice by the transmit power that must be kept low to avoid leakage between the transmitter and co-located receiver.

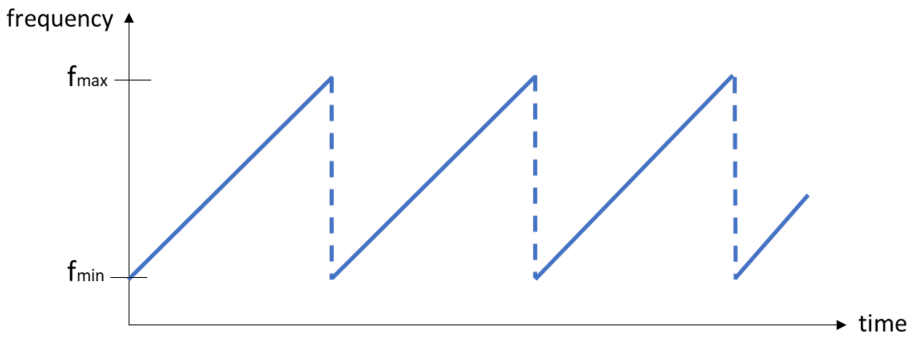


Figure 2.5: Frequency as a function of time for a transmitted wave with a sawtooth modulation pattern.

The range resolution ΔR is set by the bandwidth, $B = f_{max} - f_{min}$, where

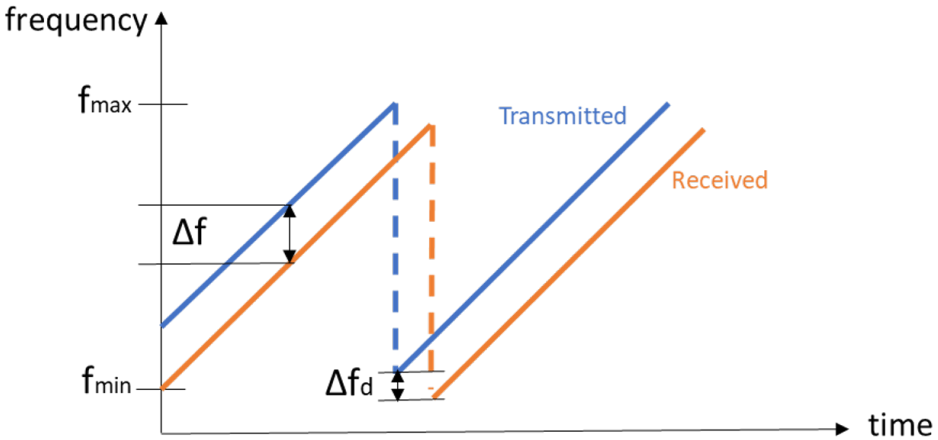
$$\Delta R = \frac{c}{2B}. \tag{2.14}$$

This is equivalent to Eq. (2.3), only substituting pulse width for $\frac{1}{B}$.

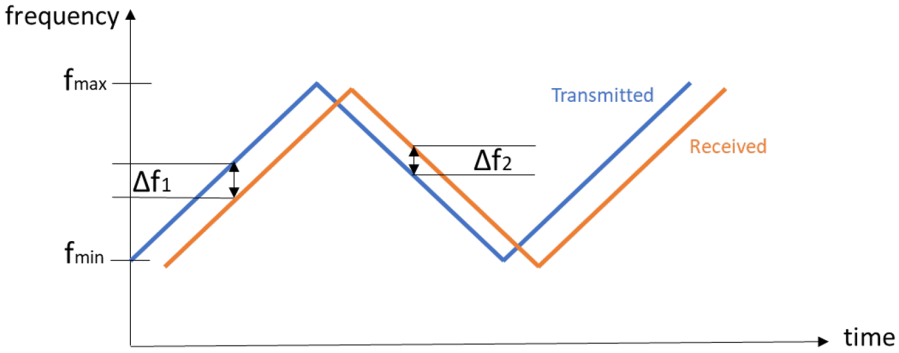
If a target is moving relative to the radar, the received signal will have a Doppler frequency shift in addition to the frequency shift due to the target range. If a sawtooth modulation pattern is chosen, there is no way to disentangle the range information from the velocity information, and changes in Doppler manifest as an error on the range measurement (Fig. 2.6a). However, triangular frequency modulation allows measurement of both range and Doppler frequency shifts ($f(R)$ and $f(D)$, respectively), through the frequency offsets at the rising and falling edges of the wave (Fig. 2.6b),

$$f(R) = \frac{\Delta(f)_1 + \Delta(f)_2}{2}, f(D) = \frac{|\Delta(f)_1 - \Delta(f)_2|}{2}. \tag{2.15}$$

Low frequency pulsed radar and high frequency FMCW radar occupy different areas within drone surveillance solution space. Higher range resolution is available using FMCW radar, at the expense of the maximum detection range which is limited



(a)



(b)

Figure 2.6: Transmitted and received waves when Frequency Modulated Continuous Wave (FMCW) radar is used to detect a moving target, a) Sawtooth modulation b) Triangular modulation.

to a few hundred metres, whereas long-range detection can be achieved using low frequency pulsed radar. Furthermore, the carrier frequency sets the magnitude of the Doppler shift [88][69]. It is possible that finer Doppler detail could lead to improved drone classification, however, it could also place limits on the coherency period causing more splitting across Doppler bins and a lower SNR. In this thesis, drone classification will be explored using both an L-band pulsed staring radar (Chapter 5 and Chapter 7) and a Ku-band FMCW staring radar prototype (Chapter 6).

Data used in Chapter 6 was collected by PlexTek [89], using their FMCW staring radar prototype with a single transmit element and eight receive elements. It

uses triangular modulation, with 6617 cycles per second. Beamforming, range, and Doppler profiling is performed, and a range-Doppler map is used for target detection.

2.6 Staring Radar

The nature of drone flight poses a number of detection challenges [90][91] for conventional scanning radar. Firstly, drones have a relatively low RCS meaning that they are often difficult to detect above noise or clutter. Secondly, drones fly at relatively low altitudes and at low velocities (in the order of 20 m/s [92]), meaning that the returns are often obscured by multi-path ground returns or by stationary or slow-moving clutter, particularly when hovering. SNR and Doppler resolution can be increased by increasing the CPI, allowing the targets to be distinguished from stationary clutter. In the case of scanning radar, a long CPI would mean longer re-visit times between each scan, or poorer angular resolution. Since drones are capable of being flown on irregular trajectories and carrying out complex manoeuvres, long re-visit times increases the risk of missed detections and dropped tracks.

Skolnik introduced the concept of a radar that is capable of staring in all directions simultaneously, coining the term "ubiquitous radar" [93][94]. Consequently, the integration period can be arbitrarily long provided the target remains within the surveillance volume, increasing both Doppler resolution and the maximum detection range [95]. Therefore, it is possible to detect drone blades and micro-Doppler signatures using staring radar, which could potentially aid in the classification of drones and their discrimination from birds [37][96][97]. Furthermore, a high update rate and multiple receiver beams allow multiple targets to be tracked even when following irregular trajectories, making staring radar ideal for the detection, tracking, and classification of drones [98][99].

The data used in Chapter 5 and Chapter 7 were collected using an L-band staring radar developed by Aveillant [100] for the detection and tracking of drones. The transmitter is a stack of eight elements that transmit a wide beam pulse covering the entire surveillance volume, whilst the receiver is a phased array of 4x16 elements where each element is separated by half a wavelength. Synthetic receiver beams are formed by processing the signal received from each antenna element and accounting for the phase shift resulting from the differing path lengths between the target and each element. This generates numerous receiver beams that cover the entire search volume simultaneously. This concept is demonstrated in Fig. 2.7 [6].

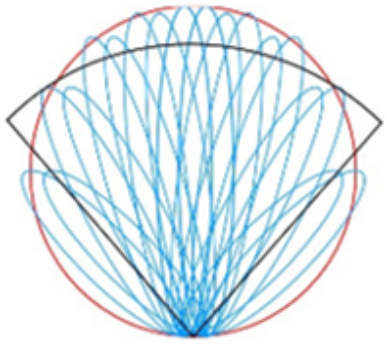


Figure 2.7: A diagram showing the transmit (red) and receive (blue) beams of a staring radar [6].

The operating parameters of the Aveillant Gamekeeper 16U radar are provided in Table 2.1. The transmit and receive arrays have considerably more elements in the vertical plane than in the horizontal plane, with receive beam widths of approximately 6° and 25°, respectively. The finer resolution in elevation helps with distinguishing drones from ground targets, and reduces the effects of clutter. Medium sized drones can comfortably be detected out to a range of 5 km, with a maximum range of approximately 10 km.

The high PRF allows for flexibility in the number of pulses integrated over, with the potential for coherent integration over many pulses resulting in high Doppler resolution and a gain in SNR. Integrating over 2048 pulses leads to a gain of up to

Table 2.1: The operating parameters of the Aveillant Gamekeeper 16U staring radar used for data collection.

Parameter	Value
Frequency	L-band
Bandwidth	2 MHz
Peak Transmit Power	2 kW
Transmit Array	1 x 8
Receiver Array	4 x 16
Azimuth Coverage	90°
Elevation Coverage	30°
Pulse Repetition Frequency	~7.5 kHz

33 dB in SNR, provided the target remains coherent within a time window of 0.28 seconds.

2.7 Spectrogram Formation

A 4-D matrix is formed, where a complex signal is recorded in a resolution cell (in range, azimuth, and elevation) for each transmitted pulse. Upon beamforming, the azimuth and elevation dimensions are combined into a single receiver channel, producing a data cube (Fig. 2.8). For opportune targets such as birds, the output of the tracker (applied to the detections) is used to identify the resolution cell containing a particular target. In the case of drones, the onboard GPS is used to identify the beam and range cell that a particular drone resides in at a given moment in time. Data from each range and channel containing the target is then flattened into a 1-D vector known as a time series, containing the signal returned from a particular target (and all other targets within the same resolution cell) at each point along its track.

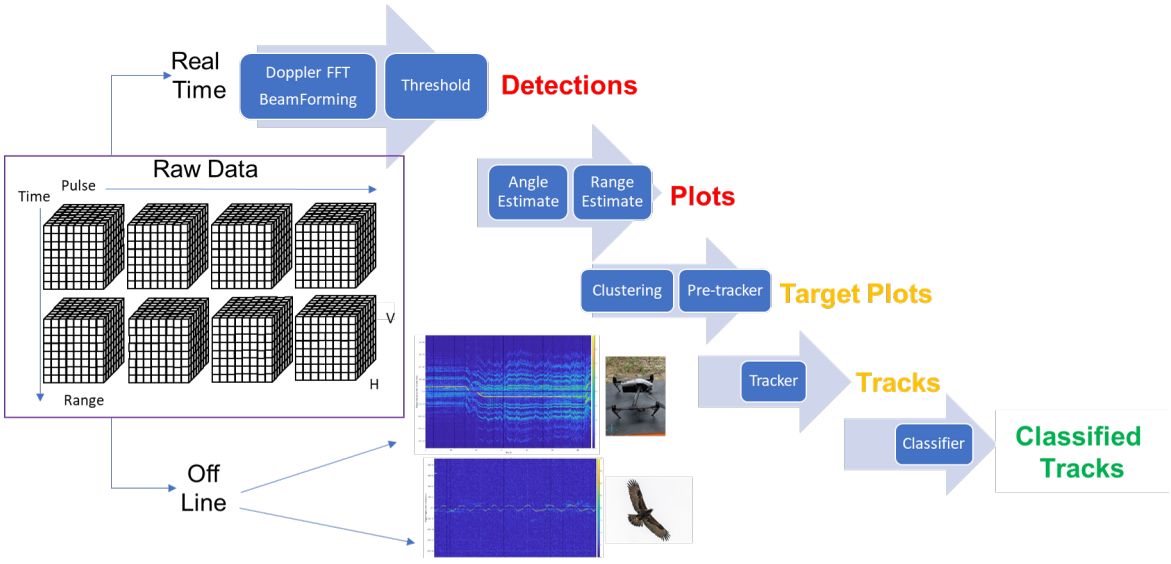


Figure 2.8: A diagram showing the preprocessing steps of the L-band pulsed staring radar. A 4-D data matrix is formed, from which targets are detected and tracked. The tracker output is then used to identify the resolution cell containing a particular target at each time step, and a time series vector is extracted and used to plot a spectrogram.

The time series data is reshaped into frames of N_{FFT} pulses, where an FFT of length N_{FFT} is taken across each frame, transforming the data into the frequency domain. T spectra are stacked together along the time axis to produce a spectrogram, where the complex signal is converted to a power in decibels. This process is shown in Fig. 2.9.

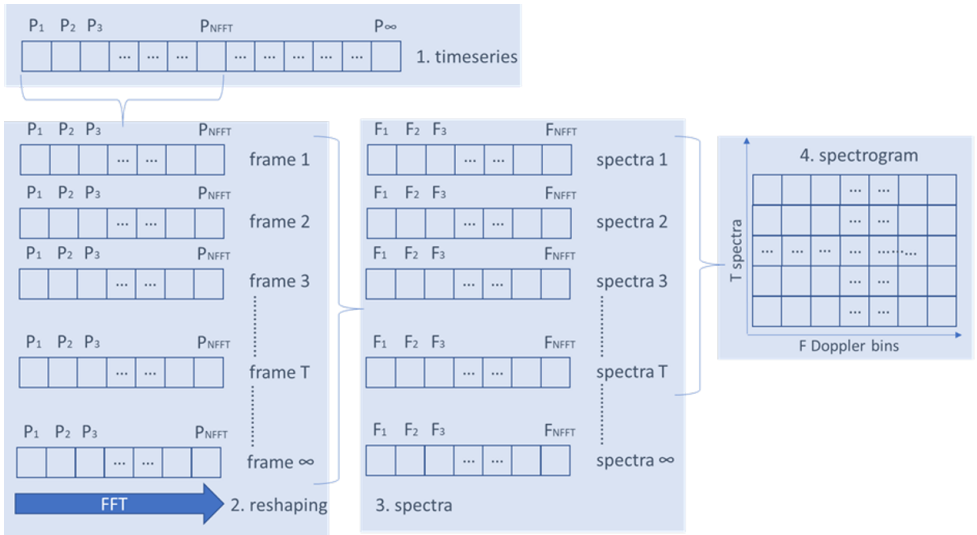


Figure 2.9: A diagram showing the process of producing a spectrogram from time series data.

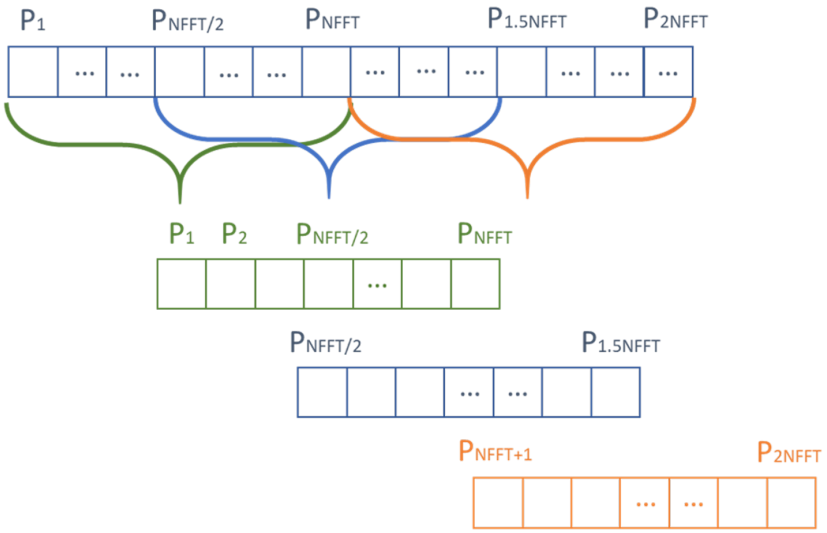


Figure 2.10: A diagram illustrating the concept of overlap in spectrogram generation.

The time resolution of a spectrogram is determined by the number of pulses integrated over. That is, an event that took place in the first pulse within a frame is in the same spectra as an event that took place within pulse number N_{FFT}. Overlapping consecutive frames before applying the FFT (as shown in Fig. 2.10) allows shorter-lived time events to be visible in the spectrogram, smoothing the appearance in the time domain whilst maintaining high frequency resolution.

Fig. 2.11 shows an example spectrogram from the Aveillant L-band radar discussed in Section 2.6, where an FFT length of 4096 was used with 50% overlap between frames, making the total integration time approximately 0.28 seconds and the Doppler resolution 1.8 Hz (velocity resolution of 0.23 m/s). The spectrogram corresponds to a time duration of approximately 5.5 seconds.

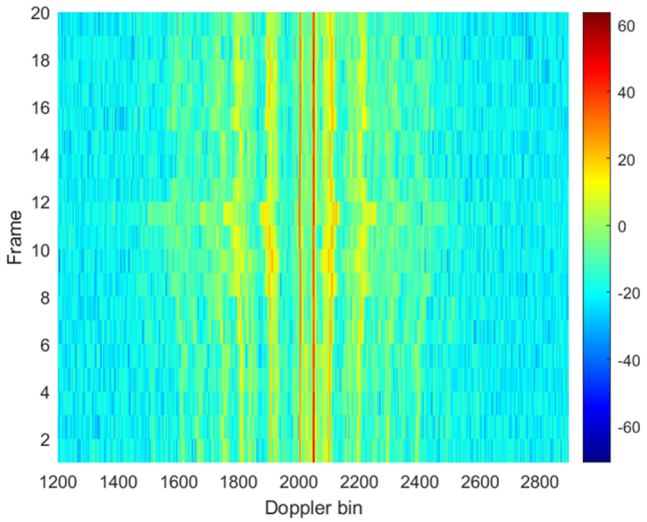


Figure 2.11: An example drone spectrogram produced using a 4096-point Fast Fourier Transform, with 50% overlap between frames and a Blackman-Harris window applied.

Chapter 3

Machine Learning Foundations

3.1 Introduction

This thesis uses convolutional neural networks to classify drones based on their radar spectrograms. More traditional machine learning algorithms are also trained in order to better understand the characteristic features of drone and bird targets, to investigate performance limitations in an explainable manner, and to provide a performance benchmark for classification success. This chapter covers the machine learning and deep learning theory relevant to this thesis. The basic principles of machine learning are presented, along with the metrics for evaluating performance. The logistic neuron and perceptron are introduced, forming the building blocks of deep neural networks. Finally, CNNs are outlined, with an explanation of their key features and how they may be used to classify drone spectrograms.

3.2 Machine Learning Concepts

Classifiers aim to determine the characteristic features of a target – robust features that may be used to differentiate between classes. Each feature extracted from the

training data is considered to be a variable, and the collection of n features forms an n-dimensional 'feature space'. Combinations of these features are mapped onto labels, minimising a loss function that measures the difference between the predicted label and the expected (correct) output. The mapping learnt during the training phase is then used to classify unseen targets in the test phase.

3.2.1 Sources of Error, Bias, and Variance

Natural variation between targets, environmental effects, measurement errors, and incomplete domain coverage are sources of error arising from a small data set that is not representative of the conditions under which the classifier will be implemented. In the context of drone and bird classification, natural variation between targets is to be expected due to the differences in shape, size, and flight styles of different drone models and bird species.

On the other hand, the classification algorithm itself is also a source of error. There are many different types of classification algorithms to choose from, and it is important to carefully select a classifier and tune the associated hyperparameters to mitigate prediction errors. Bias and variance are two types of prediction error prevalent in all types of classifier. Bias is the difference between the predicted and correct labels caused by an imperfect model or lack of training data, leading to high error when training and testing. High variance occurs when the model learns noisy features that are present in the training data but are not representative of the entire class, leading to poor generalisation to unseen test data. High bias is caused by underfitting to the training data, failing to learn the discriminative features, whereas high variance is caused by overfitting on noise. When considering model complexity, there is a balance to strike between bias and variance.

3.3 Machine Learning Algorithms

In Chapter 4, it is demonstrated that machine learning classifiers can be trained to distinguish between drones and non-drones, and to classify drone groups or models. Classifiers are trained on handcrafted features, from trajectory-based features such as velocity and acceleration, to micro-Doppler features that are characteristic of drone motion. Commonly used models include Naïve Bayes classifiers, discriminant analysis models, support vector machines, k-nearest neighbour algorithms, decision trees, and random forests (Table 4.1).

3.3.1 Naïve Bayes

Naïve Bayes classifiers are probabilistic, using Bayes' theorem to classify the test data. A Gaussian Naïve Bayes classifier fits a normal distribution to each variable within a given class, using the mean and standard deviation of the training data. Fig. 3.1 shows an example set of training data for a two-class problem with two features, where probability distributions are plotted for each feature and class. Bayes' theorem can then be used to determine the probability that a test target belongs to each class C_k, given that it occupies a particular point $\mathbf{x}$ in feature space,

$$p(C_k|\mathbf{x}_i) \propto p(\mathbf{x}_i|C_k)p(C_k).$$

The target is then assigned to the class with the highest posterior probability. Note that the Naïve Bayes algorithm assumes that all features are statistically independent, and normally distributed.

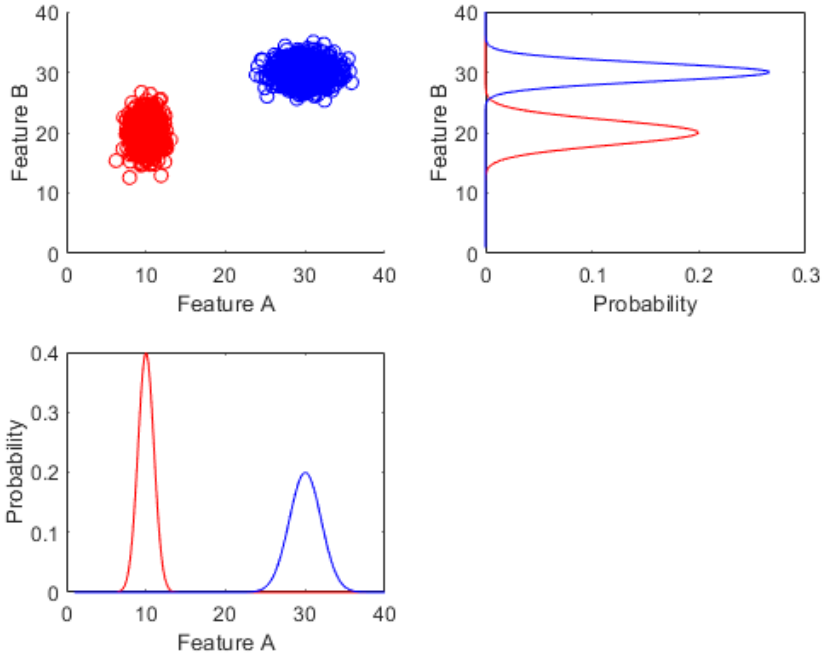


Figure 3.1: A training data set for a two class problem (red and blue), with features A and B sampled from normal distributions.

3.3.2 Discriminant Analysis

Discriminant analysis algorithms are another form of probabilistic classifier. Similarly to a Gaussian Naïve Bayes classifier, a linear discriminant analysis model fits a normal distribution to the training data, and uses Bayes' Theorem to maximise the posterior probability. However, independence between features is not assumed, and the mean and covariance matrix are used to fit a normal distribution to the data. Boundaries are then drawn between classes, defining the set of points along which there is an equal probability of a target belonging to each class. In the case where the covariance matrix is diagonal, the classifier is essentially equivalent to the Gaussian Naïve Bayes model. When the covariance matrix is identical for each class, a linear boundary is sufficient, else quadratic boundaries are required, as shown in Fig. 3.2.

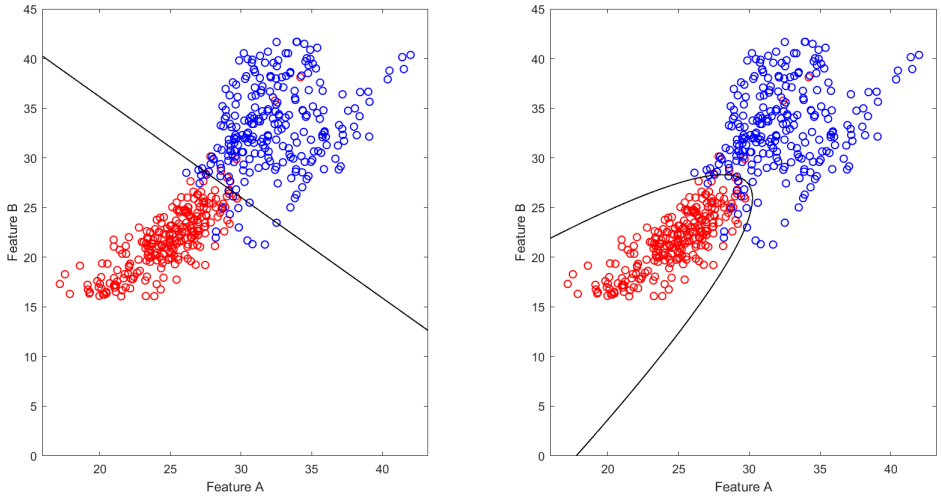


Figure 3.2: A training data set for a two class problem (red, blue), with two feature vectors, showing the linear (left) and quadratic (right) boundary between the classes when a discriminant analysis model is used.

3.3.3 Support Vector Machine

Support vector machine (SVM) models separate classes with a hyperplane, maximising the separation between the hyperplane and the points closest to it. In 2-D, the hyperplane is a line that maximises the separation between two classes, as shown in Fig. 3.3.

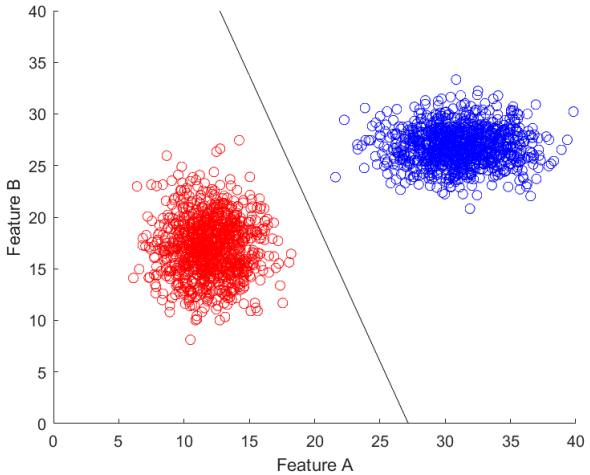


Figure 3.3: A training data set for a two class problem (red, blue), with two feature vectors, showing the hyperplane that maximises the margin between the two classes.

The model uses a linear boundary, however, different kernel functions can be used to transform the data into higher dimensions, allowing distinction between classes that are not linearly separable. SVMs are designed for two-class problems, although multiple classes can be considered by training multiple classifiers each using a one-vs-rest approach.

3.3.4 k-Nearest Neighbours

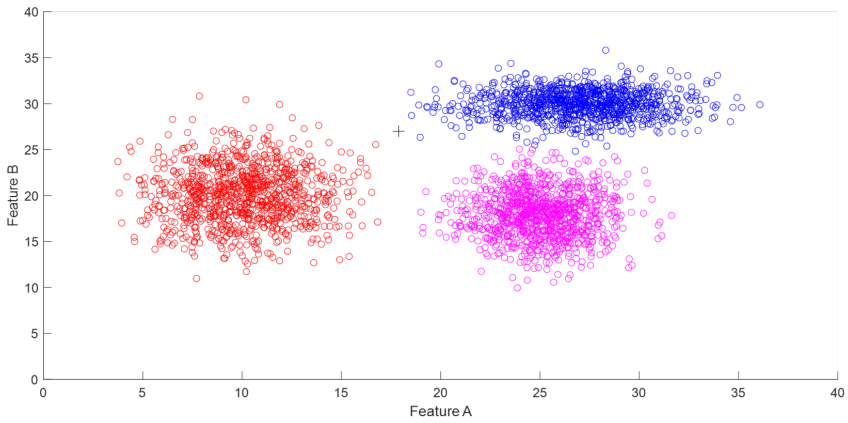


Figure 3.4: A training data set for a three class problem (red, blue, and magenta), with a test sample (black '+') that would be assigned to the blue class using a k-nearest neighbours model with $k = 3$.

K-nearest neighbour (kNN) algorithms classify test samples based on proximity in feature space to points in the training data. The distance between the test target and each training target is evaluated, and the k closest training targets are selected. In this work, the Euclidean distance is the chosen distance metric. Voting is then used to classify the test target, assigning it the label possessed by the majority of the k closest training samples. Fig. 3.4 illustrates this for a two-dimensional three-class problem, where a black test target has two nearest neighbours that belong to the blue class and one nearest neighbour belonging to the red class. Therefore, if $k = 3$, the test target would be assigned to the blue class. The k-NN algorithm

makes no assumptions about the underlying data distribution, unlike many other machine-learning models.

3.3.5 Decision Tree

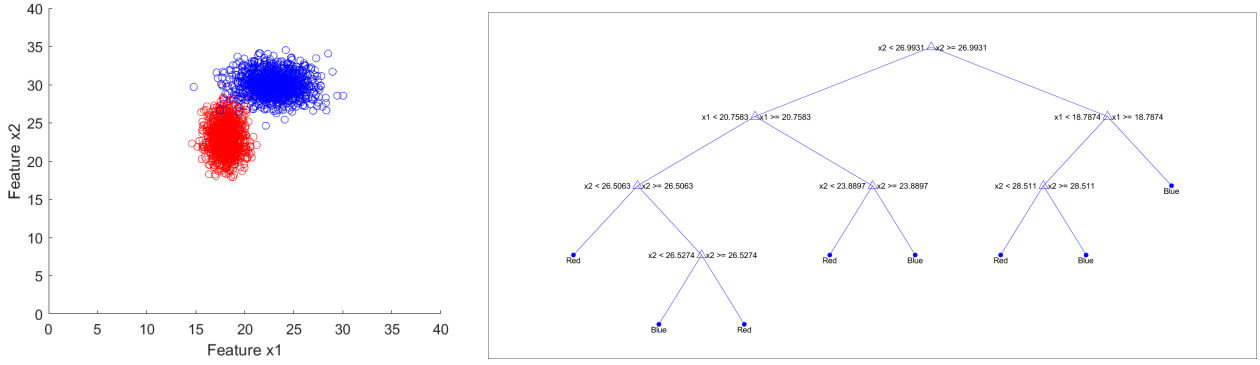
Decision tree classifiers are flow charts that can be used for classification or regression. A decision tree is a series of nodes at which a feature is compared to some threshold, and the path along the branches of the tree is determined by the outcome of the test, as shown in Fig. 3.5. The leaf node at the end of the tree determines the predicted class. The features tested at each node, and the corresponding thresholds, are learned during the training phase. The Gini Diversity Index (GDI), G, is a split criterion that is commonly used to minimise the impurity at each node following a split.

$$G = 1 - \sum_{i=0}^{i=k} (P_i)^2,$$

where P_i is the probability of belonging to class i following a split, and a pure node would have a GDI equal to zero. Predictor importance is estimated by summing changes in the weighted impurity due to splits on each feature, and dividing by the number of nodes [101]. Whilst decision trees are generally interpretable, they are prone to overfitting as trees can become very large, resulting in poor generalisation to unseen test data if tree depth is not regulated.

3.3.6 Random Forest

Random forests are ensemble learning classifiers, where majority voting between a collection of decision trees determines the output. This approach decreases the model variance, provided the individual trees are uncorrelated and each performs better



(a) (b)

Figure 3.5: An example decision tree trained to classify a target into category 'red' or 'blue', using two features x1 and x2 a) Training data feature space b) Trained decision tree.

than random guessing. To avoid correlation, trees are trained on random samples of the training data, in a process known as bagging. Feature bagging is also used, whereby trees are trained on a random subset of the full set of features. Random forest algorithms are generally high performing as the ensemble of weak predictors prevents overfitting on the training data, however, there is a loss of interpretability compared to decision trees.

3.4 Deep Learning Concepts

When training a machine learning classifier, there are only a relatively small number of parameters to tune, making them fast to train with low computational cost. In addition, the models are generally interpretable, as features used for training are manually selected. However, choosing discriminative and robust training features requires an in-depth understanding of feature space. Deep learning algorithms are capable of learning the properties of each class directly from the training data, therefore do not require manual selection of features, although a large amount of labelled training data is required for high performance.

Even simple classifiers can achieve very high accuracies provided the classes are clearly separable in feature space. Due to the similarities between drones and birds from a radar perspective (Chapter 2), this is unlikely to be the case. Convolutional neural networks can extract deep non-linear features directly from the data, drawing non-linear boundaries across many dimensions, and potentially improving classification performance.

Deep learning classifiers including CNNs have been used to classify radar data in the fields of automotive radar [50][102], human activity recognition [103][73][104], person identification [105], and gesture recognition [106]. It is therefore a natural extension to consider CNNs for classification of drones using radar.

3.4.1 Artificial Neural Networks

Neurons in the human brain are connected, and when the first neuron in a chain is activated by an electrochemical input, other relevant neurons are activated in response, propagating a signal through the network. The strength of the connection between neurons is increased through learning and exposure to new information. Artificial neural networks (ANNs) emulate biological networks of neurons, with many connected neurons organised in layers. ANNs learn by adjusting the weights between connected neurons, with the goal of reducing the overall network error.

The McCulloch-Pitts neuron [107], based on biological neurons, forms the basis of modern ANNs. The neuron accepts a binary input vector and returns a binary output depending on whether the sum of the inputs exceeds a predetermined threshold.

The perceptron is an extension of the McCulloch-Pitts neuron that is not re-

stricted to binary inputs, and each input x is weighted by some value w. If the weighted sum of the inputs exceeds a predetermined threshold, the perceptron activates,

$$0 \iff \sum_j^N w_j x_j + b < 0,$$
$$1 \iff \sum_j^N w_j x_j + b > 0,$$

where b controls the barrier for activation.

Neurons are an extension of the perceptron where the output can take a range of values, indicating the strength of an activation. The activation function determines the output from a neuron, and is chosen such that incremental changes to the network weights will lead to small changes in the output of the network, to allow learning. The activation function σ acts on the weighted sum of the input $z = \sum_j^N w_j x_j + b$ to produce an activation, as illustrated in Fig. 3.6.

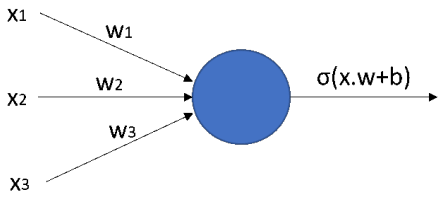


Figure 3.6: An example of a neuron with three inputs x_i, weights w_i, bias b, and an output that is determined by the activation function σ.

Activation Functions

The form of the activation function is chosen before training. Ideal activation functions are non-linear, with a non-zero gradient for all values to avoid saturation during training. Commonly used activation functions include the sigmoid function, tanh, and rectified linear unit (ReLU) (Fig. 3.7). Sigmoid and tanh activations have a

steep gradient in the centre, pushing the output towards the two extremes and assisting with class differentiation. However, sigmoid and tanh functions suffer from vanishing gradients at the extremes, therefore ReLU functions are commonly used in the middle layers of the network, as the gradient is dependent on the argument.

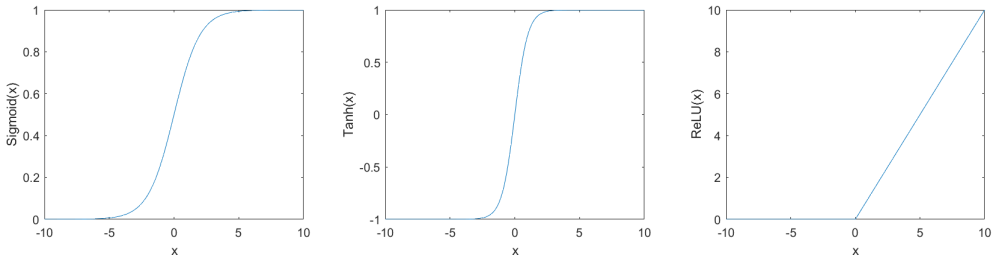


Figure 3.7: Sigmoid, tanh and rectified linear unit (ReLU) activation functions.

3.4.2 Training a Neural Network

Neural networks are trained by making small adjustments to the weights and biases in the network to decrease the overall network error. This is an optimisation problem, where the aim is to minimise a loss function that describes the network error on the training data. A backpropagation algorithm is applied to determine the gradient of the loss function with respect to a particular weight within the network, and gradient descent algorithms are used to adjust the weight.

Loss Function

A loss function calculates the difference between the network output and the correct output – that is, the true class that the training sample belongs to. The cross-entropy loss is a common choice of loss function. The output layer is passed through a softmax activation function that normalises the input vector $\mathbf{x}$ and outputs a probability that can be interpreted as a confidence in the class label. The softmax

function is defined as

$$\sigma(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}. \tag{3.1}$$

The loss function, L, is defined as

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K} T_{ni} log(Y_{ni}), \tag{3.2}$$

where N is the number of samples over which the loss is calculated, K is the number of classes, and Y_i and T_i are the predicted and correct outputs respectively. The minus sign in Eq. (3.2) ensures that the loss is positive overall, as the logarithm function results in negative values for predictions that are less than 1. The aim of training is to minimise the loss function by correctly classifying samples with a high degree of confidence.

Gradient Descent Algorithm and Backpropagation

The gradient descent algorithm minimises a function by taking steps in the opposite direction to the gradient of the loss function L. The network weights $\mathbf{w}$ and biases $\mathbf{b}$ are updated as follows:

$$\mathbf{w} = \mathbf{w} - \eta\frac{\partial L}{\partial \mathbf{w}} \tag{3.3}$$

$$\mathbf{b} = \mathbf{b} - \eta\frac{\partial L}{\partial \mathbf{b}}, \tag{3.4}$$

where the learning rate η determines the size of the steps. If the learning rate is too small then the network will take a long time to converge to a solution, and local minima may be found rather than an optimal solution. On the other hand, if the learning rate is too large, the network may diverge and be unable to find a minimum. A common compromise is to start with a high learning rate that decreases during training, in a process known as learning rate decay.

Deep learning algorithms require large amounts of training data to achieve high accuracies, therefore the data often has to be passed to the network in batches to reduce memory requirements. Minibatch gradient descent updates the network parameters after estimating the loss function using all samples within a batch. Batch size is a tuneable hyperparameter, and the choice of batch size affects the speed and stability of training. An iteration is the number of times that the network has been updated during training, and an epoch passes once the entire training data set has been processed by the model.

A momentum term is often added to the gradient descent equations. The momentum is a moving average of the gradients in previous steps, and adding this term increases robustness to noise and guards against getting trapped in local minima. This is particularly effective when using smaller minibatch sizes.

Gradient descent requires the gradient of the loss function to be computed with respect to all of the network weights and biases. This is achieved using the back-propagation algorithm, which is comprised of four fundamental equations derived in [108],

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \tag{3.5}$$

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \tag{3.6}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j \tag{3.7}$$

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j, \tag{3.8}$$

where w^l_{jk} is the strength of the connection between the k^{th} neuron in the $l-1^{th}$ layer, and the j^{th} neuron in the l^{th} layer, L is the total number of network layers, and C is the loss function. As the gradient determines the amount by which weights are updated and is calculated using the chain rule, weights in the earlier layers of

the network can change very slowly, preventing convergence. This is known as the vanishing gradient problem, and can limit network depth.

The network takes labelled training data as an input and feeds it through the network, producing an output. The loss function is calculated and the error is back-propagated through the network (Eq. (3.5) and Eq. (3.6)). The gradient of the cost function with respect to particular weights (Eq. (3.7)) and biases (Eq. (3.8)) within a layer are calculated and used to update the weights via Eq. (3.3) and Eq. (3.3). This process is repeated until the network converges.

3.4.3 Convolutional Neural Networks

CNNs [109] are deep neural networks based on the visual cortex [110], designed to classify data arranged in grids. They can learn directly from labelled images, applying filters at each region of the image to learn the properties of each class. The learned mapping is then applied to classify unseen test images.

Network Layers

In this thesis CNNs are trained to classify radar spectrograms, as spectrograms contain a wealth of features suitable for classifying UAVs (Section 1.2, Chapter 4). The CNN takes a spectrogram as an input, treating it as a matrix of pixels. Colour images are treated as a 3-D input, with separate RGB channels. CNNs consist of an input layer, hidden learning layers containing at least one convolutional layer, and classification layers (Fig. 3.8). Outputs from the previous layer become the input to the next layer.

In convolutional layers, kernels of learned weights are convolved with the input.

A kernel is overlaid on the input image and is moved across it, taking the dot product between the sampled input and the kernel, building a feature map [7]. Typically, a convolutional layer consists of many kernels, each of which act as a filter, extracting features from the input. Simple features (such as colours and edges) are extracted in shallow layers and are mapped onto increasingly complex features as network depth increases.

To increase generalisation, neurons are probabilistically switched off during training in a process known as dropout. Randomly selected neurons are prevented from contributing to the network, changing the network architecture for each input and reducing over-reliance on a particular feature [111][112].

Pooling layers are used to down-sample the output from convolutional layers, reducing the number of dimensions and thus the computational demand. Pooling also prevents overfitting by introducing some regularisation. Pooling considers a small patch of the feature map and takes either the average or the maximum value of the patch, depending on the type of pooling used.

In fully connected layers, each neuron is connected to every neuron in the next layer. Fully connected layers typically precede the classification layer, for learning non-linear combinations of the features extracted in the convolutional layers. A softmax activation function then returns the probability of belonging to each class. The output layer consists of a single neuron for each class.

CNNs are particularly suited for image classification as weight sharing ensures that all areas of an image are processed using the same set of transformations. This increases generalisation, making the classifier invariant to shifts in position [110]. Parameter sharing leads to lower memory requirements and higher efficiency, as fewer operations are required. A drawback of CNNs is the lack of interpretability

– CNNs have millions of weights and connections, and are often described as black boxes.

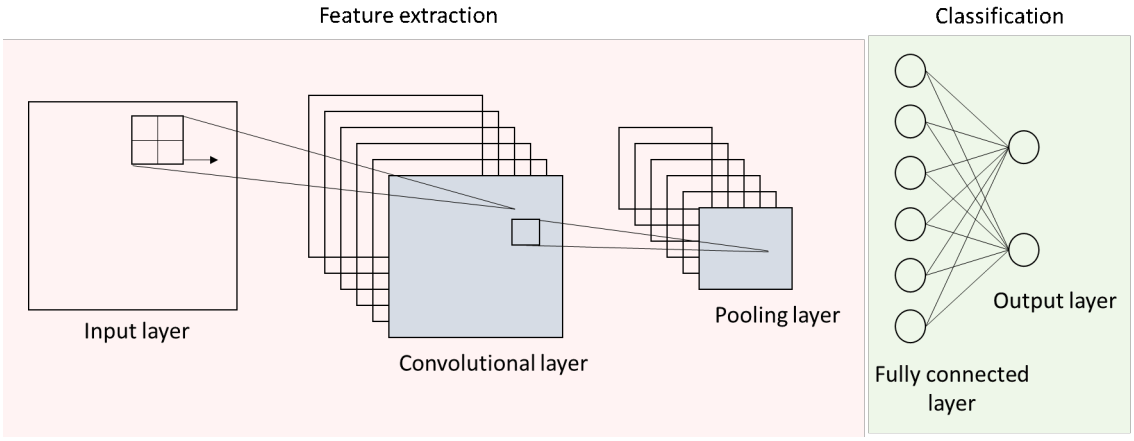


Figure 3.8: The typical structure of a convolutional neural network with a single convolutional layer.

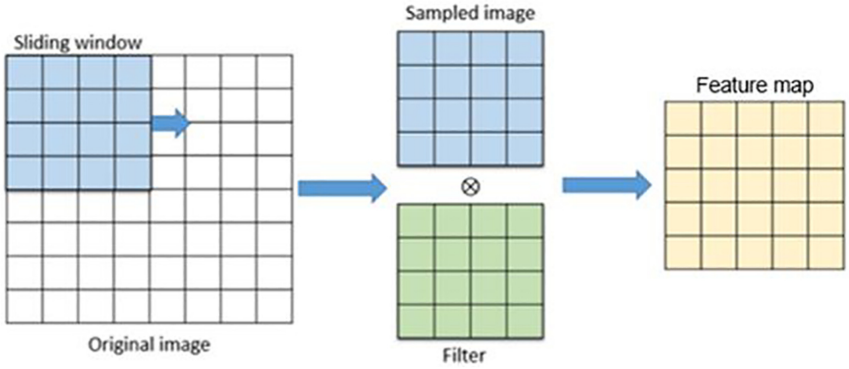


Figure 3.9: A convolution between an image and a learned filter. The dot product between the sampled image and the kernel produces a single element within the feature map [7].

Transfer Learning

Due to the high number of tuneable parameters, CNNs require large amounts of data for training. Transfer learning sets the network weights and biases using well-established pre-trained networks, rather than using random initialisation. Transfer learning has been used alongside radar data, where fine tuning the parameters of CNNs trained on optical data has been shown to reduce the training time and improve performance on small data sets compared to training a network from scratch

[113][103][114]. Transfer learning has the added benefit that there is no trial and error required to find a suitable network architecture. Transfer learning is used in this thesis, retraining CNNs that are well established in the field of computer vision, for the classification of drone spectrograms.

Network Architectures for Transfer Learning

The ImageNet project [115] is a large database of labelled training images used for research in the field of computer vision. The ImageNet Large Scale Visual Recognition Competition (ILSVRC) [116] is a contest for classification algorithms, and a selection of the highest performing CNNs from this competition are considered in this thesis for differentiating between drone and bird spectrograms.

AlexNet [8] won the ILSVRC in 2012 by a significant margin. AlexNet is based on LeNet-5 [109], a pioneering CNN designed in 1998 for the classification of handwritten digits. Advancements in computing allow AlexNet to be deeper than its predecessor, with many more filters. AlexNet (Fig. 3.10[8]) consists of five convolutional layers, three fully connected layers, and max pooling layers, and uses ReLU activations. Dropout and data augmentation are used to reduce overfitting. AlexNet is often credited with inspiring others to use CNNs for computer vision [117].

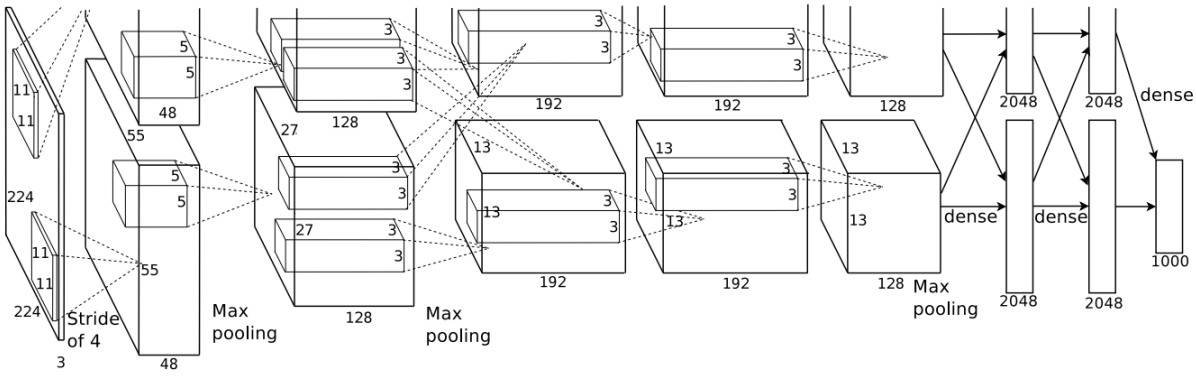


Figure 3.10: AlexNet architecture [8].

SqueezeNet was designed as a more compact replacement for AlexNet, achieving comparable accuracy with 50× fewer parameters, leading to faster training and a model size 510× smaller than that of AlexNet [9]. SqueezeNet uses 1x1 convolutions to decrease the number of channels in 'squeeze' layers, with corresponding 'expand' layers containing a mixture of 1x1 and 3x3 kernels. Squeeze and expand layers together form a 'fire module' (Fig. 3.11[9]).

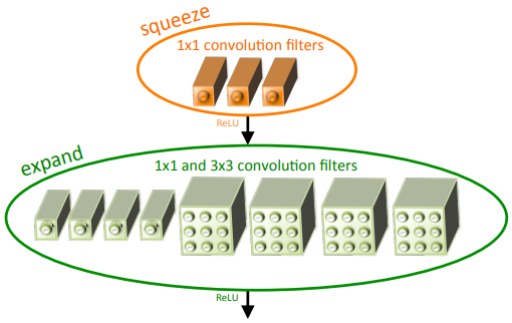


Figure 3.11: SqueezeNet fire module [9].

GoogLeNet [10] won the ILSVRC competition in 2014, achieving near human-level performance. GoogLeNet has 22 layers, and uses an inception module (Fig. 3.12 [10]) that adds network depth, extracting features at varying scales. The distribution of features within the input determines the appropriate kernel size; large kernels are ideal for extracting global features, whilst smaller kernels are needed to extract localised features. The inception module carries out four operations in parallel: a 1x1 convolution, a 3x3 convolution, a 5x5 convolution, and a 3x3 pooling. 1x1 convolutions are used to reduce the number of channels, preventing overfitting and reducing the time taken to train the network. GoogLeNet contains three softmax classifiers, two of which are between convolutional layers. The final loss function is the weighted sum across all three classifiers, to avoid the problem of vanishing gradients.

Inception v3 [118] is based on GoogLeNet, and also uses inception modules and auxiliary classifiers. Convolutions are factorised to increase efficiency (e.g. 5x5

convolutions are replaced by two 3x3 convolutions, and nxn convolutions are replaced by 1xn and nx1 convolutions), and inception modules are made wider. Regularising components are added to the loss function, and a RMSProp optimiser is used rather than stochastic gradient descent.

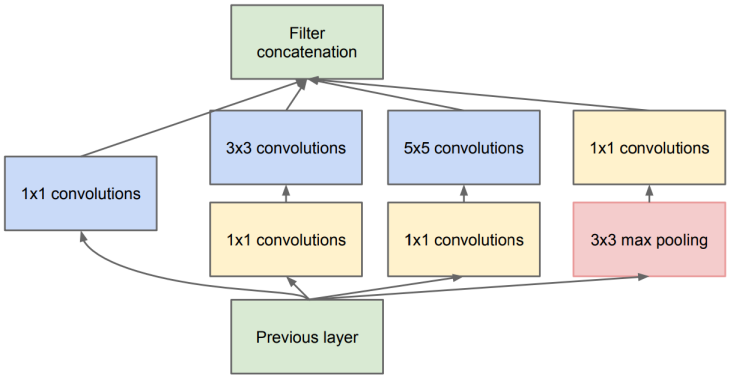


Figure 3.12: Inception module [10].

The residual neural network (ResNet) [11] uses residual blocks (Fig. 3.13 [11]) to increase network depth whilst avoiding the vanishing gradient problem, producing an architecture whose performance increases with depth. Residual blocks contain shortcut connections that skip the convolutional layers, performing identity mappings, whilst convolutional layers learn a residual mapping, the output of which is added to the input to the block. ResNet-18 and ResNet-50 are versions of ResNet architectures with 18 and 50 layers, respectively.

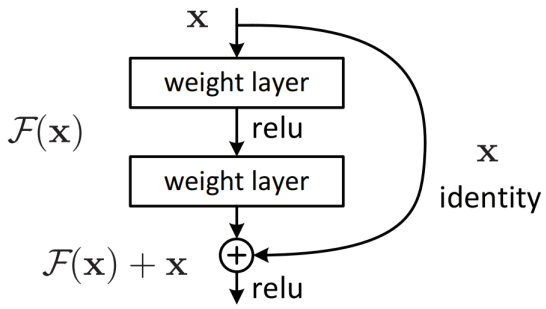


Figure 3.13: Residual block [11].

Network depth and the number of parameters both have implications on the

complexity of the features that can be extracted, as well as the speed of training and likelihood of overfitting. The CNNs listed here inevitably represent just a subset of all possible CNN architectures, and each model will have its own strengths and weaknesses. In addition to classification performance, computational complexity and memory requirements are also metrics to consider when making a choice of classifier. Computer Vision is a popular area of research and can be expected to lead to the development of increasingly accurate and efficient CNNs. Of course, the networks discussed here are optimised for the classification of optical images, whereas better performance may be achieved using a custom network designed for the classification of spectrograms, although this lies beyond the scope of this thesis.

3.5 Performance Evaluation

Performance metrics are needed in order to quantify the capabilities of a classifier, to identify limitations, and to make comparisons. The data is split into training, validation, and test subsets for performance evaluation. The classifier is tested on the validation set during the learning phase, and training is stopped once performance plateaus or drops, to avoid overfitting. Performance is evaluated on the test data, to ensure that the classifier is robust enough to generalise to data that was unseen during training.

Generalisation is a key feature of drone classifiers, as drones come in many different shapes and sizes, and are capable of carrying out a wide range of manoeuvres. It is not possible to include every representation of drone or non-drone in the training data, yet it is vital that the model can recognise 'out of distribution' targets as belonging to the correct class.

For a binary problem, one class is selected as the positive class, whilst the other is negative. In the case of drone classification, the drone class is positive. This allows the calculation of a number of metrics:

True positives (TP): The count of instances where the test sample is correctly classified as belonging to the positive class.

True negatives (TN): The count of instances where the test sample is correctly classified as belonging to the negative class.

False positives (FP): The count of instances where the test sample is incorrectly classified as belonging to the positive class. A misclassification of this kind is known as a Type I error.

False negatives (FN): The count of instances where the test sample is incorrectly classified as belonging to the negative class. A misclassification of this kind is known as a Type II error.

Recall: Expresses how often the prediction is correct when the test sample belongs to the positive class,

$$\frac{TP}{FN + TP}.$$

Precision: Expresses how often the prediction is correct when a test example has been assigned to the positive class,

$$\frac{TP}{TP + FP}.$$

Classification accuracy : The percentage of correct predictions,

$$\frac{TP + TN}{TP + TN + FP + FN}.$$

Accuracy is conceptually simple, but assumes an equal cost of misclassification for both classes. In the case of drone classification, the potential consequences of failing to identify a malicious drone are more severe than the consequences of raising a false alarm, therefore a type I error is preferable to a type II error. Accuracy can be particularly misleading when the classes are imbalanced: if 99% of training and test data belongs to the non-drone class, a classifier could simply label all targets as non-drones and achieve an accuracy of 99%, whilst being operationally useless.

Instead of accuracy, the F1 score is often used to evaluate model performance. The F1 score is the harmonic mean of the precision and recall,

$$\frac{2 * precision * recall}{precision + recall}.$$

Confusion matrices are a method of calculating and displaying a selection of performance metrics. Fig. 3.14 shows an example confusion matrix for a binary drone-vs-bird classification problem.

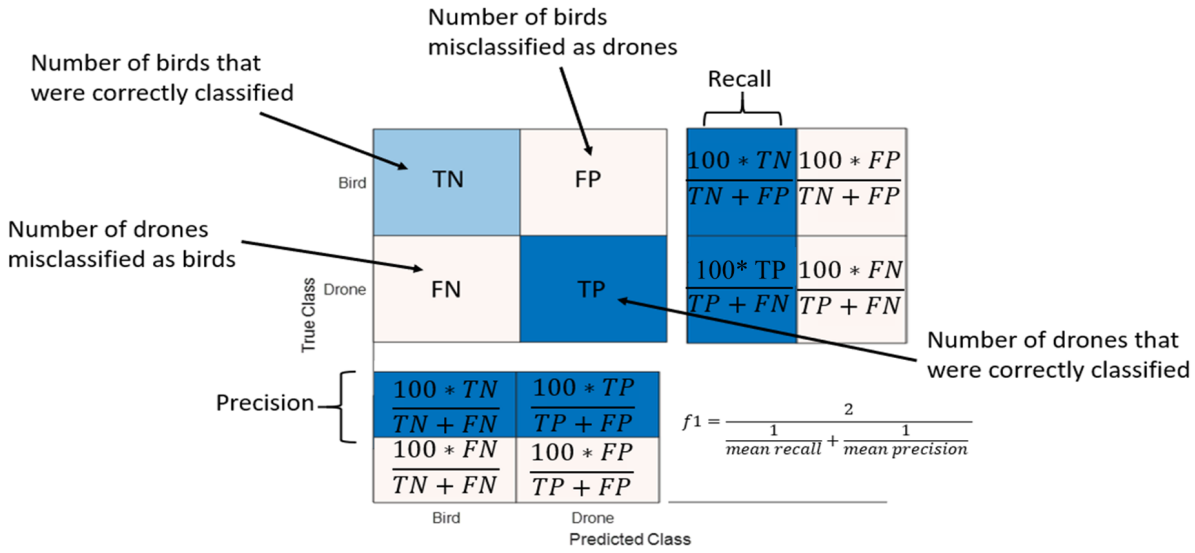


Figure 3.14: An annotated confusion matrix. Recall and precision are calculated using the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) metrics.

A drone vs non-drone classifier must have a high sensitivity in order to identify drones, but also a high precision such that the output is reliable. Low precision would mean that the drone surveillance system becomes overwhelmed by false alarms, rendering it ineffective. The output of a classifier is a numerical value between 0 and 1. If the output exceeds a predetermined threshold, the target is given the label that corresponds to the positive class. The threshold value can be tuned to adjust the sensitivity of the classifier. Decreasing the threshold increases the true positive rate, but also increases the number of false positives. Receiver operator characteristic (ROC) curves (Fig. 3.15) plot the true positive rate (TPR) against the false positive rate (FPR), demonstrating the relationship between recall and precision. The ROC curve of a random classifier would be a straight line, whilst the ROC curve of a perfect classifier would be a single point where the TPR is 1 and the FPR is 0. The shape of a ROC curve is indicative of performance, and the point closest to the top left-hand corner maximises the TPR whilst minimising the FPR, allowing the operator to select the threshold value that maximises performance.

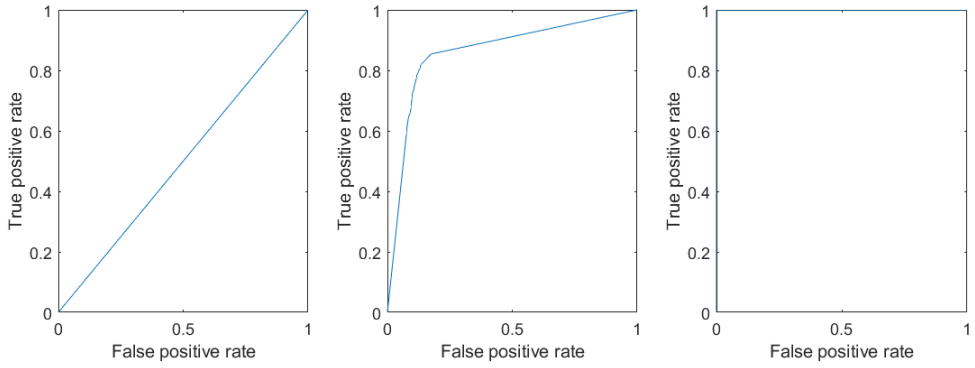


Figure 3.15: Example receiver operating characteristic (ROC) curves for a) A random classifier b) A good classifier c) A perfect classifier.

Of course, classifiers trained to identify drone model will have more than two classes. Confusion matrices are easily extended for multi-class performance evaluation. The metrics used to evaluate the performance of binary classifiers can be extended for multi-class performance using micro-averaging or macro-averaging [119]. Micro-averaging calculates the metric over the aggregate of classes, whilst macro-

averaging calculates the metric for each class individually, and then takes the average result. Macro-averaging is more suited for imbalanced classes, therefore is used throughout this thesis.

Chapter 4

Radar-Based Drone Classification Review

4.1 Introduction

This chapter provides an overview of the current state-of-the-art in UAV classification using radar, encompassing both drone vs bird discrimination and intra-drone classification. Rather than aiming to provide a comprehensive review, only the aspects that are relevant to the later chapters of this thesis are considered.

In recent years there have been a number of research papers published on the topics of detection, tracking, and classification of drones. This literature review considers the use of traditional machine learning classifiers as well as deep learning models for the purpose of drone classification. Machine learning describes the use of models to identify patterns within a dataset, using the learned features to make predictions upon new data. Deep learning is a subset of machine learning that is inspired by the human brain, using layers of artificial neurons to extract more

complex patterns and relationships from the data.

The purpose of this review is two-fold, firstly to understand and learn from drone classification techniques that are currently in use, and secondly to illustrate how this thesis contributes to the landscape of UAV classification.

4.2 Rule-based Approaches

As discussed in Chapter 2, there are marked differences between the micro-Doppler signatures of birds and drones that can be exploited to aid in classification. In [75], the authors collected data from a DJI Phantom Vision 2 drone and three species of birds at S-band. Wing-beat signatures and the presence of HERM lines provided a discriminating feature between the bird and drone spectrograms, respectively.

In [77], the authors discuss a number of features for drone vs bird classification based on radar spectrograms and cepstrograms, including RCS, velocity, periodicity, bandwidth, and symmetry. Drone spectrograms and cepstrograms were generated using both simulation and experimental data gathered with an X-band CW radar, demonstrating the presence of drone micro-Doppler. Finally, the authors gathered data from a helicopter and an octocopter, demonstrating how the number of rotors and periodicity affects the micro-Doppler signature.

Estimation of the number of rotors and blades possessed by a UAV will aid in the classification of drone type, whilst estimation of rotor length facilitates an approximation of drone size. In [3], an X-band CW radar was used to gather data from a helicopter and quadcopter, where the rotation rate, blade tip velocity, and rotor length were derived directly from the resulting spectrograms. It was demonstrated that the HERM line spacing was twice the rotor rate, due to contributions from two

blades. The authors show that spectrogram analysis and subsequent classification is impacted by the number of rotors, wind effects, and SNR. The authors of [120] demonstrate the effects of changing drone blade length and rotation frequency on the appearance of simulated spectrograms. Real data collected using an S-band CW radar is used to estimate blade length in a laboratory environment, achieving an error of 0.8% at a range of 1 metre from the rotating blade. Blade length is an invariant feature, therefore could form a robust basis of discrimination between drone models.

The authors of [79] developed an algorithm for automatic estimation of blade length from the UAV micro-Doppler signature. The maximum Doppler frequency f_{max} is estimated from the envelope of the micro-Doppler spectrum, the rotation frequency ω is estimated as the average time difference between consecutive peaks in the power spectrum divided by the number of blades, and the relationship $L = \frac{\lambda f_{max}}{4\pi\omega}$ is used to calculate blade length. The algorithm was verified using data collected at W-band for a helicopter, quadcopter, and octocopter, with an average error of 1.3%.

HERM line spacing and shaping is characteristic of the number of rotors, number of blades, rotational frequency, and blade length. Whilst previous work has used the blade-flash regime to extract discriminative features, the authors of [82] and [83] developed algorithms for estimating the blade rotation rate from HERM lines. A log harmonic summation algorithm is used to find the fundamental frequency of the spectrogram, whereby the log Doppler spectrum is cross-correlated with a harmonically spaced delta sequence. In [82], this algorithm was tested at S-band for a stationary micro-helicopter, and at W-band for a stationary hexacopter at a maximum range of 10 metres from the radar. In [83], the methodology is extended to consider multi-rotor drones, where the chopping and spool frequencies are extracted from UAV spectrograms, and the number of blades is estimated as the ratio of chopping frequency to spool frequency. The methodology is validated for a stationary

helicopter, quadcopter, and hexacopter using a C-band CW radar and a W-band FMCW radar, at a maximum range of 1 metre in a laboratory environment.

4.3 Machine Learning Approaches to Drone vs Non-Drone Classification

A common theme in the literature is the use of support vector machine (SVM) models for classifying drones and non-drones using CW radar. In [121], data is collected for 11 target classes (including fixed wing drones, a quadcopter, helicopters, rotating blades, and birds) using an X-band CW radar at a maximum range of 30 m. SVM and Naïve Bayes (NB) classifiers are trained on the first five eigenpairs extracted from the spectrogram's correlation matrix, achieving an accuracy of approximately 95%. In [122] an SVM is trained on features extracted from spectrograms and CVDs. Drone data were collected across various platforms (fixed wing, helicopter, four quadcopters, and an octocopter) using a Ka-band CW MIMO radar, whilst bird data was simulated. The algorithm was found to correctly classify 100% of drones and birds using just three micro-Doppler features relating to bandwidth and periodicity. This is because the velocity of a bird's wing is much lower than that of a drone rotor tip, and birds' wings beat at a lower frequency than the frequency of rotation of a drone's rotor. An SVM was also trained to classify small-sized and medium-sized drones, with accuracy >96%. In [123], an SVM was used to classify two quadcopters from non-drones using features measured using a K-band CW radar. The aim of this research was to explore the feasibility of drone detection using distributed radar sensors in urban environments, therefore classification ranges were low. Features were extracted from either the spectrogram, CVD, or cepstrogram using principal component analysis (PCA) and clustering. Best performance is achieved using the

CVD, and accumulating the confidence value over successive time segments.

SVM models have also been used for classification of FMCW radar signals. The authors of [124] demonstrate a method of classifying drones using an X-band LFMCW surveillance radar. Three measurement campaigns are considered, with maximum ranges of 624 m, 2.1 km, and 4.2 km, with eight rotary wing drones, three fixed wing drones, and opportune non-drone targets. 50 features are extracted from the range-Doppler plot and track segments, and an SVM is trained for classification, achieving an accuracy of 98%. The classifier is then extended to distinguish between rotary wing drones and fixed wing drones, with an accuracy of 93%. The authors assess performance as a function of number of antenna rotations, demonstrating that performance is improved by increasing the length of track segments. The authors of [125] gathered data from four different bird species and three drone models (DJI Phantom 3 Standard, DJI Inspire 1, and DJI S900) using a W-band FMCW staring radar. A linear discriminant analysis (LDA) model and an SVM were trained on the micro-Doppler spread and periodicity, classifying the drones and birds with an accuracy of 100%, demonstrating the importance of micro-Doppler sensitivity for drone-bird discrimination. The LDA model was also able to classify the drone models with an accuracy of 100%. However, the size of the data set was limited, with only four spectrograms of each class used for training.

In [126], a TERRM model is used to classify drone signals recorded with an X-band FMCW surveillance radar at a maximum range of 3 km. Empirical mode decomposition is used to decompose the signal into intrinsic mode functions (IMFs), from which 13 features are extracted. A classification pipeline is presented, where rule-based pruning automatically classifies targets as 'non-drone' if there are no micro-Doppler returns in the range-Doppler map, before extracting features from the signal, performing classification, and carrying out scan to scan filtering. Two class (drone/non-drone) and three class (fixed wing drone/rotary wing drone/non-drone)

problems are considered, achieving accuracies of 94.4% and 90.6%, respectively. Rule-based pruning and scan to scan filtering is shown to improve performance by 16.6%.

In [127], an S-band polarimetric radar was used to gather data from flapping and gliding birds and two models of drone, at a maximum range of 400 m. The authors aimed to reduce the dependence on micro-Doppler features for classification, turning instead to polarimetric features that are preserved when bandwidth, dwell time, and carrier frequency are reduced. Nine polarimetric features were extracted in the time-frequency domain and used to train a k-nearest neighbours (kNN) algorithm. The drone models were classified with a recall of >99%. Promising results were also shown when classifying soaring birds and a small drone with plastic rotor blades that exhibited negligible micro-Doppler.

In [128], a multi-layer perceptron (MLP) was trained on a set of 20 statistical features extracted from drone and bird tracks. Real drone data was collected using an S-band pulsed radar, whilst additional drone and bird tracks were simulated. The dominant classification features were found to be velocity based – 100% classification performance was achieved using the mean acceleration, variance of the jerk, and mean of the peak amplitudes. Retraining on the remaining features led to an accuracy of 99%.

A number of research papers have been published demonstrating the use of decision tree classifiers for filtering out non-drone targets at L-band. In [96], an L-band staring radar was used to detect and track a small hexacopter drone. Decision trees were then trained on trajectory-based features including height, micro-Doppler, acceleration, and jolt, to filter out non-drone confuser targets, correctly classifying 82% of drones. In [129], the authors use a two-stage decision tree to filter non-drone tracks. A hand-crafted tree filters the targets based on height, RCS, and veloc-

ity, and a second tree is trained on further kinematic features and micro-Doppler information. The model was trained on features extracted from a DJI Inspire 1 drone flown at a rural location, and tested on features extracted from a DJI Inspire 2 drone flying along an unseen trajectory at a different location, achieving a true positive rate of 85% with a false positive rate of 0.37%. Finally, the authors of [130] present a methodology for extracting the number of symmetrical frequency components about the target body. A decision tree is then trained on the number of symmetrical pairs, along with kinematic features (height, velocity, RCS, and the total number of micro-Doppler components). Training and test data are from DJI Inspire 1 and 2 drones, at a maximum range of 2.6 km from the radar, achieving a true positive rate of 95.7%. It was further demonstrated that this classifier is robust enough to generalise to classify fixed-wing drones, achieving a true positive rate of 92.8%. This suggests that the presence, number, and symmetry of HERM lines may be a robust feature for the discrimination of drones from birds.

4.4 Machine Learning Approaches to Drone vs Drone Classification

An emerging theme within the literature is the use of machine learning algorithms to classify drones based on the number of rotors. In [131] a dual-band CW radar (K-band and X-band) was used to obtain drone spectrograms which were then decomposed into feature vectors using PCA and used to train an SVM model for drone classification. Helicopters, quadcopters, and hexacopters were classified at a maximum range of 1.2 m with an accuracy of 94.7% when information from the two radar sensors were combined. In [132] helicopter, quadcopter, and hexacopter data were obtained using a K-band CW radar, with multiple drones flying at the same

time. A k-means classifier with seven centroids was trained on cadence frequency spectrums, achieving an average accuracy of 94.2%.

The authors of [133] present a methodology for classifying drones based on the number of rotors and blades. An SVM algorithm was trained on features extracted from drone HERM lines, including the location and power of micro-Doppler peaks. Drone spectrograms were simulated at S-band, with either 4 or 6 rotors and 2 or 3 blades on each rotor. The model showed promise for the classification of both hovering and moving drones, with true positive rates of 99% and 93% respectively. In [134], an SVM algorithm was trained on concatenated spectrograms and cepstrograms, following PCA. The drone data were simulated at X-band, with varying blade lengths, RCS, and placement of point scatterers, based on six common quadcopter models. Performance was highly dependent on model, with the probability of correct classification ranging from 41% to 81%.

The authors of [135] present a methodology for estimating drone blade length and rotation frequency, in addition to the number of rotors and blades. Helicopter and quadcopter time series were simulated with either two or three blades per rotor, with a randomly selected blade length and frequency of rotation. Five preprocessing pathways were generated, using the real and imaginary components in the time and frequency domains, and the absolute signal in the frequency domain. MLP models were trained to classify the number of drone propellers and blades, and regression trees were subsequently used to estimate the blade length and frequency of rotation. When the SNR exceeded 5 dB, the number of propellers and blades were estimated with almost 100% accuracy. Frequency and length estimates were found to be more accurate for helicopters than for quadcopters. At high SNR (>15 dB), the maximum error was 4% in frequency and 6% in blade length.

In addition to model classification, machine learning techniques have also been

used to determine whether or not a hovering drone is carrying a payload. In [136], data is gathered using an S-band multistatic radar, where it is demonstrated that heavy payloads increase the Doppler bandwidth and result in more uniform HERM lines. NB and LDA classifiers are trained on micro-Doppler centroid features, to estimate the payload carried by a hovering drone at a maximum range of 80 m. The authors demonstrate that performance can be improved by combining the output of multiple classifiers each trained on data from a single node, achieving accuracies up to 100%. In [137], this work is extended to consider five classes of drone payload. Two classification methodologies are presented: the first uses singular value decomposition (SVD) on the temporal signal, and the second method uses centroid and bandwidth features from the spectrogram. Features extracted using SVD are more suited for the classification of moving drones (accuracy >98%) than centroid features, whereas the opposite is true for hovering drones. Using voting to combine information from each node improves performance by 10% compared to the monostatic case. Finally, in [138] the spectral kurtosis is computed from the spectrograms of six loaded drone classes, and used as the input to a kNN algorithm. Data is gathered from a DJI Phantom Vision 2+ drone, with a payload weight ranging from 0 g to 600 g at a maximum range of 90 m. Flying drones are considerably more difficult to classify, with a probability of correct classification of 70.8% compared with 92.6% for hovering drones. Whilst payloads are not considered in this thesis, it is evident that micro-Doppler signatures contain information pertaining to drone mass, which could be exploited for drone model classification.

4.5 Deep Learning Approaches to Drone vs Non-Drone Classification

Recent years have seen the development of deep learning techniques for discriminating drones from non-drones. This subsection focuses on classifiers that use layers of

artificial neurons to extract patterns from data.

The authors of [139] used an X-band LFMCW surveillance radar to gather data from drones flying at both short (up to 600 m) and intermediate (up to 2 km) ranges. A two-branch CNN was used to classify drones from non-drones, whereby the range profile matrix was processed independently to RCS, SNR, and radial velocity features. The two branches were then combined and fed through a softmax layer to output a 'drone' or 'non-drone' label, achieving an accuracy of 95%. In [140], the authors also used a CNN to classify drones from non-drones based on their range-Doppler spectra, achieving an accuracy of 90%. Data were collected from 6 different drone models (up to a maximum range of 200 m) and a variety of non-drones (birds, humans, and corner reflectors) using a W-band staring FMCW radar. The authors demonstrate that performance falls when testing the classifier on drone models that were not included in the training phase, but that including a simulated drone class during the training phase improves accuracy by 10%. This highlights the importance of representation within the training data.

In [141], a deep belief network (DBN) is used to distinguish between two micro-drones, a remote-control bird, and a reference signal. Data are recorded using an S-band CW radar in a laboratory environment, with a maximum range of 3 m. The spectral correlation function (SCF) of the micro-Doppler signature formed the input to a DBN. The SCF is the Fourier transform of the autocorrelation function, representing the correlation between signals that have been shifted in frequency by the micro-Doppler effect, in the frequency domain. Gaussian noise is added to investigate classifier robustness, where an accuracy of 90% could be achieved provided SNR exceeded -2 dB.

In [142], the authors used various deep learning techniques to classify drones and distinguish them from birds. Complex fully temporal convolutional networks,

multi-layer perceptrons (MLP), and decision trees were used to classify various representations of simulated drone signatures: the temporal signal, the covariance matrix, and kinematic features. The authors developed a classification pipeline consisting of a convolutional network (FCN) designed to perform an FFT, fully convolutional layers, and MLP blocks adapted for covariance matrices. A decision tree boosting method is used to distinguish drones from birds, and to classify drone models, using kinematic features extracted from simulated bird and drone trajectories.

The authors of [143] propose two methodologies for the discrimination of a DJI Phantom 4 Pro drone from non-drones (birds and ships). Data is collected at ranges up to 4 km using an S-band horizontal polarised pulsed radar. The first method consists of a shallow neural network trained on eight track features (including area, signal amplitude, and velocity) extracted over 10 scans. The second method comprises a CNN-LSTM combination trained on patches of the plan position indicator (PPI), whereby the CNN extracts discriminative features and the long short-term memory network (LSTM) captures the relationship between consecutive scans. The image-based method results in an accuracy of 90%, with a false alarm rate of 7%, compared to the track-based approach with an accuracy and false alarm rate of 81% and 12% respectively. This demonstrates the suitability of image-based deep learning techniques for drone classification, compared to approaches using hand-crafted features.

In [113], the authors use CNNs for the classification of drone spectrograms. Data from three drone models and four bird species are collected at ranges up to 100 m, using K-band and W-band FMCW radars. Transfer learning is used alongside the GoogLeNet [10] architecture, whereby the weights in the first 10 layers of the network were frozen and the training data was used to train the final layers only. Data from a different W-band FMCW radar was used to validate the performance of the classifier, achieving an accuracy of 99% for the two class (drone vs bird)

problem, and 98.5% for the four class (drone, bird, clutter, noise) problem. Note that in this work, the drone micro-Doppler signature gives the appearance of blade flashes rather than HERM lines.

4.6 Deep Learning Approaches to Drone vs Drone Classification

The authors of [144] compare three deep learning methods for classifying drones based on simulated micro-Doppler blade-flash signatures. The number of blades, blade rotation rate, RCS, and blade length are varied to simulate the returns from three drone models: a S1000+ octocopter, a DJI Phantom 2 quadcopter, and a Vario helicopter. An FCN is trained, allowing each time step to be classified individually, whilst the second method uses an LSTM trained on sequences of spectra. Finally, performance is compared to an MLP trained on spectrograms. The FCN slightly outperforms the LSTM, whilst the performance of the MLP classifier is significantly lower. The authors demonstrate that accuracy falls with decreasing SNR, and that an SNR in the order of 20 dB is required to classify with 90% accuracy.

In [145], a CNN is trained to classify three models of drone from spectrograms collected using a Ku-band FMCW radar at a maximum range of 412 m. The authors show that reducing the phase noise due to leakage from the transmitter to the receiver improves performance by 10%, demonstrating the importance of high signal to background ratio for reliable classification. The integration time is shorter than that of the blade rotation period, meaning that the UAV micro-Doppler signatures appear as blade-flashes.

In [146], the authors use a CNN to classify a DJI Inspire 1 quadcopter and a F820 hexacopter, measured using a Ku-band FMCW radar. The drones are either

stationary (oriented horizontally or vertically) in a laboratory environment at a maximum range of 5 m, or hovering outdoors with a maximum range of 100 m. The classifier performs poorly when the plane of blade rotation is perpendicular to the radar, due to a lack of visible blade-flash features in the spectrogram. Merging the spectrogram with the CVD leads to an increase in performance from 89.3% to 94.7%. In [147] the authors extend this work by encoding polarimetric information within the RGB channels of the merged spectrogram-CVD inputs, boosting the accuracy from 89.9% to 99.8%.

The authors of [148] trained a CNN to perform drone model classification using radar spectrograms measured with an X-band CW radar. Data are collected from five drone models: two fixed wing models, a helicopter, a quadcopter, and an octocopter. Integration time is varied from 2.7 ms to 10.7 ms, placing classification within the blade flash regime, where it is demonstrated that the shorter the integration time, the better the performance. The CNN is shown to outperform an LSTM network, achieving a accuracies of 97.7% and 89%, respectively. The authors also introduce deep learning techniques for pre-processing (denoising), and for generating additional training data for supplementing sparse data sets.

In [29] a CNN is trained to classify five simulated drone models. Drone spectrograms are simulated using a long window STFT, performing classification based in the HERM line regime. The authors investigate the relationship between performance and SNR, achieving a mean average precision of 0.46 at an SNR of 5 dB. Data are simulated at a range of carrier frequencies, from S-band to W-band, however, results are not presented at Ku-band.

4.7 Summary

The literature relevant to this work is summarized in Table 4.1, with details about the radar used for data collection and the classification method applied. Methods are compared in terms of the classifier used, the features used for training, the maximum range at which drone data is gathered, the problem set (e.g. drone/drone classification or drone/non-drone discrimination) and the number of classes. Several conclusions can be drawn from the reviewed literature, as discussed below.

- **Information present in the micro-Doppler signature can be used to differentiate drones from birds using simple rule-based approaches.**
  Under high SNR conditions drone spectrograms are visually distinguishable from those of birds, due to the presence of either high frequency blade flashes or HERM lines, depending on the integration period.

- **Micro-Doppler features can be used to characterise drones.**
  Blade flashes indicate the number of rotors and blades and can be used to estimate the frequency of rotation and blade length. When longer integration times are used, HERM lines are indicative of the frequency of rotation and number of blades. Estimating these parameters and comparing them against a library of known drone models could assist with classification.

- **Traditional machine learning approaches can be used to automatically distinguish between drones and non-drones.**
  Features extracted from the micro-Doppler signature (e.g. bandwidth and periodicity) are among the most commonly used features for drone classification. Track-based kinematic features such as acceleration and jolt have also shown promise for drone-bird classification.

- **Machine learning algorithms can utilise micro-Doppler signatures for classification of drone type, and for payload classification.**

  The number of rotors, number of blades, blade length, and weight are robust discriminative features that affect the appearance of drone spectrograms.

- **Deep learning techniques show promise for classifying drones and non-drones directly from the data.**

  CNNs have been used to classify radar images, including r-D maps, PPI frames, and CVDs.

- **Transfer learning improves performance whilst minimising the training time.**

  Well established CNNs trained on optical images have shown promising performance following retraining on small amounts of radar data, validating the approach taken in this thesis.

- **Drone classification performance falls with decreasing SNR.**

- **Decision fusion can be used to improve classification performance.**

Table 4.1: An overview of the existing research into drone classification using radar.

Reference	Frequency	Radar Details	Classifier	Features	Maximum Range	Labels	Number of Classes
[128]	S-band	Pulsed	MLP	Simulated trajectory and RCS features	Not specified	Drone/Non-drone	2
[121]	X-band	CW	SVM, NB	Micro-Doppler eigenvectors and eigenvalues	30 m	Drone/Non-drone and Drone/Drone	11
[136]	S-band	Multistatic pulsed (NetRAD)	NB, DA	Micro-Doppler features	78 m	Loaded/Unloaded drone	3
[96]	L-band	3-D staring radar	DT	Trajectory-based features	1 km	Drone/Non-drone	2
[146]	Ku-band	FMCW	CNN	Merged spectrogram and CVD	5 m indoors, 100 m outdoors	Drone/Drone	2
[127]	S-band	Pulsed (birdRAD)	kNN	Polarimetric features	400 m	Drone/Non-drone and Drone/Drone	4
[137]	S-band	Multistatic pulsed (NetRAD)	NB, DA, RF	SVD and centroid of micro-Doppler signature	103 m	Loaded/Unloaded drone	5
[122]	Ka-band	CW MIMO	SVM	Features from spectrogram and CVD	Not specified	Drone/Non-drone and Drone/Drone	8
[141], [149]	S-band	CW	DBN	SCF of micro-Doppler signature	3 m	Drone/Non-drone and Drone/Drone	4
[135]	Not specified	Not specified	MLP, DT	Temporal signal and DFT	Not specified	Drone/Drone	4
[131]	K-band and X-band	Dual band CW	SVM	Spectrogram	1.2 m	Drone/Drone	3
[144]	Not specified	Pulsed	MLP, LSTM, FCN	Simulated spectrogram	Not specified	Drone/Drone	3
[132]	K-band	CW	K-means clustering	Cadence frequency spectrum	Not specified	Drone/Drone	3
[133]	S-band	Pulsed	SVM	Micro-Doppler features	20 m	Drone/Drone	2
[150]	Not specified	Not specified	CNN	Simulated spectrogram	Not specified	Drone/Drone	Not specified
[123]	K-band	CW	SVM	CVD	Not specified	Drone/Non-drone	2
[148]	X-band	CW	CNN, LSTM	Spectrogram - simulation and experiment	Not specified	Drone/Drone	5
[124]	X-band	LFMCW surveillance	SVM	Signature and kinematics features, from r-D plot and track segment	4.2 km	Drone/Non-drone and Drone/Drone	2
[126]	X-band	FMCW surveillance	Total Error Rate Reduced Multivariate model	EMD on signal	3 km	Drone/Non-drone and Drone/Drone	2,3
[125]	W-band	FMCW staring	LDA, SVM	Micro-Doppler features	Not specified	Drone/Non-drone and Drone/Drone	2, 3
[139]	X-band	FMCW surveillance	CNN	Range profile matrix	2 km	Drone/Non-drone	2
[130]	L-band	3-D staring radar	DT	Micro-Doppler and kinematic features	2.6 km	Drone/Non-drone	2
[151], [142], [142]	Not specified	Not specified	FTCN, MLP, DT	Spectrogram, covariance matrix, kinematic features	Not specified	Drone/Non-drone and Drone/Drone	4, 11
[129]	L-band	3-D staring radar	DT	Kinematic features and micro-Doppler features	Not specified	Drone/Non-drone	2
[147]	Ku-band	FMCW	CNN	Merged spectrogram and CVD	6 m	Drone/Drone	3
[134]	X-band	Pulsed	SVM	Simulated spectrograms and cepstrograms	365 m	Drone/Drone	6
[138]	S-band	Multistatic pulsed (NetRAD)	kNN	Spectral kurtosis	90 m	Loaded/Unloaded drone	6
[145]	Ku-band	FMCW	CNN	Spectrogram	412 m	Drone/Drone	5
[113]	K-band, W-band	FMCW	CNN	Spectrogram	100 m	Drone/Non-drone	2
[152]	K-band, W-band	FMCW, CW	LDA, SVM, CNN	SVD on spectrogram	Not specified	Loaded/Unloaded drone	2, 5
[153]	Ku-band	Not specified	KNN, SVM, RF	Chebychev moments from CVD	Not specified	Drone/Non-drone and Drone/Drone	10
[29]	S-band to W-band	Pulsed	CNN	Simulated spectrogram	5 km	Drone/Drone	5
[143]	S-band	Horizontal polarised pulsed	CNN-LSTM	Track features, PPI	4 km	Drone/Non-drone	3
[154]	E-band	FMCW	Not specified	r-D map	150 m	Drone/Non-drone	2
[140]	W-band	FMCW scanning	CNN	Spectrogram	200 m	Drone/Non-drone	2
[155]	X-band	Not specified	CNN	Spectrogram	50 m	Drone/Noise	2
[156]	S-band	FMCW polarimetric	LDA, NB, DT, SVM	Spectral and polarimetric features	575 m	Loaded/Unloaded drone	2

Whilst conducting a review of the literature, a number of areas have been identified for further investigation. These are summarised below.

- **CNNs for classification of drone and non-drone spectrograms.**

  At the time the work presented in this thesis commenced, there was little evidence of CNN use for drone-bird classification. It was hypothesised that CNNs could lead to improved performance due to local connectivity and extraction of non-linear features for learning, and it is believed that the work presented in this thesis is among the first to use CNNs in this way [44].

- **Consideration of staring radar for UAV classification.**

  The majority of existing research takes place using high frequency CW or FMCW radar, where range is limited due to the co-location of the transmitter and receiver. This thesis will use data collected from a pulsed staring radar at L-band, at ranges of up to 5 km.

- **Classification within the HERM line regime.**

  Whilst recent years have seen CNNs used for drone model classification using radar spectrograms, the bulk of that work has been in the blade-flash regime. This thesis uses data collected using staring radar, where long integration times are used to increase SNR, resulting in drone micro-Doppler exhibiting as HERM lines.

- **Use of long-dwell spectrograms.**

  Increasing the number of spectra could increase the likelihood of capturing not only micro-Doppler features but also kinematic features from the amplitude and shaping of the return from the main body, potentially enhancing performance.

- **Classification of drone signatures under challenging conditions.**

  When simulated data has been used within the literature to train and test

models it is unclear how closely the simulations align with observed signatures. If not simulated, much of the training data is otherwise idealised, making some assumptions that may not be realistic under operational conditions [157][158].

- **Investigation of classification performance as a function of SNR and range.**
  This theme is explored throughout the thesis in order to better understand the effects of range and drone model upon classification performance.

- **Improving robustness to low SNR test data.**
  It has been demonstrated in the literature that classification performance is dependent on SNR. This thesis will consider methods of improving robustness to noise [7].

- **Consideration of multiple classification strategies for UAV classification.**
  Existing research has largely considered single drone classification problems in isolation: either group classification (fixed wing or rotary wing drone), classification of the number of rotors, or model-level classification. A natural extension is to consider a holistic approach to classification, from group classification down to model identification. It is believed that the work presented in this thesis is among the first to use CNNs to systematically classify real drone spectrograms in the HERM line regime [41].

- **Fusion of the CNN output across consecutive spectrograms.**
  Possible approaches include majority-rules voting and integration of classifier confidence to reduce variance and improve performance.

This thesis presents results validated on real outdoor data collected using a low frequency pulsed staring radar. Data is collected across a variety of drone models at long ranges and multiple locations (both rural and urban). Data collected using a Ku-band FMCW staring radar prototype is used to train classifiers for drone model classification, where drone micro-Doppler signatures appear as HERM lines. The

approach to classification is systematic, and classification performance is improved using optimisation of training hyperparameters, decision fusion, and considering numerous network architectures.

Chapter 5

Establishing CNNs For Drone Classification

5.1 Introduction

The work presented in this chapter explores the use of CNNs for classifying drones and birds. Whilst statistical methods and machine learning techniques had been used with some success to discriminate between drone and non-drone targets, at the beginning of the author's research the use of CNNs for drone and bird classification was still in its infancy.

The first step is to establish the suitability of CNNs for drone-bird classification using spectrograms recorded at L-band. Features extracted from the tracker are used to investigate the feature space of the drone vs bird classification problem, and the extracted features are used for training traditional machine learning models. These models are used to gain an understanding of the features that could be used to distinguish drones from birds, and to provide a benchmark for classification success.

A CNN is trained, and performance is evaluated and compared with the machine learning benchmarks [42].

The main sources of error in classification problems are training data noise (class noise and attribute noise), and unsuitable models [159][160]. Sources of noise in the training data include labelling errors, measurement errors, variations between samples, and incomplete coverage of the domain. Natural variations between training samples are to be expected, due to the wide scope of the problem: in the context of drone vs bird classification, targets belonging to the same class will have noticeably different radar returns due to differences in shape, size, and flight style. For example, the flight dynamics of a passerine-type bird can be expected to be significantly different to that of a soaring bird. Similarly, drones are capable of highly dynamic flight, with the ability to carry out a wide range of manoeuvres. Therefore, a reliable classifier needs to be robust to unseen cases. To measure this, real and dynamic data will be split into training and test subsets and the performance of the classifier will be examined on the (unseen during training) test data.

Sources of model error include the choice of network architecture and hyperparameters, choice of threshold, and over or under fitting of the model to the training data. Transfer learning is used to compare the performance of six high performing networks from the field of computer vision. The results indicate that shallower networks are less prone to overfitting, resulting in a more robust classifier. ROC curves are used to illustrate performance and select the most appropriate threshold, maximising the true positive rate whilst minimising the false positive rate.

In the majority of previously published work, data has been simulated, or, in the cases where real data has been used, the drone range was in the order of 1 km [158][157]. The natural extension of the literature is to consider the relationship between SNR and classification performance, and to investigate performance under

increasingly demanding conditions. Since SNR is affected by RCS and target range, understanding performance as a function of SNR allows limits to be placed on the size of drone that can be classified reliably, and the distance at which it can be classified.

The limitations of a CNN and decision tree are explored in this chapter. CNN performance is systematically evaluated as a function of SNR through the addition of Gaussian noise to the test spectrograms, providing a novel contribution to the field. Augmentation methods are then used to enhance performance, resulting in a classifier that is more robust to noisy test data.

The layout of the chapter is as follows: Section 5.2 gives an overview of the methodology followed, including the target labelling process, the choice of spectrogram parameters, and an overview of the training and test data sets. In Section 5.3, drone and bird feature space is explored and a decision tree classifier is trained and tested. In Section 5.4, a CNN is trained and tested, and compared with the machine learning benchmark. In Section 5.5, the SNR is artificially reduced through the injection of Gaussian noise and the subsequent effect on performance is investigated. Methods of improving robustness to noise are investigated, where Bayesian Optimisation is used to maximise performance under less than ideal conditions. Finally, the chapter conclusions are given in Section 5.6.

5.2 Data Collection

Data was collected using the Aveillant L-band staring radar described in Chapter 2, where an I1-D drone flew on various trajectories corresponding to different predetermined scenarios, at a 0.7-2.5 km range from the radar. Fig. 5.1 shows an

example trajectory from the experimental campaign, with the drone GPS track in white. The recorded data consists of both drone and non-drone tracks, where the non-drones are primarily birds but also include non-biological targets such as cars, aircraft, and tractors. As well as genuine tracks, there are also false tracks caused by false detections, and erroneous tracks due to multi-path effects.



Figure 5.1: The GPS track (white) of an I1-D drone flight in Google Earth [12], overlaid onto the radar field of view (purple). The radar position is denoted as GK.

5.2.1 Ground Truth

Supervised learning algorithms such as CNNs require large amounts of accurately labelled data for training [159], to allow for boundaries to be drawn between classes. To ensure accurate labelling, truth information was used to identify tracks belonging to targets of interest, and to assign them with a class label (drone, bird, or car). The sensor measurements are linked to the geo-referenced truth information by mapping both in Google Earth Pro [12]. In a joint trial with Aveillant, drone tracks were identified by association with the tracks recorded by the onboard GPS [56]. Photographs and associated metadata were used to identify tracks belonging to opportune birds [161], whilst car tracks were identified using the recorded height and proximity to roads.

An iPad tablet and the Theodolite application [13] were used to capture photographs of opportune targets, along with a label to record the target's identity. Metadata including the GPS coordinates, altitude, and the azimuth and elevation angles of the tablet at the time of taking the photograph were saved alongside the image (Fig. 5.2). The metadata was then extracted forming a KMZ file that can be viewed using Google Earth Pro [12], showing vectors along which the opportune target was located. Radar tracks belonging to opportune birds can be identified by correlating the tracks with the truth vectors in time and bearing. If multiple photographs of the same target are taken in succession, then a further check is that the direction of travel of the observed target is in agreement with the associated track. Fig. 5.3b shows an example bird track along with the associated truth vectors, showing strong correlation between the recorded track and field observations. The ground truth process for opportune targets is explained in more detail in [161].

The time series data of the confirmed drone, bird, and car tracks were then extracted, forming a labeled bank of data for training and testing machine learning algorithms.

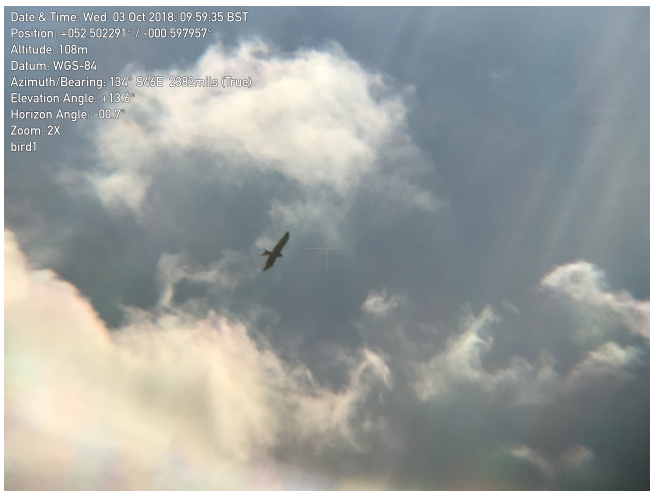
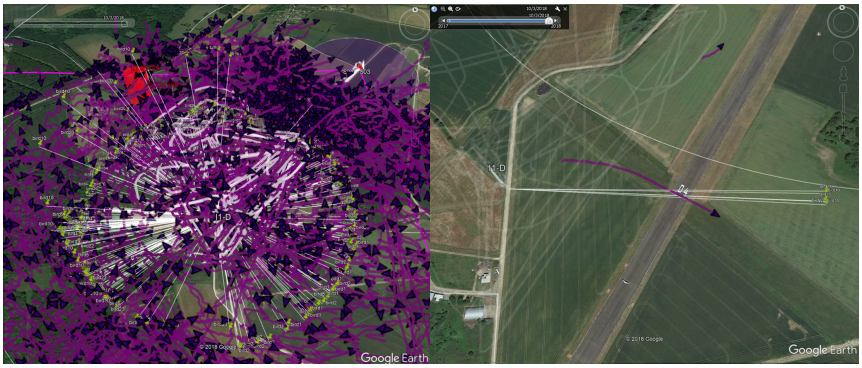


Figure 5.2: An opportune bird target photograph overlaid with metadata recorded using Theodolite [13].



(a) (b)

Figure 5.3: Radar tracks in Google Earth [12] a) All tracks recorded during a single trial (purple) b) A single track (purple) and vectors along which field observations were recorded (white).

5.2.2 Data Set A

Data were provided by Aveillant across two locations: Deenethorpe (Northamptonshire, UK), and Antwerp (Belgium) (Fig. 5.4). Deenethorpe is a rural environment, whilst the site at Antwerp is a combination of industrial and semi-urban. This data set will be referred to as "Data set A" to distinguish it from a second Aveillant data set, "Data set B", used in Chapter 7.



(a) (b)

Figure 5.4: a) Deenethorpe (UK) site map, b) Antwerp (Belgium) site map. Displayed in Google Earth [12].

Fig. 5.5 shows the two models of drone considered in this chapter - both drones are similar in size ($\sim$ 0.5 m diameter). In total, there is approximately 4.3 hours of drone data and 1.8 hours of bird data, where the amount of data for each target is summarised in Table 5.1.



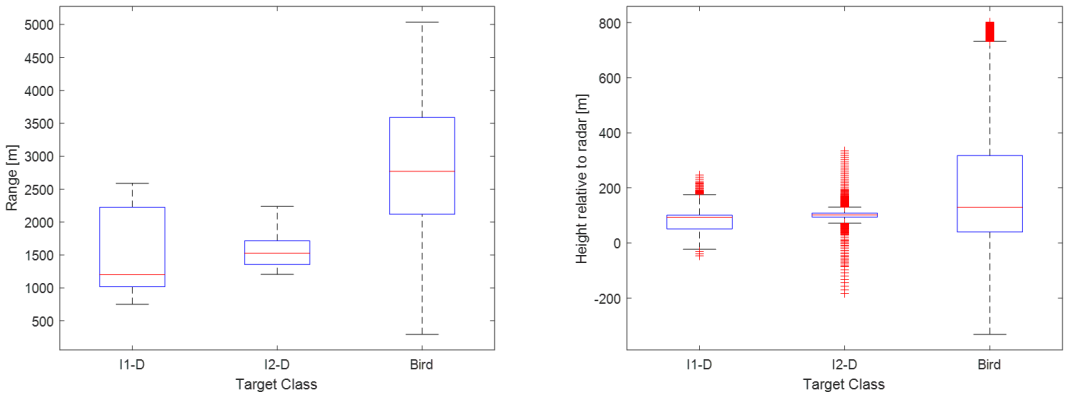
(a) (b)

Figure 5.5: Images showing a) I1-D, b) I2-D.

Table 5.1: Summary of data set A.

Location	Deenethorpe		Antwerp	
Target	I1-D	Bird	I2-D	Bird
Number of frames	27893	20935	27436	1845
Time (minutes)	129.7	97.3	127.6	8.6

Fig. 5.6a shows the spread in range of the targets in data set A. The median is represented as a red line, the boxes represent the interquartile range, and outliers (beyond 1.5 times the interquartile range from the box) are shown as a red "+". All of the drone flights are below 3 km in range, with the majority of data ranging from 500 m to 2 km. Classification performance is expected to fall with increasing range, as key classification features become masked by the noise floor [43]. Although SNR can be artificially reduced [7], this is not equivalent to operating at longer ranges, therefore real long-range data is needed in order to test the limitations of the classifier by validating classification performance at longer ranges. Fig. 5.6b shows the spread in target height above the ground, indicating that the majority of data is at a single height – around 100 m for the two drones.



(a) (b)

Figure 5.6: Box plots showing the coverage of data set A in a) Range, b) Height relative to the radar.

Data set A Subset

Data set A will be used in full in Chapter 7, but in order to establish baseline performance, data from a single site (Deenethorpe) was used for training and testing the classifiers in this chapter. The classes were balanced by randomly discarding some of the drone spectrograms to provide a reduced data set consisting of 966 spectrograms, equally balanced between birds (of various species) and an I1-D drone. 60% of the data was used for training and 30% was used for testing, whilst the remaining 10% was used for validation during training.

Non-Biological Confuser Targets

Birds are not the only type of confuser target from a drone surveillance perspective - humans, vehicles, animals, and even air conditioning units can all be mistaken for drones. In order to investigate how well the drone-vs-bird classifier generalises to non-drone or bird targets, 39 aircraft and 185 car spectrograms were recorded. The

classifiers will be tested on this data set and the false positive rate will be calculated to determine the approach for tackling non-biological confuser targets.

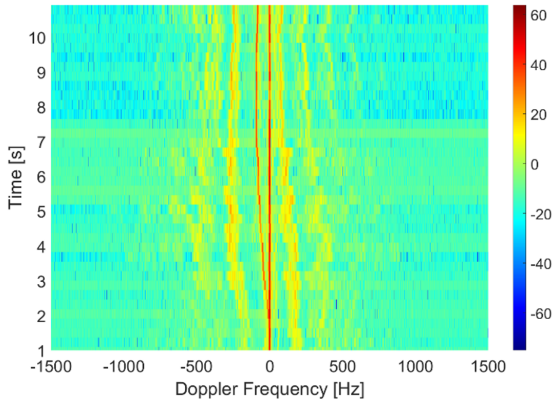
5.2.3 Bird and Drone Spectrograms

To generate the spectrograms used in this chapter, a Hamming window is applied to the time series data before applying an FFT of length 2048 pulses to transform into the frequency domain. This produces Doppler spectra, 40 of which are combined to form a single spectrogram with a time resolution of 0.279 seconds.

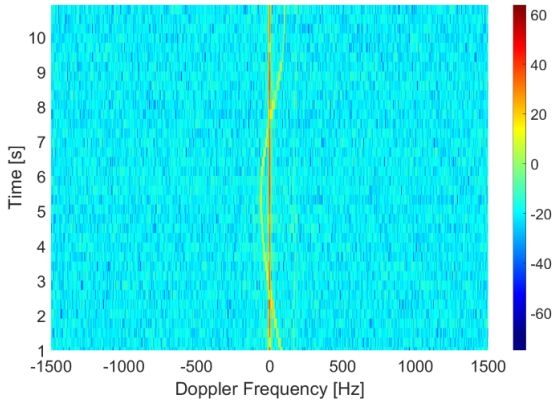
Fig. 5.7a shows an example I1-D drone spectrogram. The colour map represents the signal power in decibels, relative to unit power. The long dwell time captures the time evolution of the Doppler returns from the target body and its constituents. The drone body return is discernible as the series of high amplitude echoes close to zero Doppler frequency, and HERM lines due to the rapidly rotating rotors are distinctly visible as modulations either side of the body return. The SNR of the body was approximately 45 dB, and the first HERM lines sit approximately 15 dB below the body return.

Fig. 5.7b shows an example spectrogram belonging to a bird of unknown species, where the bird body is represented by the strongest observable Doppler return, with an SNR of approximately 35 dB. The bird follows a circular trajectory, resulting in an oscillating Doppler frequency from positive to negative. Comparison with Fig. 5.7a shows that the bird trajectory is noticeably less uniform than that of the drone, indicating that this could be a potential feature for classification. The bird spectrogram shows an obvious lack of HERM lines, and no wing-beat return is observable with a window length of 0.279 seconds. Fig. 5.7a-b are typical of the spectrograms generally observed in the data set, and the presence (or absence) of

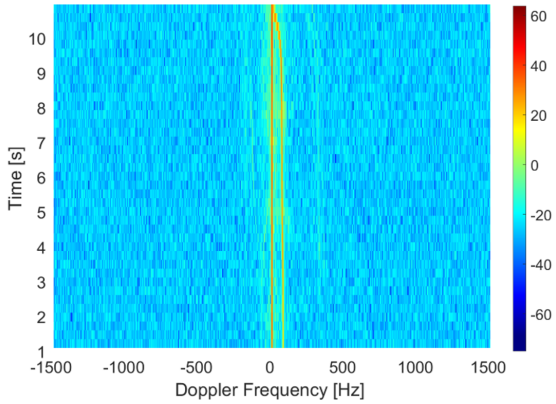
HERM lines is an obvious basis for classification, indicating that high performing target discrimination ought to be possible.



(a)



(b)



(c)

Figure 5.7: Example spectrogram showing the Doppler profile of a) an I1-D rotary wing drone b) a Bird c) an H-D rotary wing drone.

As target range increases or RCS reduces, the HERM line power will be reduced and will eventually become undetectable above the noise floor, as shown in Fig. 5.7c [7][64]. The separation between the two target classes is reduced, making the classification problem more challenging. Therefore, classification accuracy can be expected to degrade with SNR.

5.3 Decision Tree Benchmark Performance

The literature demonstrates that decision trees can be used to reliably filter out non-drone targets using kinematic and micro-Doppler features [161][96][130][129]. Therefore, a decision tree classifier is trained on data set A, to establish a benchmark performance for classification success. The machine learning benchmark forms a basis of comparison for a CNN, demonstrating the feasibility of using CNNs for drone and bird classification.

5.3.1 Classification Features

Five features (absolute velocity and acceleration, jolt, detection amplitude, and the number of Doppler components) were extracted from each frame of drone and bird data. These features were selected as they were demonstrated in the literature to be key features for the classification of drones using decision trees [96][97][161][130], whilst also being features that are present within the spectrograms therefore are theoretically also accessible to the CNN. The features are predominantly kinematic, with a single representation of spectral information. Each frame is labelled according to the target type using the method outlined in Section 5.2.1, and the set of features and their corresponding labels were used to train various machine learning algorithms.

Before training any models, feature space is explored in order to investigate the separability of drones and confuser targets, to identify potential limitations and to determine the feasibility of distinguishing drones from non-drones using a small number of parameters.

Fig. 5.8 shows a plot of echo amplitude, velocity, and the number of Doppler components. There is significant overlap in the bird and drone echo amplitude, as expected due to the similarity in RCS [58]. The drone spectra typically contain many more Doppler peaks on average (5.06) than those of the birds (0.01) due to the HERM lines caused by the rotating blades, providing clear separation between the drone and bird classes. It is therefore expected that the number of Doppler components will be a key feature for classification. This hypothesis will be tested by evaluating decision tree feature importance.

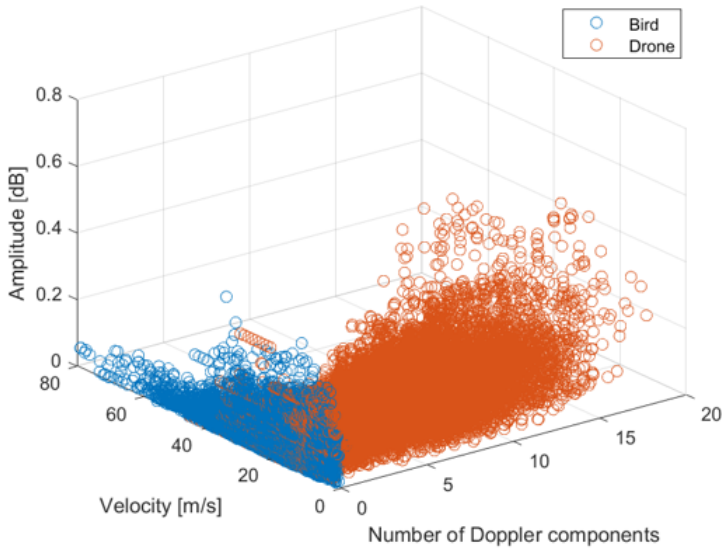


Figure 5.8: A cluster plot showing the number of Doppler components, velocity, and echo amplitudes of the bird and drone data frames.

Fig. 5.9 demonstrates how the kinematic features may be used to separate the drone and bird data frames. There is overlap in the velocity, acceleration, and jolt of the drones and birds, however, the birds typically have a higher velocity

and acceleration. Note that the bird velocity is susceptible to over-estimation due to errors in the positional measurements and incorrect association of detections to tracks. It was expected that the jolt (rate of change of acceleration) would be a differentiating feature as drones are relatively smooth flying, with changes in position and voltage controlled by the rotors, whilst birds often fly on more irregular trajectories, particularly when foraging. However, this is not observed in practice, perhaps due to wind conditions, or due to the species of birds for which data was collected, as short-lived tracks close to the ground were less likely to be detected both by the radar and by the field spotters.

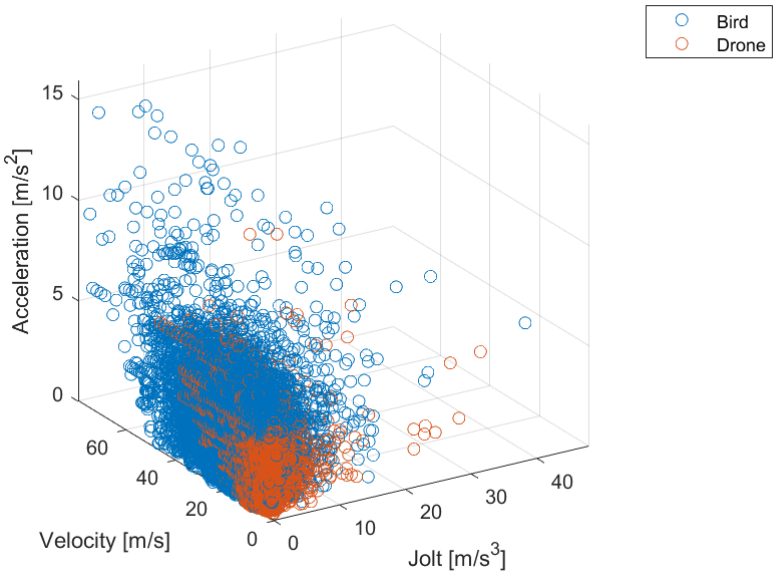


Figure 5.9: A cluster plot showing the velocity, acceleration, and jolt of the bird and drone data frames.

Finally, two other types of confuser targets were considered - cars, and aircraft. As shown in Fig. 5.10, these types of target are easily separated using simple filters on height, velocity, and amplitude. In [129], it was demonstrated that high classification performance could be achieved by pre-filtering the targets using simple thresholds on these values. Therefore, cars and aircraft will not be treated as target classes, and only two classes (bird and drone) are used for training the models, with the

assumption that other confuser targets can be pre-filtered. The models are tested on car and aircraft data to assess the false positive rate caused by non-biological confuser targets, to assess robustness in the case where the pre-filtering may fail.

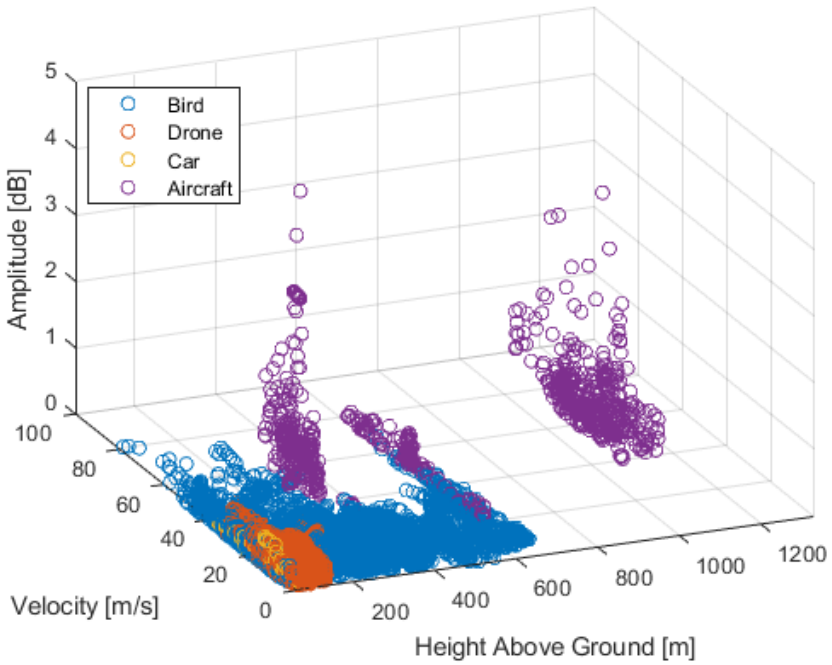


Figure 5.10: A cluster plot showing the velocity, height, and amplitude of the bird, drone, car and aircraft data frames

5.3.2 Decision Tree Performance

A decision tree was trained on the features outlined in Section 5.3.1, where 60% of the frames were used for training, and 40% for testing. A standard classification and regression tree (CART) was trained using GDI as the split criterion [162]. To prevent overfitting, a limit of 25 was set on the maximum number of splits, where this number was chosen empirically.

Fig. 5.12 shows that the decision tree performs with a high accuracy, where 97.6% of the drone frames are correctly classified. Fig. 5.11 shows predictor importance

(Section 3.3.5), where the dominant classification feature was the number of Doppler components, as predicted.

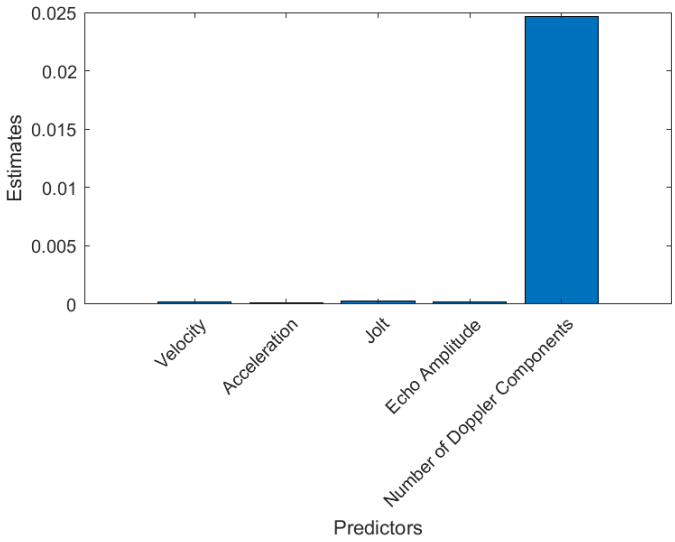


Figure 5.11: Bar chart showing the ranking of decision tree predictor importance.

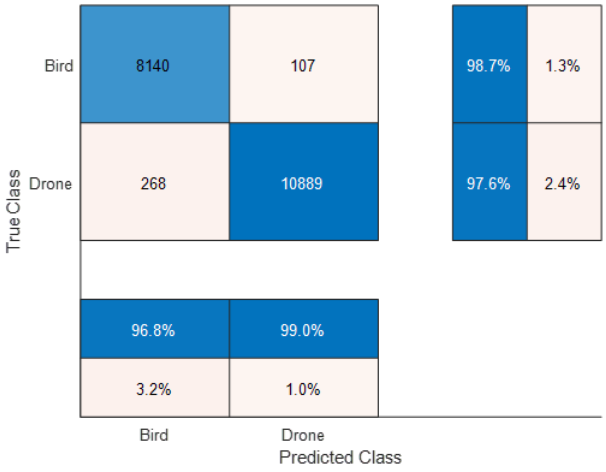


Figure 5.12: Confusion matrix displaying results of a decision tree classifier tested on bird and drone data frames.

Inclusion of Unknown Class

The decision tree was then tested on car and aircraft spectrograms, where 29% gave false positives. An 'Unknown' class was then introduced, whereby a target is

classified as unknown if the maximum score output by the classifier did not exceed a particular threshold. The threshold was varied from 0 to 1, and a ROC curve was produced to show the true positive rate and false positive rate (Fig. 5.13).

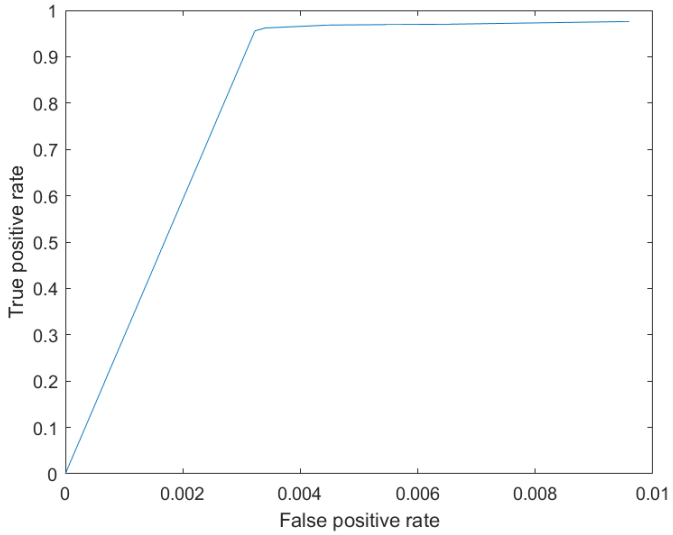


Figure 5.13: Receiver operating characteristic (ROC) curve displaying the relationship between the true positive rate and false positive rate of a decision tree classifier tested on bird and drone data frames.

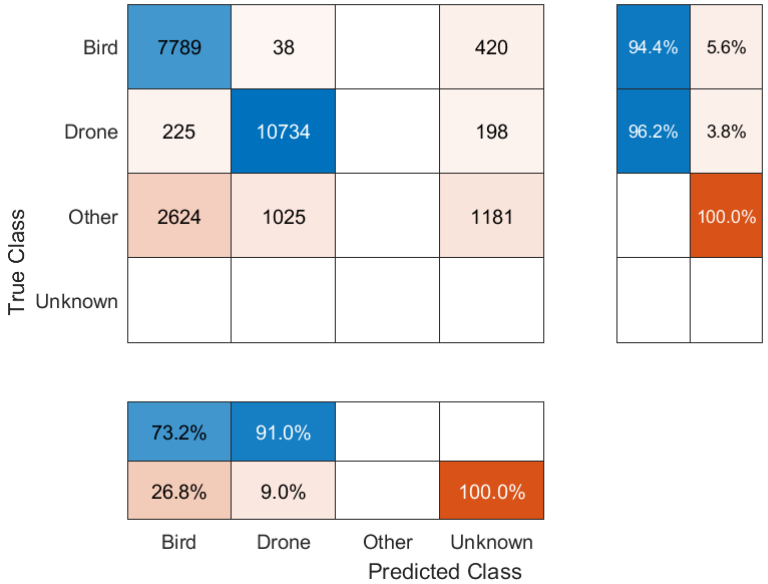


Figure 5.14: A confusion matrix displaying the results of a decision tree classifier tested on bird, drone, car, and aircraft data frames.

The optimal threshold is that which maximises the true positive rate whilst minimising the false positive rate, in this case, this is equal to 0.95. Fig. 5.14 shows the confusion matrix when targets are classified as unknown if the decision tree confidence is less than 95%. In this case, 21% of the non-biological confuser targets are misclassified as drone.

Effect of Noise on Decision Tree Performance

Data collected from drones flown at longer ranges or with a low RCS will have a lower SNR, making the micro-Doppler returns more difficult to detect. As demonstrated in [163], decision trees are sensitive to noisy test data, and the effect of noise on performance depends on the strength of the relationship between the variable and class label. The number of Doppler components is the feature that has the biggest impact on the prediction, therefore one can expect noise in this variable to have a significant impact upon performance.

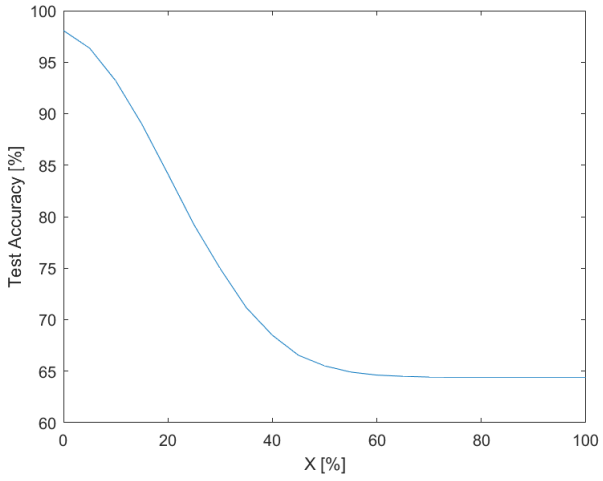


Figure 5.15: The accuracy of a decision tree classifier where X% of frames are corrupted, resulting in zero Doppler components measured, emulating performance on low signal to noise ratio test data.

In order to investigate the effect of SNR on performance, the number of Doppler

components feature is purposefully corrupted, following a similar process to [163]. For each test sample where the number of Doppler components is non-zero, the value is replaced by zero with a probability of X%. Fig. 5.15 shows the test accuracy as X is increased from 0 to 100, where performance decreases with increasing X, up to $X = 60$. Beyond this point, performance plateaus.

In [163], the author demonstrated that it is counter-productive to remove the noise in the training data set if noise is expected to be present in the test data. "It would seem that the tree-forming algorithm tends to avoid using noisy attributes. but this feature is nullified if the training set is noise-free and so gives no indication of the noise to be expected in practice" [163]. It is therefore hypothesised that introducing noise in the training data set will improve performance on noisy test data. In order to test this theory, the decision tree was retrained after corrupting the number of Doppler components in both the training data and test data. In this case, X was chosen to equal 60.

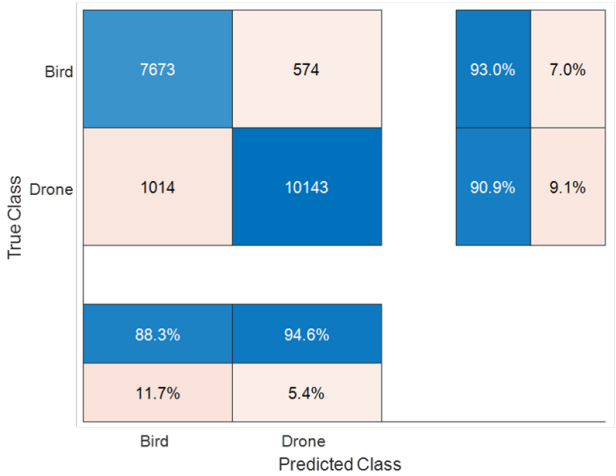


Figure 5.16: A confusion matrix displaying the results of a decision tree classifier tested on bird and drone data frames, where 60% of the non-zero number of Doppler component values were set to zero, to emulate performance on low signal to noise ratio test data.

Fig. 5.16 shows an accuracy of 91.8%, which is an improvement of almost 42%

over the case where the decision tree was trained on the uncorrupted training data. The reason for this is that the decision tree is less reliant on the noisy feature, as demonstrated in Fig. 5.17, supporting the findings of [163][164]. This suggests that performance can be improved by training on imperfect data, if the test data is expected to contain a significant amount of noise.

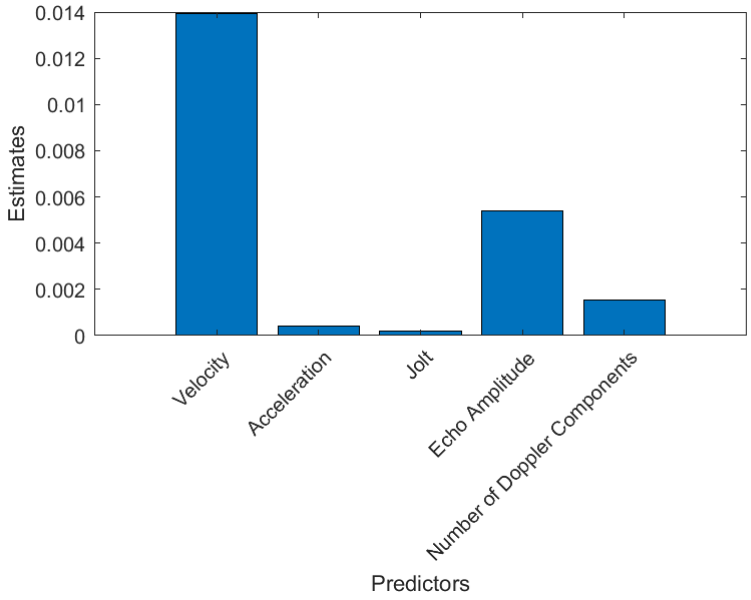


Figure 5.17: A bar chart showing the ranking of decision tree predictor importance, where the number of Doppler components feature has been corrupted prior to training.

Finally, the decision tree was retrained and tested on only four features, after removing the number of Doppler components entirely. Fig. 5.18 shows the resulting confusion matrix, where an accuracy of 90.2% was achieved. In this case, performance is improved over the case where the model was trained on uncorrupted data and tested on corrupted data, and is slightly lower than the case where the tree was trained on corrupted data. This is because the number of Doppler components still contained some useful information, as only 60% of the non-zero values were corrupted. Instead, the classifier has to rely more on the velocity and echo strength (Fig. 5.19), which are not sufficient to separate the two classes (Fig. 5.8).

This demonstrates that simply removing noisy variables will not necessarily lead to improved performance, provided there is still useful information to be extracted.

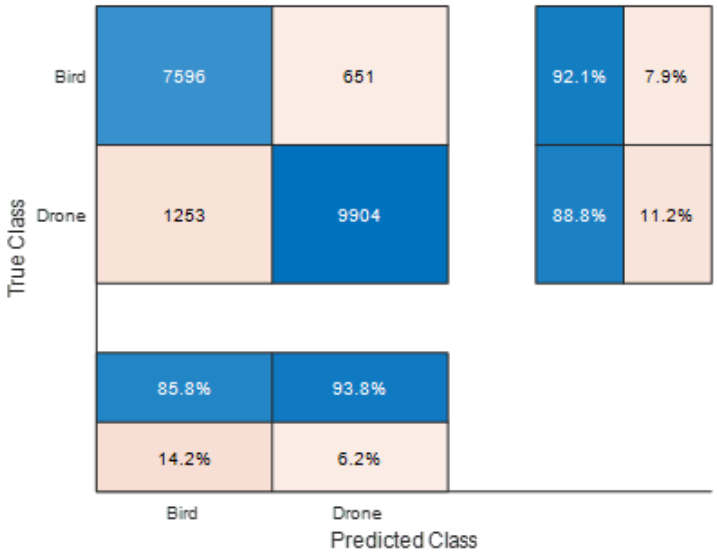


Figure 5.18: A confusion matrix displaying the results of a decision tree classifier tested on bird and drone data frames, where the micro-Doppler feature was not used.

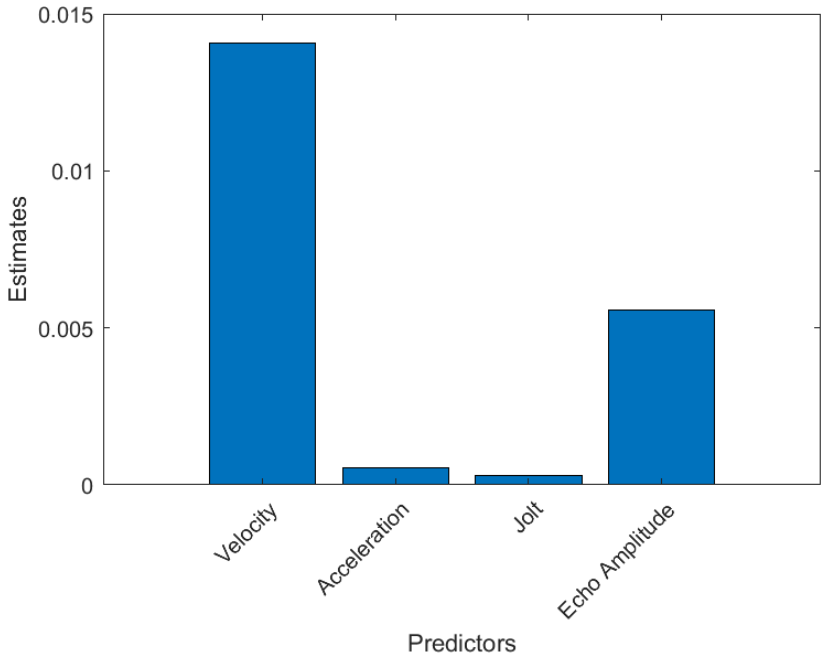


Figure 5.19: A bar chart showing the ranking of decision tree predictor importance, where the micro-Doppler feature was not used.

5.4 Convolutional Neural Network Performance

A CNN was trained on a set of bird and I1-D drone spectrograms, where each spectrogram contained 40 spectra (corresponding to approximately 11 seconds in time). 60% of the spectrograms were used for training, 30% for testing, and the remaining 10% for validation. The network was based on the architecture of AlexNet [8], where transfer learning was used to reduce the training time. AlexNet's weights were used to initialise the network, then every layer was retrained using the training spectrograms.

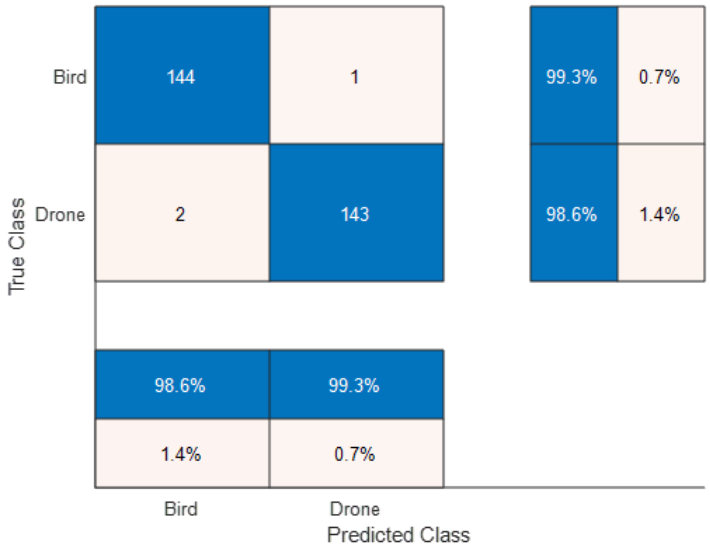


Figure 5.20: A confusion matrix displaying the results of a convolutional neural network (CNN) classifier tested on bird and drone spectrograms.

Fig. 5.20 shows that the CNN performs with a high accuracy, correctly classifying 98.6% of the drone spectrograms and all but one of the bird spectrograms. The CNN was then tested on car and aircraft spectrograms, where 37% gave false positives. An 'Unknown' class was then introduced, and Fig. 5.21 shows the resulting ROC curve, where the optimal 'Unknown' threshold is 0.998, leading to a true positive rate of 0.95 with a false positive rate of 0. Fig. 5.22 shows the confusion matrix when an unknown class is included. In this case, 24% of the non-biological confuser

targets are misclassified as drone. This is an improvement over the case where there is no 'Unknown' class, but is still high, highlighting the importance of pre-filtering the target returns to restrict the input to the CNN to birds and drones only.

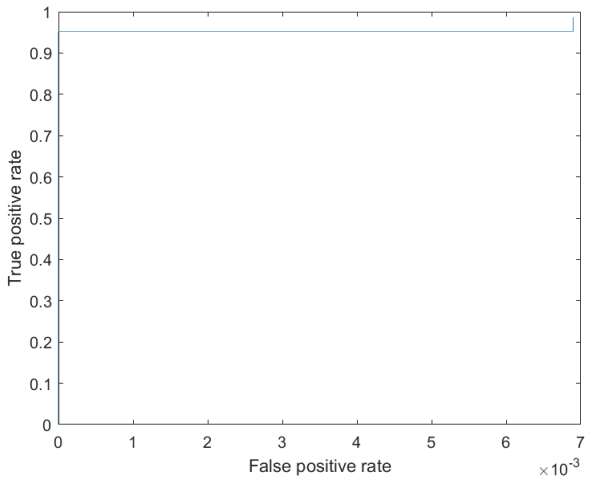


Figure 5.21: A ROC curve displaying the results of a convolutional neural network (CNN) classifier tested on bird, drone, car, and aircraft spectrograms.

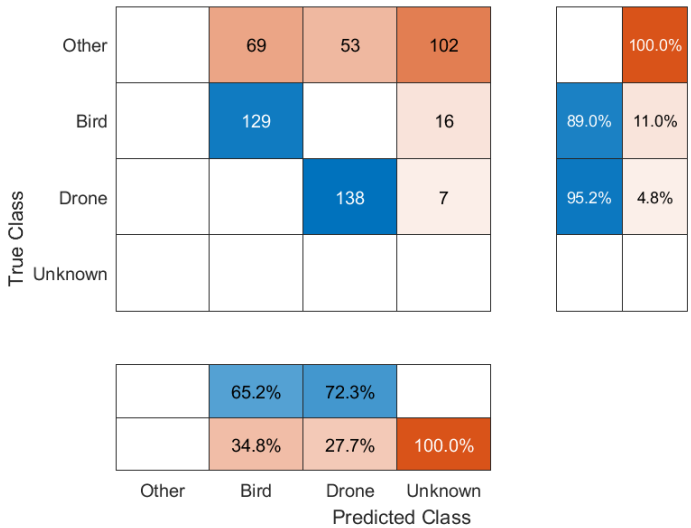


Figure 5.22: A confusion matrix displaying the results of a convolutional neural network (CNN) classifier tested on bird, drone, car, and aircraft spectrograms.

5.4.1 Comparison With Decision Tree Benchmark

The decision tree classifies on a per-frame basis, whilst the CNN classifies entire 40 frame spectrograms. To allow meaningful comparison between the two models, the decision tree was retrained and tested on the same data sets as the CNN, with the exception of a single bird track for which the feature data was corrupted. A majority-rules approach was implemented into the decision tree framework to output a single label across 40 data frames, based on the modal output. As failing to identify a potentially harmful drone could have severe consequences, the algorithm was instructed to classify the target as a drone in the case of a draw (although this situation did not arise in practice).

Fig. 5.23 shows the results of testing the decision tree, where all test samples were correctly classified. The performance of the CNN (Fig. 5.20) and decision tree are comparable. If an unknown class is introduced, the maximum score threshold for the decision tree to achieve perfect performance on the bird and drone spectrograms is 0.507.

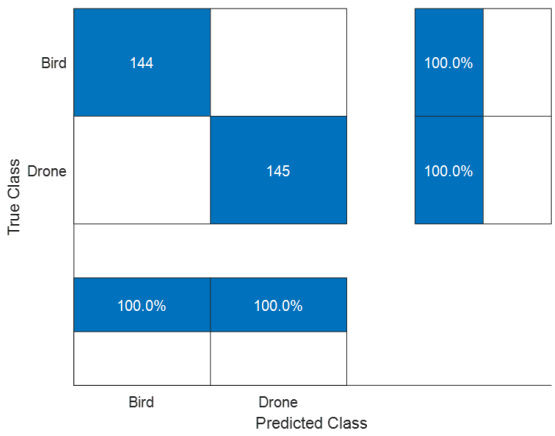


Figure 5.23: A confusion matrix displaying the results of a decision tree classifier tested on bird and drone frames, using a majority-rules approach across 40 data frames.

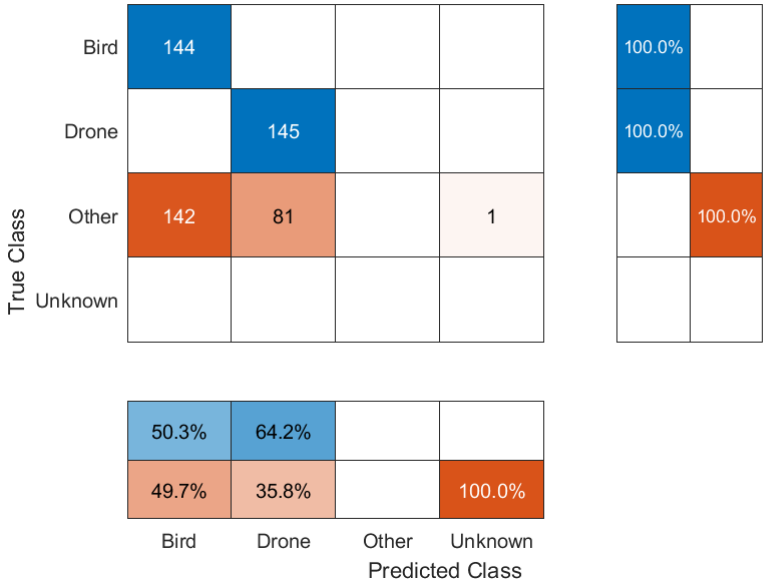


Figure 5.24: A confusion matrix displaying the results of a decision tree classifier tested on bird, drone, car, and aircraft frames, using a majority-rules approach across 40 data frames.

Fig. 5.24 shows the resulting performance on the 'Other' class, where 36% of non-biological confuser targets are misclassified as drones, compared to 24% for the CNN.

5.4.2 Other Machine Learning Algorithms

Decision trees are interpretable, but they can become very deep making them prone to overfitting, leading to a lack of generalisation and interpretability. A number of other machine learning algorithms (including a Naive Bayes classifier, support vector machine, and K-Nearest neighbour algorithm (where $k = 3$)) are also trained for comparison. Table 5.2 displays the accuracy of each of the six machine learning algorithms considered. Performance is comparable for all six classifiers in the case where all features are used. Performance falls across all classifiers when the micro-Doppler feature is removed from the training and test subsets, demonstrating that

this is a key classification feature. The decision tree classifier is the most robust classifier, and has the added benefit of being particularly explainable.

Table 5.2: A table comparing the performance of six machine learning algorithms (with and without micro-Doppler features) and a CNN, for bird-drone classification.

Model	Accuracy [%]		
	Kinematic Features + microDoppler Feature	Kinematic Features	Spectrogram
K-Nearest neighbour	98.4	86.0	-
Naïve Bayes	98.4	87.4	-
Support vector machine	98.4	90.4	-
Linear discriminant analysis	95.6	90.2	-
Quadratic discriminant analysis	98.4	87.8	-
Decision tree	100.0	96.2	-
CNN	-	-	99.0

When classes are clearly separable in feature space, simple classifiers are sufficient to achieve high classification performance. Classifiers trained on a small number of simple kinematic and micro-Doppler features show high performance. However, performance degrades when the micro-Doppler signatures are not detectable, and the model becomes more dependent on kinematic features. Drones are capable of a wide variety of manoeuvres, even mimicking the flight styles of birds [165], complicating separation based on kinematics alone.

The downside of traditional machine learning approaches is that a manual selection of features is required, where the choice of features is not necessarily obvious. Deep learning algorithms such as CNNs extract deep features directly from the data, and draw non-linear boundaries in feature space, potentially increasing separability of classes leading to improved classification performance on hard-to-classify samples.

The performance of the CNN is comparable that of machine learning classifiers, demonstrating that deep learning algorithms such as CNNs show promise in the domain of drone classification. However, CNN performance has only been established on a relatively small data set, with a mean SNR of 44 dB. The data set used for training and testing consists of only a single model of drone, at ranges that permit visible HERM lines. In order to explore the limitations of the classifier, performance will be investigated as a function of SNR.

5.5 Low SNR classification

In Section 5.3.2 it was demonstrated that decision tree performance is expected to fall with decreasing SNR - this result can be expected to hold up regardless of the classifier used. It is suggested that training exclusively on high SNR data will lead to over-reliance on features that may be noisy in the test data set, resulting in poor performance. A method of mitigating this effect is to train on similar data to that which will be used for testing in practice, therefore, it is hypothesised that CNN range dependence could be reduced by training on lower SNR data. It is further suggested that the training set should contain some high SNR data, to facilitate classification of high SNR test data. In this section, these hypotheses are tested by adding noise to the spectrograms used for training and testing a CNN.

5.5.1 Methodology

SNR Definition

In this chapter, the SNR is defined as the average ratio of the body return to the average noise floor. Fig. 5.25 shows an example drone Doppler spectrum, where

a high pass filter with a cut-off frequency of 15 Hz has been applied to remove stationary and low velocity clutter [166]. The contribution of the drone body is circled in red, and the bins over which the noise floor is estimated are shown in black. The signal to noise ratio for this frame is 60.5 dB. The SNR is then averaged over all frames within the spectrogram to calculate an SNR per spectrogram. Fig. 5.26 shows the evolution of target power, background power, and SNR over a drone track, with a mean SNR of 53.2 dB. Note that the SNR of the HERM lines is typically in the region of 15 dB lower than that of the body return.

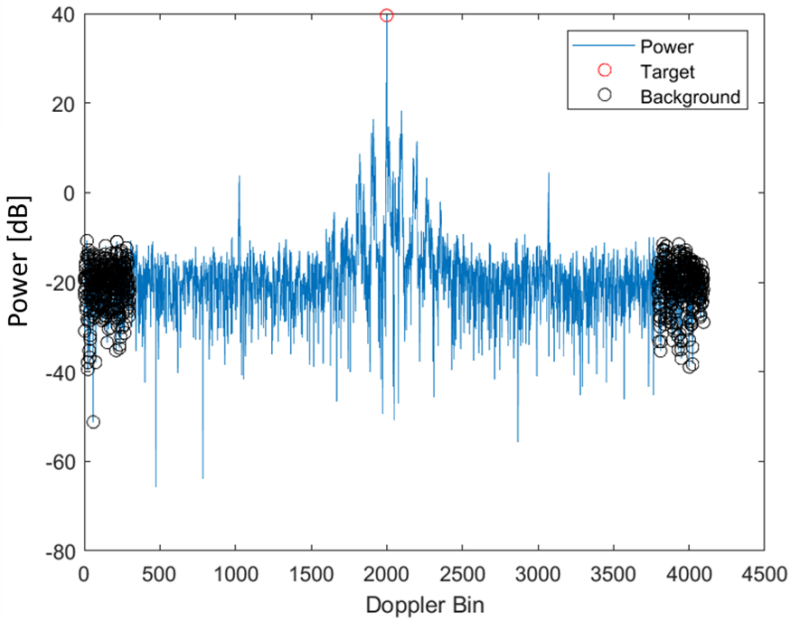


Figure 5.25: Doppler profile showing the power, target return, and background bins used for calculating the signal to noise ratio.

SNR Reduction

To simulate a reduction in SNR, Gaussian distributed noise is injected into the real and imaginary components of the target signal, resulting in Rayleigh distributed noise power. The SNR of the data set outlined in Section 5.2.2 is reduced in increments of 3 dB up to a 24 dB reduction, corresponding to an approximate quadrupling of range through the radar range equation. This results in nine data sets that are further split into training, test, and validation subsets. The nine data sets differ

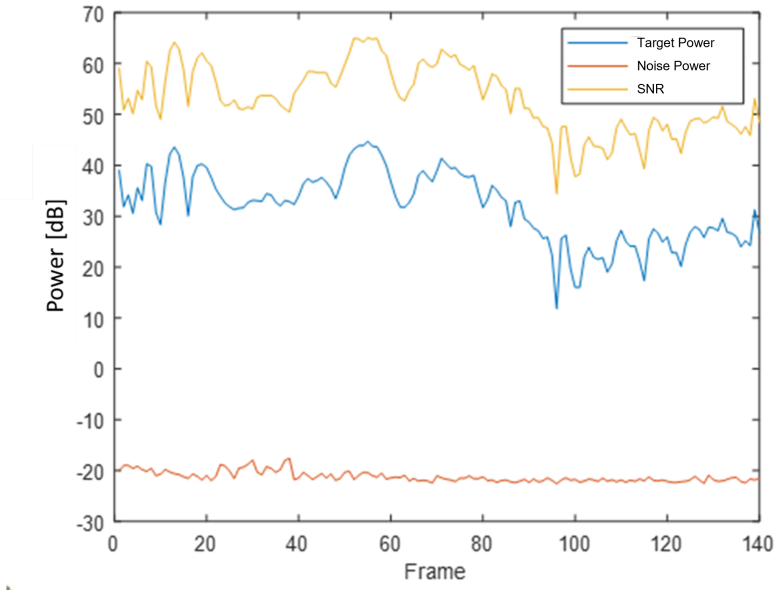


Figure 5.26: Target power, background power, and signal to noise ratio (SNR) over a drone track.

only in SNR, and the spectrograms are allocated to training and testing subsets independently of SNR.

Hyperparameter Optimisation

Model hyperparameter values are tuned to maximise classifier performance in a process known as hyperparameter optimisation. Due to the black box nature of CNNs, the impact of hyperparameters on performance must be approximated through repetition of the training and testing process rather than calculated explicitly. Methods such as grid search and random search [167] are computationally expensive and time consuming, particularly when many hyperparameters are to be considered.

Bayesian optimisation [168] is a method that is commonly used to optimise the output of a black box problem by tuning the hyperparameters. First, the classifier is trained using a set of initial parameters to estimate the objective function - that is, the function to minimise. In this work, the objective function is the performance

loss on a set of test data, and the expected improvement [169] is used as the acquisition function to select the next set of hyperparameters for training. The expected improvement is used as it strikes a compromise between investigating unexplored parameter-space and investigating regions where the objective function is estimated to be optimal. After training the model with the new hyperparameter values, the posterior distribution is updated and the process is repeated.

In [170], Bayesian optimisation is used to tune the hyperparameters of a CNN trained on the CIFAR-10 dataset (commonly used for training and testing machine learning algorithms, this dataset comprises of 60,000 images across 10 classes [171]). Parameters such as the learning rate, number of training epochs and pooling parameters were tuned, and it was demonstrated that Bayesian optimisation tuned parameters fastest resulting in an accuracy greater than the state of the art at the time. A Bayesian optimisation algorithm was also used in [172] to tune CNNs trained on the CIFAR-10 [171] and MNIST [173] databases, optimising decay rate, learning rate and mini batch size.

5.5.2 Testing on Low SNR Data

A training subset of unmodified (high SNR) data was used to train the CNN, and performance was evaluated on test subsets of spectrograms with varying levels of noise added.

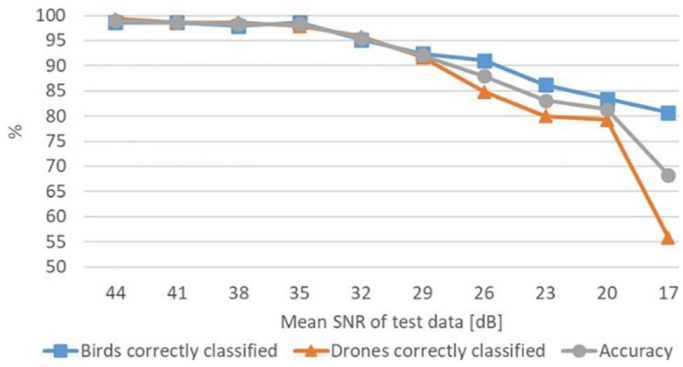


Figure 5.27: Drone recall (orange), bird recall (blue), and accuracy (grey) of a convolutional neural network (CNN) trained on high signal to noise ratio (SNR) data, as the SNR of the test data is decreased in increments of 3 dB [7].

Fig. 5.27 demonstrates that as more noise is injected into the signal, accuracy falls. Therefore, there will be a point beyond which SNR is too low for the classifier to reliably discriminate between drones and birds. A 12 dB reduction in test data SNR (corresponding to a doubling in range) results in a decrease in accuracy from 99.0% to 95.5%, whilst reducing the SNR by 24 dB results in an accuracy of 81.3%. Almost half of the drones are misclassified upon further reduction of SNR.

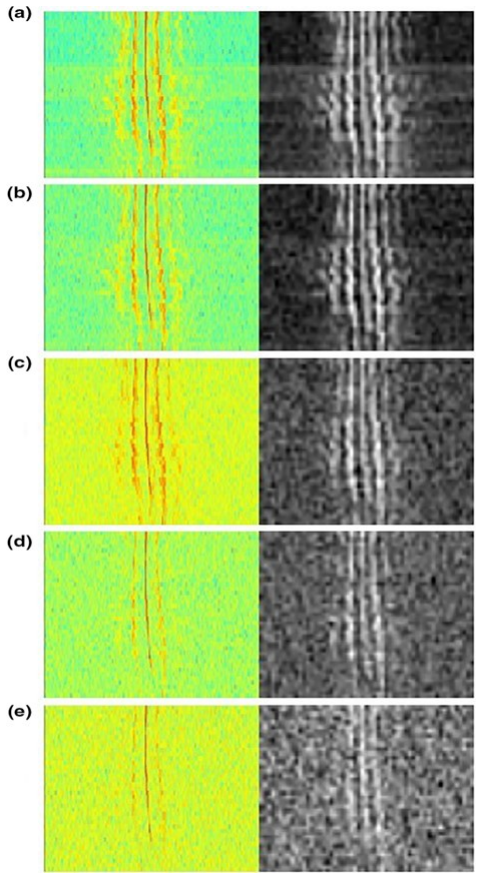


Figure 5.28: Maximum activations in the first convolutional layer when testing a convolutional neural network (CNN) on a drone spectrogram, where white indicates areas of high activation. (a) 46 dB signal to noise ratio (SNR), (b) 40 dB SNR, (c) 34 dB SNR, (d) 26 dB SNR and (e) 20 dB SNR [7].

A reduction in test data SNR has a greater impact upon drone recall compared to bird recall. This suggests that the classifier may be overly reliant on the presence of HERM lines for drone classification, impeding performance at low SNR when the HERM lines are not visible. It may be more challenging to classify drones that have certain properties, for example a fixed wing drone model with a single

rotor that has a relatively low RCS and angular velocity. Further, the classifier is at risk of exploitation by malicious drone operators that have intentionally reduced blade RCS. These findings support the initial hypothesis that CNNs will become less effective as SNR is decreased. Viewing the activations from the first convolutional layer (Fig. 5.28a-e) shows that the filters most strongly activated by the input images are those which detect HERM lines, further supporting the hypothesis that HERM lines are a key feature for classification.

5.5.3 Network Architecture

In this chapter, transfer learning has been used alongside AlexNet to establish CNN performance for drone vs bird classification, however, there are many other well-established CNN architectures to choose from. In this section, the performance of six different networks are compared, firstly to assess suitability for classification in the radar domain, and secondly to assess robustness to low SNR test data.

Broadly speaking, the networks that perform best in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) [116] are very deep (e.g. Inception-v4 [174], VGG-16 and VGG-19 [175]). Seven high performing models were selected for evaluation in [176], for training on camera images across only three classes - drone, bird, and no target present. The authors demonstrate that the shallower networks (AlexNet [8], Resnet [11], and Squeezenet [9]) performed better than their deeper counterparts when the number of classes was small, contrary to the performance shown in ILSVRC where there were over 1000 classes.

Six networks (AlexNet [8], Squeezenet [9], Inception-v3 [118], Resnet-18 [11], Resnet-50 [11], and GoogLeNet [10]) were selected from the list of pre-trained CNNs available in MATLAB [14] for performance comparison in the radar domain. The

networks were chosen across the spectrum of accuracy versus prediction time, whilst minimising the network size as shown in Fig. 5.29 taken from [14].

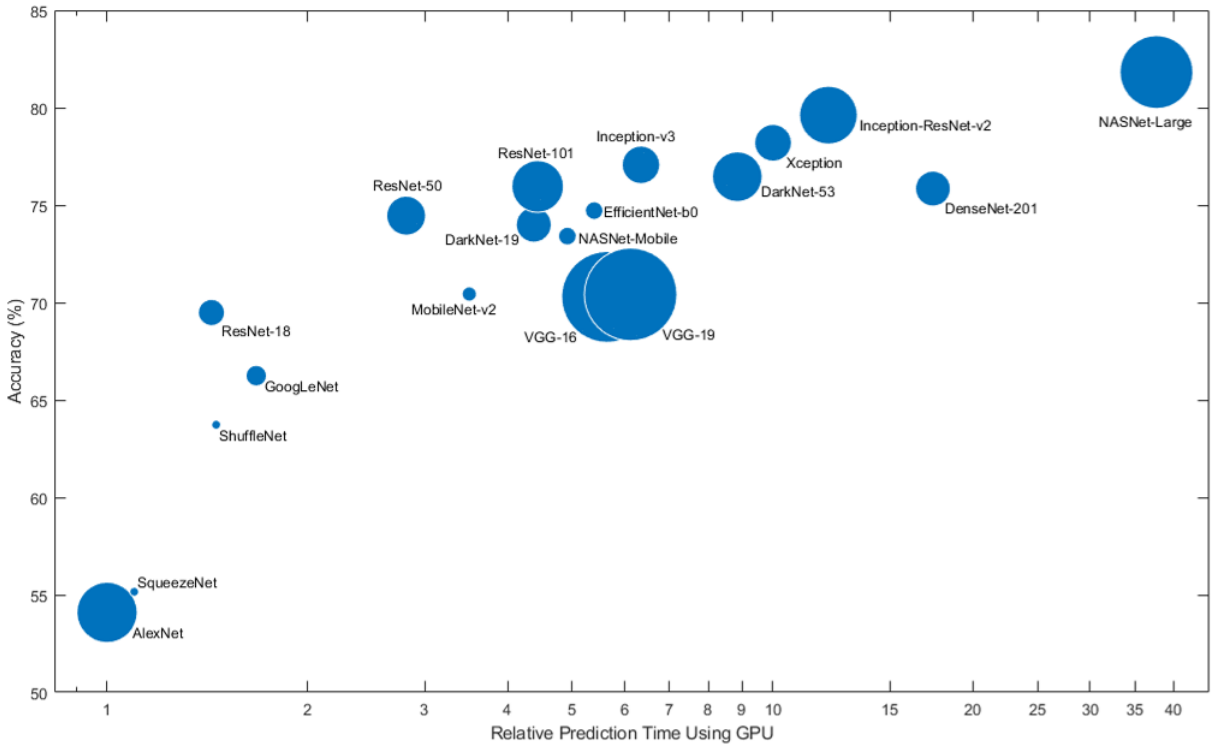


Figure 5.29: Accuracy against prediction time for the pretrained image classifiers available in MATLAB, taken from [14]. The area of each marker represents network size.

The six networks were trained on the original (high SNR) bird and drone spectrograms, and tested on the reduced SNR test subsets. Fig. 5.30 shows the accuracy of each network as a function of test SNR. All six networks achieve high performance on the original test data. It is not until SNR is reduced by approximately 12 dB that architecture has an impact on performance. AlexNet is the highest performing classifier at low SNR, maintaining an accuracy of 80% despite a 24 dB reduction in SNR. The poorest performing networks are Resnet-50 and Inception-v3, which are also the two deepest networks, supporting the findings from [176]. For the deeper networks, performance is comparable to randomly guessing the target class. The results indicate that the deeper networks are more prone to overfitting on the training data set, preventing generalisation to noisier test data.

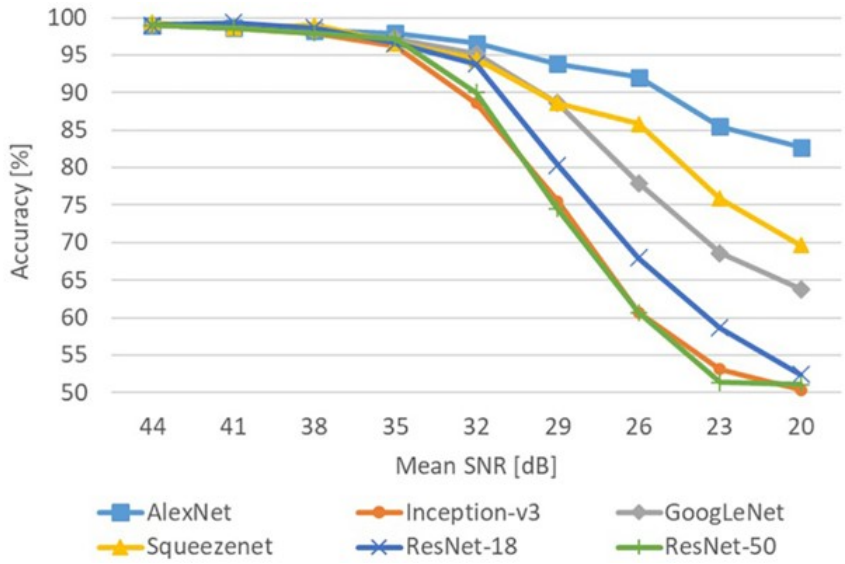


Figure 5.30: Classification accuracy as a function of mean test data signal to noise ratio (SNR) for six network architectures [7].

Table 5.3: Comparison of the depth, number of parameters, and training times of six different CNN architectures [7].

Network	Number of layers	Number of learnable parameters [million]	Training time [s]	Inference time per 100 spectrograms [s]
AlexNet	8	61	2029	2.8
Squeezenet	18	1.24	1668	2.9
Resnet-18	18	25.6	3821	4.8
GoogLeNet	22	7	4136	6.9
Inception-v3	48	23.9	16808	15.7
Resnet-50	50	44.6	10842	9.7

Computational cost, memory requirements, and ease of training will all be factors influencing the choice of CNN architecture, alongside performance. Table 5.3 summarises the number of layers in the network, the number of parameters, training time, and inference time, for each of the six networks considered. Whilst choice of network will depend on the requirements of an individual application, the results

indicate that AlexNet is a good choice of classifier for low SNR data, balancing high performance with the lowest training and test times. AlexNet is used as the network architecture for the remainder of this work.

5.5.4 Training and Testing on Similar SNR Data

In Section 5.3.2 it was demonstrated that training on data with similar SNR to the test data improves decision tree robustness. To investigate whether the same is true for CNNs, AlexNet was trained on each of the nine training data sets, testing the model on the test subset with a similar mean SNR. Comparison between Fig. 5.27 and Fig. 5.31 shows that the effect of decreasing performance with SNR is less pronounced when the classifier is trained on data of a similar SNR. Upon reducing the SNR by 24 dB, test accuracy is reduced by only 3.4% compared to a 19% reduction when training on the original data set. This is because the training spectrograms are more representative of the test conditions, preventing overfitting on HERM lines during training. Therefore, it may be possible to extend the range at which reliable classification is possible by ensuring that data of a similar SNR is represented in the training data, extending the range of potential applications.

It is suggested that training on a range of SNRs will produce a classifier that is robust across a span of ranges, without necessitating the training of a large number of classifiers. Whilst training on lower SNR data improves robustness, training exclusively on low SNR data will lead to poorer performance under ideal high SNR conditions [7]. Therefore, there is a balance to strike between preventing overfitting and enabling the extraction of discriminating features when conditions allow.

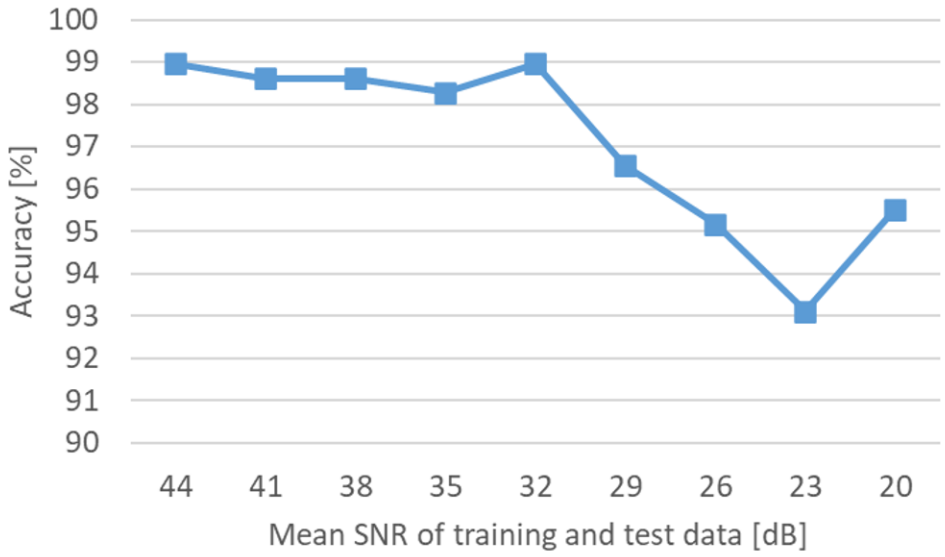


Figure 5.31: Classification performance when training and testing AlexNet on data of a similar signal to noise ratio (SNR) [7].

5.5.5 Augmentation for Robust Classification

In machine learning, perturbing the training data slightly can reduce overfitting and improve generalisation - this is known as data augmentation. For example, geometric transformations (e.g. scaling, cropping, and rotation) and colour space transformations (e.g. changing brightness, contrast, and saturation) are commonly used techniques when working with CNNs [177]. Taking inspiration from this, the proposed method of training a more robust classifier is to augment the training data set by injecting varying amounts of noise into the training spectrograms, simulating training across a spread of ranges. This method introduces additional hyperparameters for training, namely, the amount of noise to add during training, and how many of the training images to corrupt.

Uniform Sampling

To establish a performance baseline, a classifier was trained on a data set composed of equal samples from each of the training subsets. That is, each spectrogram was

represented at most once in the training data, at a randomly selected noise level. The test subsets were then used to test the classifier, with the results shown in Fig. 5.32. Whilst accuracy falls with decreasing SNR, fewer drones (0.7%) are misclassified at 20 dB compared to the original (Fig. 5.27, 20.7%), resulting in a classifier with significantly higher drone recall at lower SNR. However, this is at the expense of performance at high SNR, where the accuracy is 95.2% with a mean test SNR of 44 dB compared with 99.0% in the original case.

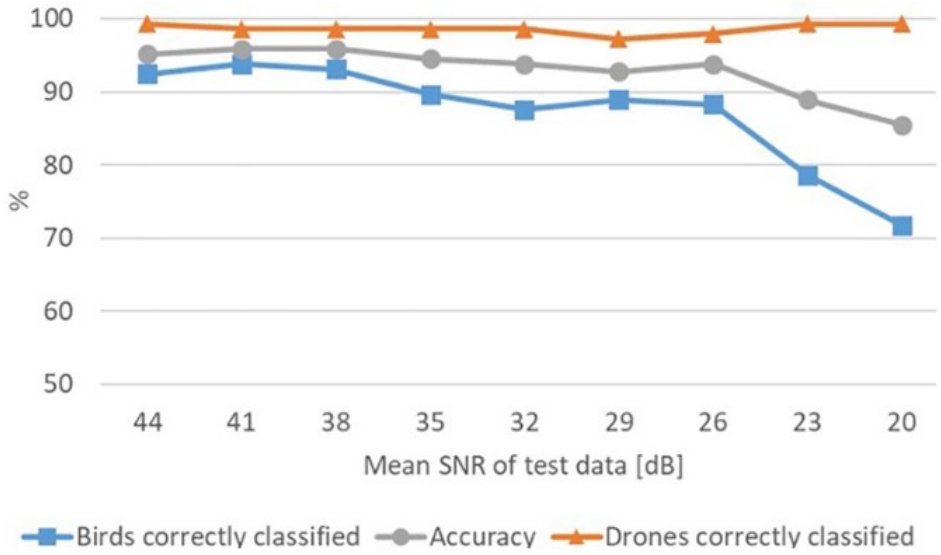


Figure 5.32: Convolutional neural network (CNN) classification performance showing drone recall (orange), bird recall (blue) and accuracy (grey) as a function of test signal to noise ratio (SNR), when each SNR bin is equally represented in the training data [7].

Bayesian Optimisation

Rather than sampling uniformly, weighting the training data towards a particular noise level may provide an optimal solution. Bayesian optimisation was used to find the percentage split across the training subsets that minimised performance loss on the test data.

Six hyperparameters were introduced to control the distribution of noise across

the training data SNR bins: $\%^{0}$, $\%^{6}$, $\%^{12}$, $\%^{18}$, $\%^{21}$, and $\%^{24}$, with the constraint that the sum of these values is equal to 100. After training the classifier with randomly initialised hyperparameter values, the loss is calculated on the test set and the Expected Improvement is used to choose the next set of hyperparameters.

Table 5.4 shows the optimal hyperparameters after 12 hours of calculation. The optimal solution is to reduce the SNR of almost 60% of the training samples by 12 dB, whilst only 2% of the training data is uncorrupted. Robustness to both medium and low SNR data can therefore be improved by adding a moderate amount of noise to the majority of training samples, as a CNN trained on lower SNR data learns not to rely solely on the presence or absence of HERM lines for classification.

Fig. 5.33 demonstrates high performance at high SNR, and only a moderate decline in performance with decreasing SNR. An accuracy above 90% is maintained despite reducing the test data SNR by 24 dB. Optimisation resulted in a more robust classifier compared to the case where the training data was equally split across the SNR bins, with an improvement in accuracy of over 10%.

Table 5.4: The optimal percentage split of the training data across various signal to noise ratio (SNR) bins for training a robust convolutional neural network (CNN), found using Bayesian optimisation.

Hyperparameter	$\%^{0}$	$\%^{6}$	$\%^{12}$	$\%^{18}$	$\%^{21}$	$\%^{24}$
Optimal value	2	7	57	4	28	1

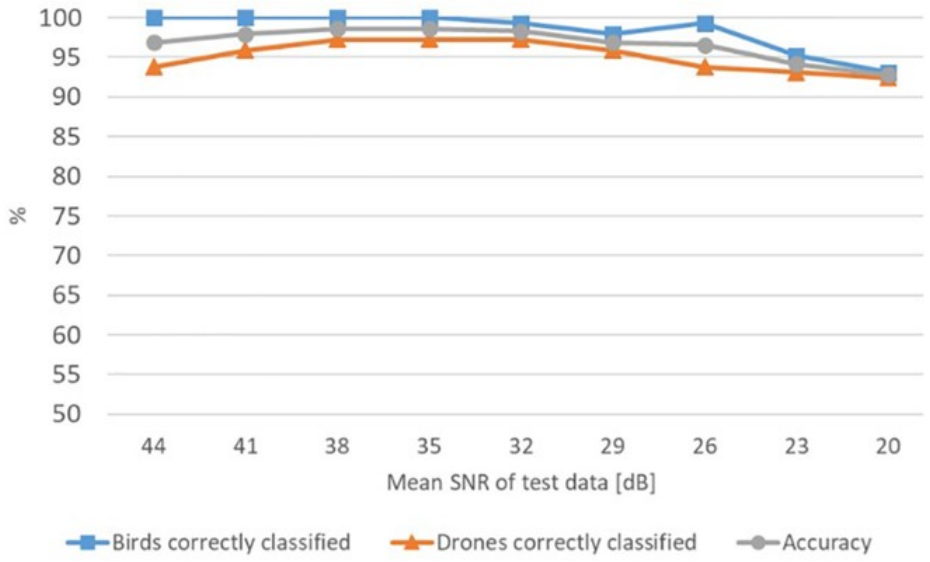


Figure 5.33: Convolutional neural network (CNN) classification accuracy showing drone recall (orange), bird recall (blue) and accuracy (grey) as a function of test signal to noise ratio (SNR), when the hyperparameters for training data augmentation have been found using Bayesian optimisation [7].

5.6 Chapter Conclusions and Future Work

5.6.1 Conclusion

A number of insights can be drawn from the work presented in this chapter, advancing the community's understanding of how CNNs may be used for discriminating between drone and bird spectrograms. These findings are the result of the first comprehensive investigation of CNN use for drone vs bird spectrogram classification under real operational conditions, whereby the classification problem is considered systematically. The following have been demonstrated:

- **The suitability of CNNs for drone vs bird classification.**

  A CNN achieved comparable performance (99%) to machine learning benchmarks (decision tree, Naïve Bayes, kNN, SVM, DA) when tested on an identical dataset, showing the potential of CNNs for the classification of drones using

radar.

- **An exploration of feature space of the drone vs bird classification problem.**
  The number of Doppler components emerged as a reliable predictor of target class for a decision tree, establishing a high performing baseline (98% accuracy).

- **Classification accuracy decreases with SNR.**
  Classifiers trained under ideal conditions are prone to overfitting on the presence of HERM lines, limiting drone classification performance at low SNR.

- **Methodologies for developing robust classifiers.**
  Techniques have been developed for reducing the impact of SNR on classification performance. Since SNR is affected by both RCS and target range, these results have significant implications for the conditions under which reliable classification can be performed. Maximising the range at which threats can be identified will give more time to form a response and allow for a broader range of applications.

5.6.2 Recommendations

Recommendations for training robust classifiers are as follows.

- **Consider a range of network architectures.**
  Of the CNN architectures considered in this chapter, the shallower networks were shown to be more robust to noisy test data than their deeper counterparts.

- **Train on a range of SNRs, either through direct measurement or by deliberately degrading training data acquired at a high SNR.**

Training on purposefully corrupted data has been shown to improve performance by 42% and 14% for a decision tree and a CNN respectively. An initial recommendation is to have 55%–60% of the training data at the halfway point between the minimum and maximum range at which classification is desired.

- **Bayesian optimisation can be used to maximise classifier robustness through the augmentation of training data.**
  This could be used to design a bespoke training strategy based on a specific application, mitigating the effect of low SNR upon performance.

5.6.3 Future Work

There are a number of natural extensions to the work presented in this chapter, as discussed below.

- **Validation of results using real low SNR data.**
  A key limitation of the work presented in this chapter is that simply adding noise to the training and test data is not equivalent to operating at longer ranges. Increased beam width, the presence of multiple targets and increased clutter will all affect performance. In Chapter 7, drone classification will be attempted on real data collected at ranges up to 5 km.

- **Acquisition of data from additional drone platforms.**
  The data set used in this chapter is limited in volume, and data consisted of just a single model of drone at a single location. The HERM line power depends on the RCS of the propeller, thus, as SNR is defined as the ratio of the body signal to the noise floor, HERM line visibility at a given SNR will be dependent on drone model. Therefore performance is likely to be more sensitive to SNR when considering a drone with a lower ratio of propeller

to body RCS. In Chapter 6 and Chapter 7, many models of drone will be considered.

- **Optimisation of the radar hardware or software to improve performance under low SNR conditions.**

  Changing the parameters used when collecting or processing the data could have a profound impact on spectrogram appearance, and enhancing the visibility of robust differentiating features could lead to a significant improvement in performance. For example, increasing the integration time (whilst maintaining coherence) will enhance SNR, improving the visibility of HERM lines and facilitating classification. The effect of integration time on performance is investigated in Chapter 6.

- **Creation of custom networks for classification of radar spectrograms.**

  The 'off the shelf' networks considered in this chapter are designed for classification in the optical domain and are not necessarily optimised for the radar domain. Further investigation into domain specific factors is strongly recommended, as better performance could be achieved by designing custom networks for the classification of drones. Further research in this area could have wider implications for classification of radar spectrograms, for example, the classification of human activity for medical or security purposes.

Chapter 6

Drone vs Drone Classification

6.1 Introduction

In the previous chapter, bird vs drone classification was explored in order to reduce false alarms from a drone surveillance perspective. Radar signatures can also be used to infer information about the type of drone present, in order to develop a more complete picture of airspace occupancy. The ability to classify drone models will aid in the risk assessment of potentially harmful drones, placing limits on the battery life, and the maximum payload that could be carried.

Features derived from drone micro-Doppler signatures can be used to perform coarse drone model classification [80][141][121][131][135]. Despite the high reported performance, these results are obtained at short distances, as low as 1.5 m, meaning that the SNR and Doppler sensitivity are high compared to those achievable under realistic conditions. In many cases drone returns are simulated, stationary in a laboratory environment, hovering, or are repeating the same linear trajectory [158]. The performance of drone classification algorithms under ideal conditions has been

shown to be formidable, and a natural extension is to assess drone classification performance in a realistic environment, which will form the objective of this chapter.

It is demonstrated in the literature that CNNs show great potential for use in drone model classification, using both real data and simulation [148] [178] [150] [146] [145] [147] [29]. In [148] and [178], CNNs are trained to perform drone model classification using radar spectrograms measured at X-band. The key difference in the work presented in this chapter (aside from using a Ku-band radar and collecting a more extensive data set obtained entirely under real conditions), is that the integration time is longer than the period of propeller rotation, meaning that the propellers show as HERM lines [77] [76] [4]. The shape and spacing of the HERM lines are determined by both the radar parameters (frequency, dwell time, etc.) and the target parameters (rotor length, rotational frequency, and RCS), therefore can potentially be exploited to aid classification [81][82].

Drone-to-drone classification is especially challenging, for a number of reasons. Firstly, several models of drone have similar RCS, meaning that classification cannot be performed using echo intensity alone. Drone RCS is low, therefore the resulting radar signatures have a low SNR which is a fundamental barrier to robust classification [7]. Drones are capable of highly dynamic flights, which limits the possibilities for classification based on kinematic features (e.g. trajectory, speed and its derivatives or acceleration). Drones also fly at relatively low altitudes, meaning that their signatures can become embedded in ground clutter, further complicating their detection and classification. A key enabler for drone classification is the presence of HERM lines caused by drone rotor blade rotation. However, as the RCS of blades is usually in the order of 15-20 dB below that of the drone body, acceptable SNR margins and the target distances at which they can be obtained are narrow, impeding classification [42][43][7]. In addition, drone manoeuvres and wind conditions add to the variability of HERM line behaviour, meaning that their micro-Doppler

returns (if visible) are variable for a number of reasons. Thus, there are a number of challenges to consider when performing radar classification of drones using data collected under real conditions.

In Chapter 5, it was demonstrated that CNNs are capable of extracting the spectral features of drone HERM lines, achieving accurate bird-drone discrimination under real operational conditions. In this chapter, a CNN is trained for drone-to-drone classification, and performance is assessed under both high and low SNR conditions. AlexNet is the CNN used in the remainder of this work. Spectrograms used for training and testing the classifiers in this work were provided by Plextek and obtained using a Ku-band staring radar operated in FMCW mode. The key contribution of this chapter is a data-driven, comprehensive and step-by-step analysis of a CNN algorithm for drone classification using a long integration time, identifying the capabilities and reporting on significant limitations [44]. Classification is performed at a variety of levels - from coarse drone group classification down to model recognition, to explore how finely it is possible to reliably classify. In [129], it was demonstrated that a two-stage decision tree reduced the false alarm rate in a drone vs non-drone problem compared to a single stage classifier, resulting in a low false alarm rate. In this chapter, various levels of hierarchy are considered for drone model classification, to investigate whether using multiple classifiers in series reduces the number of misclassifications. Performance is measured as a function of dwell time, ranging from a single frame to a whole track, and the time required to reliably classify drone group, subgroup, and model is estimated.

One might expect to achieve higher classification performance using finer Doppler resolution, allowing a higher degree of HERM line detail and better separation between the target and stationary or slow-moving clutter. In reality however, there will be cases where fast drone manoeuvres conducted within the coherent integration time will cause energy to spread along multiple Doppler bins, reducing SNR

resulting in signatures that differ substantially from those obtained under idealistic conditions. In this chapter, drone SNR is explored as a function of integration time, and the effect of integration time on classification performance is investigated.

The layout of the chapter is as follows: Section 6.2 gives an overview of the methodology followed, including the spectrogram parameters, CNN implementation details, and an overview of the training and test data sets. Section 6.3 presents the results of training a binary classifier to attempt drone group classification, between fixed wing drones and rotary wing drones. Section 6.4 presents the results of training classifiers to carryout drone subgroup classification based both on number of rotors and size of drone. In Section 6.5, the results of model-level recognition are presented, with multiple hierarchies investigated. In Section 6.6, the impact of processing interval on drone classification performance is explored. Finally, the chapter conclusions are given in Section 6.7.

6.2 Methodology

CNNs can not be interpreted directly, therefore the input to the classifier was carefully controlled and the corresponding outputs were examined to extract meaningful information. This inherently involved grouping data together in some cases, and separating or reducing them in others, with a particular focus on the overall classification performance.

6.2.1 System and Processing Parameters

The radar used for data collection was a Plextek Ku-band staring radar prototype in FMCW mode. This radar has a PRF of 6,617 Hz, and a maximum range of a few hundered metres, with an 8 metre range resolution.

Time series data in the form of I/Q samples were provided across nine different drone models, including fixed wing drones, quadcopters and hexacopters. MATLAB code was provided by Plextek, to process the data to form spectrograms for input to the CNN and power spectra for visualisation and SNR calculations.

The spectrograms were created using an STFT along time with overlapping segments, followed by a high-pass filter (with a cut-off frequency of 52 Hz) to reduce stationary and low velocity clutter. The SNR, time resolution, and frequency resolution are determined by the signal processing parameters, including the integration length used in the FT, the overlap between segments, and the weighting window (Fig. 6.1). The choice of processing parameters will therefore impact upon the classification performance, as explored further in Section 6.6.

A simple parameter sweep was conducted to find a compromise between SNR, frequency, and time resolution, where parameters were selected based on the resulting classification accuracy upon a subset of the full data set. For the data tested, and the parameters swept, a 512-point STFT with a 75% overlap and a Hanning weighting window was found to be most suitable. Under these parameters, the frequency resolution of the spectrograms is approximately 13 Hz (6,617/512). For a 75% overlap, there are 128 uncommon pulses between consecutive STFTs, giving a time resolution of 19.3 ms.

6.2.2 SNR Definition

Data were provided in frames of 1 second duration, equating to 6,617 pulses. On a power spectrum calculation via FT, the power of the drone body was measured by taking the maximum power in the 10 Doppler bins either side of the measured target Doppler bin, within the range and azimuth bin containing the target. The

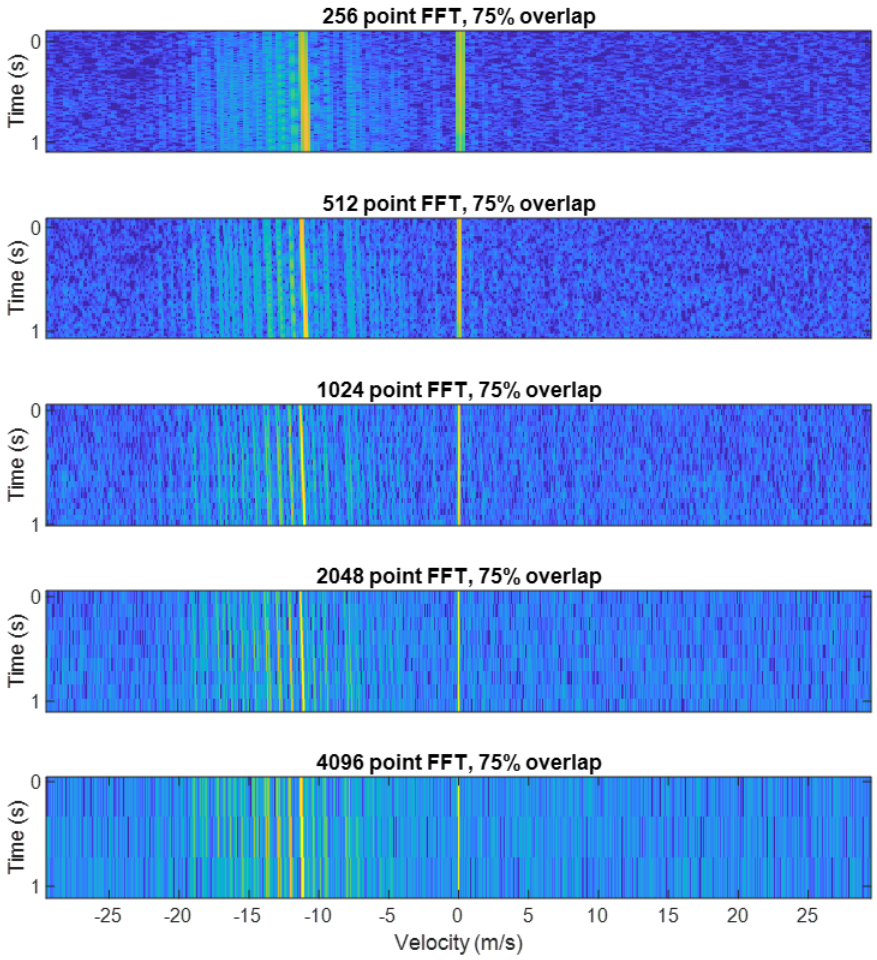


Figure 6.1: Example DJI Matrice 210 spectrograms, with a Fast Fourier Transform (FFT) length ranging from 256 pulses to 4096 pulses.

average background power is defined as the power at the target range, azimuth, and Doppler index, averaged throughout the file at the times where the target was not in the particular range index, at the output of a 6,617-point FT. The SNR is measured as the ratio of the drone body power to the average background power, in decibels. The SNR was calculated using MATLAB code provided by Plextek.

Under these conditions, the SNR included the signal processing gain arising from the application of a FT with an integration time of 1 second. It should be noted that the SNR in the spectrogram is different as 512-pulse STFTs are employed instead,

however to state that the SNR is nearly 11 dB less ($10log_{10}\frac{512}{6617}$) is misleading due to the inherent assumption of coherent integration gain. Moreover, the power of the HERM lines is different to that of the body, typically in the region of 15 dB less, although that is more difficult to measure automatically.

6.2.3 Data Coverage

The data spanned a variety of drone platforms, including fixed wing drones, quadcopters, and hexacopters. The models of drone and approximate size and weights are summarised in Table 6.1. Approximately 100 minutes of data were provided, and the amount of data collected for each model is summarised in Table 6.2, along with the mean SNR measured for each model. The mean SNR ranges from 39.9 dB for the Yuneec H920, which is the heaviest model of drone represented in the data set, to 25.1 dB for the Ares Gamma Pro V2, which is the lightest model considered. There is significant variation in the data belonging to each drone model, as various flight scenarios and velocities were covered in the experimental campaign.

Table 6.1: An overview of the models of drone for which data was obtained at Ku-band.

Model	Group	Subgroup	Diagonal Length /Wing span [m]	Weight [kg]
DJI Matrice 210	Rotary wing	Quadcopter	0.6	4.8
DJI Mavic 2 Pro	Rotary wing	Quadcopter	0.4	0.9
DJI Mavic Air 2	Rotary wing	Quadcopter	0.3	0.6
DJI Phantom 3	Rotary wing	Quadcopter	0.4	1.2
Dynam A10 Warthog	Fixed wing	Fixed wing	1.1	1.0
Ares Gamma Pro V2	Fixed wing	Fixed wing	0.9	0.5
FMS Easy Trainer	Fixed wing	Fixed wing	1.3	0.67
Yuneec H520	Rotary wing	Hexacopter	0.5	1.6
Yuneec H920	Rotary wing	Hexacopter	0.9	5.0

Table 6.2: A summary of the data set that was obtained at Ku-band.

Model	Amount of data [s]	% of Total	Mean SNR [dB]	Max SNR [dB]	Min SNR [dB]
DJI Matrice 210	732	12.0%	37.5	63.7	1.5
DJI Mavic 2 Pro	657	10.8%	30.1	62.2	2
DJI Mavic Air 2	499	8.2%	28.2	49.9	2.6
DJI Phantom 3	773	12.7%	35.1	67.7	-13.1
Dynam A10 Warthog	322	5.3%	30.6	52.6	-6
Ares Gamma Pro V2	478	7.8%	25.1	50.8	-6.6
FMS Easy Trainer	943	15.5%	29.6	53.2	3.9
Yuneec H520	825	13.5%	35.3	63.1	1.6
Yuneec H920	868	14.2%	39.7	71.7	-17.2

6.2.4 Example Spectrograms

In Fig. 6.2a-c examples of higher SNR spectrograms are given for three different drone models: a DJI Matrice M210 (quadcopter), a FMS Easy Trainer (fixed wing) and a Yuneec H920 (hexacopter). For comparison, Fig. 6.2d-f are examples of lower SNR spectrograms of the same drone models, where the HERM lines are noticeably less visible. There are other examples in the data set where HERM lines are not visible in the spectrograms at all, however this is expected to some degree due to the high degree of within-class variability in the data set due to manoeuvres and target ranges.

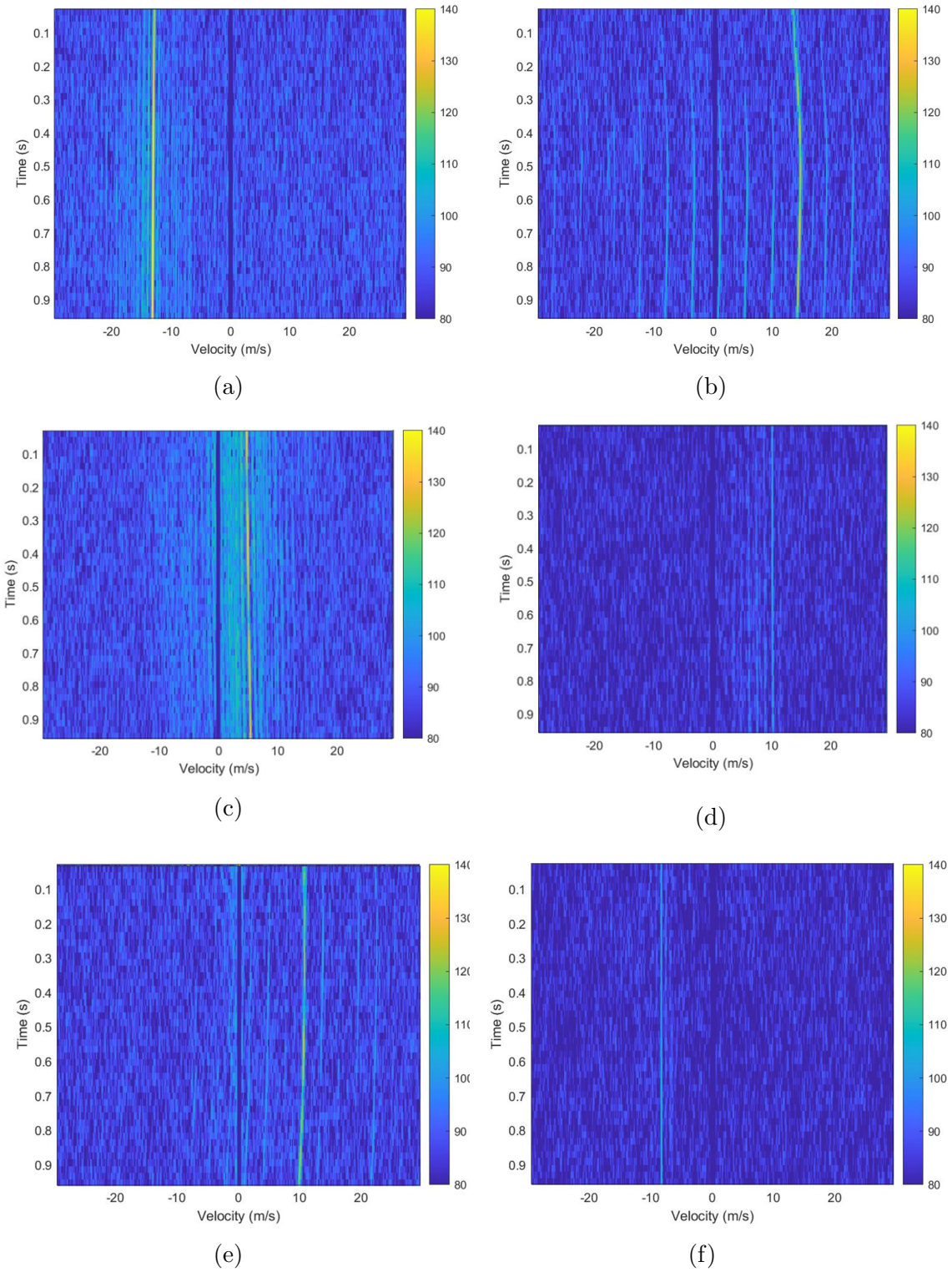


Figure 6.2: Example drone spectrograms a) High signal to noise ratio (SNR) DJI Matrice M210 b) High SNR FMS Easy Trainer c) High SNR Yuneec H920 d) Low SNR DJI Matrice M210 e) Low SNR FMS Easy Trainer f) Low SNR Yuneec H920.

6.2.5 Classification Approach

An AlexNet [8] architecture was used, along with transfer learning to initialise the network weights. This model has shown great promise for bird vs drone classification in Chapter 5, and is robust to noisy test data [43]. It must be noted however that the success of this algorithm was validated at L-band rather than Ku-band, and with a longer dwell-time and different processing parameters, therefore further work is needed to determine whether AlexNet is the optimal network architecture for this task.

Classifier performance is reported in full in confusion matrices and summarised using F1 scores. Here, the F1 score is defined as the harmonic mean of the macro-averaged precision and recall [179]. The F1 score has been defined as such in order to allow generalisation beyond the binary classification problem. ROC curves are presented to show the relationship between the false positive rate and true positive rate.

A systematic approach was followed to identify the capabilities and limitations of the proposed algorithm. The problem was split into 3 semi-independent tasks, namely:

- 2-class problem: Drone group classification.
- 3-class problem: Drone subgroup classification.
- 9-class problem: Drone model recognition.

Each problem was initially considered separately, although multi-stage classification approaches are considered in Section 6.5.2.

Evidence seems to suggest that major features for drone classification are the presence and characteristics of HERM lines due to drone blade rotation [136], [138], [29]. High SNR data would normally be used for training a classifier, to achieve maximum separation between the target classes. However, in order to maximise the applications for which it can be used, the classifier needs to be robust to classification of low SNR test data. In Section 5.5 it was demonstrated that CNNs are sensitive to test data SNR when performing bird vs drone classification, albeit with a different radar system and processing parameters. In this chapter, classifiers trained on high SNR data are tested on lower SNR data, to evaluate the dependence of the classifier's performance on SNR under real conditions. The results obtained on a high SNR baseline subset will be used as a basis for comparison with the more general case where SNR will be highly variable. Specifically, three cases were considered:

- "Baseline" case: A subset of the data where the SNR exceeded 30 dB was selected for training and testing.
- "Low SNR" case: The classifier was trained on data with an SNR above 30 dB, but tested on data whose SNR was lower than the threshold.
- "All data" case: Training and testing samples are randomly selected from the entire data set, independently of SNR. In this case, low SNR data can be used for training the classifier.

A trade-off was made between high SNR in the baseline subset and the abundance of training and testing data. A threshold of 30 dB was chosen empirically: a body SNR of 30 dB was found to be sufficiently high to result in HERM lines in the majority of spectrograms, and resulted in databases that were sufficiently large for training and testing the model (Fig. 6.3). An SNR of 30 dB is that of the drone body after a 1 second coherent integration, therefore the HERM lines are not guaranteed to be visible in the spectrograms.

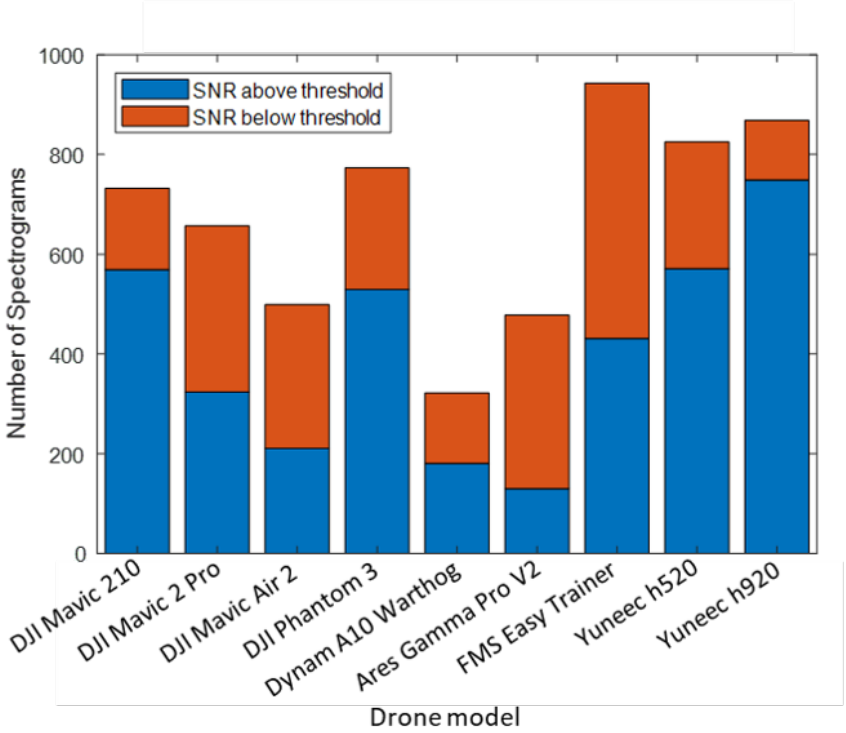


Figure 6.3: The number of spectrograms recorded with a signal to noise ratio (SNR) above and below the baseline threshold of 30 dB, for each model of drone.

The starting hypothesis was that the best classification performance would be achieved using the baseline case, where both training and testing data have similar and high SNR [160], followed by the all data case where low SNR data is used for training. The low SNR case was expected to result in the poorest performance. This hypothesis is tested and quantified throughout the chapter.

A data "file" was recorded for each track, typically ∼40 s long. Classification was attempted using either single 1 second spectrograms (a data "frame"), or sequences of all data frames within a data file by averaging the score output by the CNN over all of the frames within the file. The average score was then interpreted as the class label, resulting in a single output for each track. Per-file classification always outperformed the per-frame results, however, there is a loss of interpretability since the number of frames per file is not fixed. For this reason, the per-frame results are

included throughout the chapter and, where the per-file confusion matrices are not given in the full text, they are included in Appendix A. For each level of hierarchy, classification performance was also measured as a function of the number of frames over which the score was integrated, to estimate the time on target required for reliable classification.

In every classification problem, training and testing was repeated seven times, randomly selecting the files used for training, testing, and validation. The overall F1 scores presented are calculated from a confusion matrix that is summed across all of the experiments for that problem, but the spread of individual F1 scores are also presented throughout the text and in Appendix B. Effectively, this process enlarges the amount of training and testing data which is particularly important where data are split based on their SNR as this significantly reduces the amount of data available.

Weighted Loss Function

As shown in Table 6.2, there is an imbalance in the amount of data collected for each drone platform. If the training data is skewed towards a particular class, this could lead to bias. Data imbalances could be compensated for by under sampling the over-represented classes (although this means discarding some data), over sampling the underrepresented classes, recording or simulating additional data, or modifying the loss function to give equal weight to the classes overall.

In order to mitigate the effect of class imbalance, the loss function was weighted depending on the amount of training data belonging to each class, where samples belonging to under-represented classes were weighted more heavily than samples from over-represented classes. The loss function, L, is given by:

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K} w_i T_{ni} log(Y_{ni}), \tag{6.1}$$

where N is the number of observations in each class, K is the number of classes, w is the class weighting, Y is the predicted score, and T is the training score.

6.3 Drone Group Classification

The data contain signals from three fixed wing drone models and six rotary wing (quadcopter/hexacopter) drone models. AlexNet was trained to perform drone group classification. Training and testing frames were always sampled from different data files to avoid the classifier overfitting on target trajectory or background within the same file. 60% of the data files were used for training, 30% for testing and 10% for validation.

6.3.1 Baseline Case

The classifier was first trained and tested only on data with SNR >30 dB. The tracks are split into training, test, and validation sets, but the number of frames in a track varies from 6 to 88, with 35 frames on average. In each experiment, the files used for training, testing, and validation are randomly selected. Due to the stochastic nature of sampling, the number of frames/spectrograms in each subset for each model of drone can vary from one experiment to another. Table 6.3 shows an example of the number of frames in each subset for a single experiment in the baseline case.

Table 6.3: The number of frames used for training, testing and validation for drone group classification, where testing is performed on the baseline data set.

Set	Model								
	DJI Matrice 210	DJI Mavic 2	DJI Mavic Air 2	DJI Phantom 3	Yuneec H520	Yuneec H920	Ares Gamma Pro V2	FMS Easy Trainer	Dynam A10 Warthog
Training	337	183	110	257	382	394	70	232	85
Testing	197	102	81	212	155	267	44	121	68
Validation	32	39	20	60	34	88	16	78	28

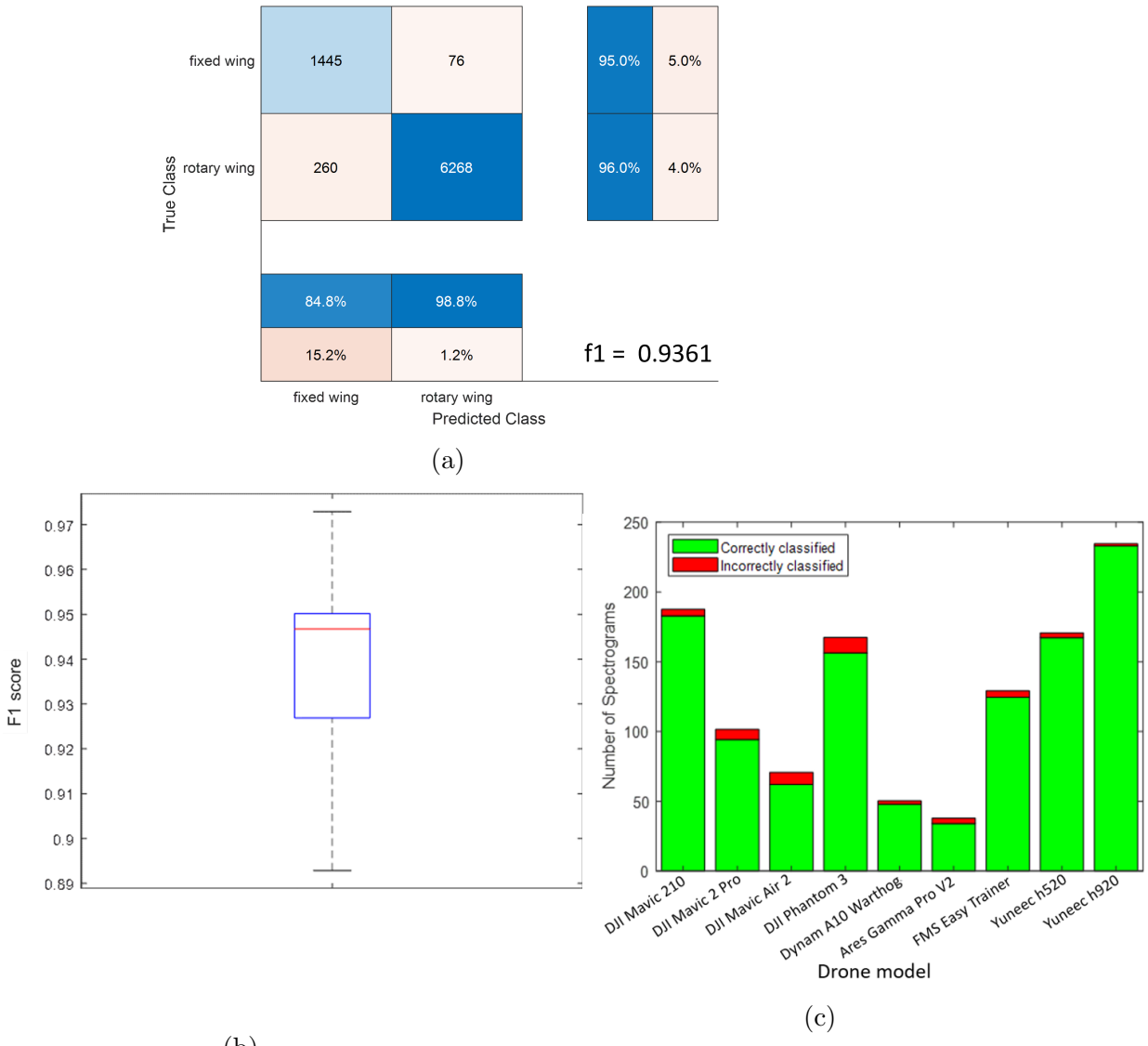
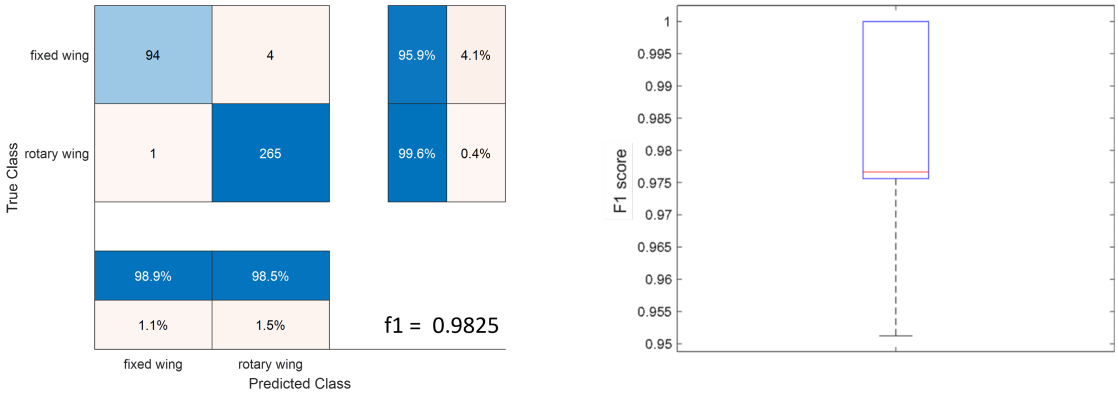


Figure 6.4: Drone group classification, tested on the baseline data set a) Confusion matrix, b) F1 scores, c) Classification outcomes per drone model.

Fig. 6.4a shows the performance of the drone group classifier on single frame (1 second long) spectrogram inputs, for the baseline case, summed over seven repetitions, with an F1 score of 0.94. Fig. 6.4b shows the span of individual F1 scores over the seven experiments, with the median F1 score shown as a red line. There is considerable variation in the F1 score due to three main factors: the particular samples in the testing subsets, and the network's approximation of the target function due to the training data samples and the stochastic nature of the training process. Fig. 6.4c shows the number of correctly classified spectrograms for each model of drone, averaged over the seven experiments. The Yuneec H920 model is classified most accurately out of all of the models, closely followed by the Yuneec H520.

Fig. 6.5a-b shows the resulting confusion matrix and span of F1 scores where classification is performed on a per-file basis, rather than for a single radar frame. Averaging the score over entire tracks leads to an improvement of 5% in the total F1 score. Integrating the result over a number of spectrograms increases the amount of information available to the classifier, as well as reduces the reliance upon a single output, resulting in fewer misclassifications overall.

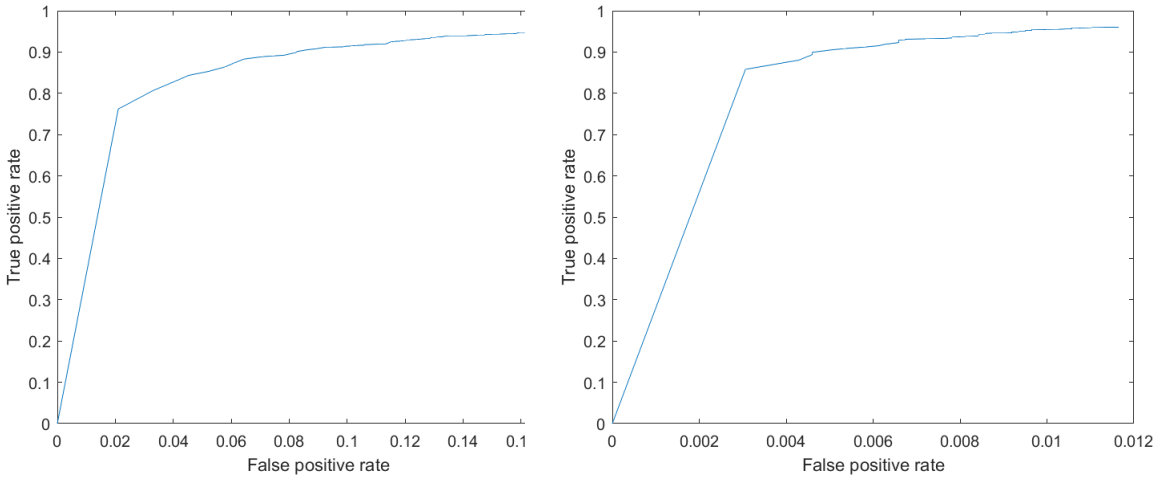


(a) (b)

Figure 6.5: Drone group classification, tested on the baseline data set, where classification has been performed on a per-file basis a) Confusion matrix, b) F1 scores.

An unknown class was included by placing a threshold on the scores. The re-

lationship between true and false positives was established through ROC curves, Fig. 6.6a-b. The results show that the CNN can discriminate between fixed and rotary wing drones with high accuracy (through the obtained F1 scores, Fig. 6.4) and confidence (through the ROC curves, Fig. 6.6). When including an unknown class, classifying based on a track rather than 1 second frames leads to a 5% increase in the F1 score. In this case, the improvement is modest although noticeable, since the F1 score was very high to begin with.



(a) (b)

Figure 6.6: Drone group classification receiver operating characteristic curves, tested on the baseline data set a) Fixed wing drone group, b) Rotary wing drone group.

6.3.2 Low SNR and All Data Cases

A reduction in SNR may compromise the visibility of the HERM lines, which are a major source of information about drone model and motion. One might expect the performance of a classifier trained exclusively on higher SNR data to drop when testing on lower SNR spectrograms if the HERM line features learned by the classifier are no longer visible.

The next step was to quantify the extent to which classification performance degraded when testing the classifier trained using high SNR data on low SNR spectrograms. Fig. 6.7a shows a confusion matrix for per-frame classification. Comparing Fig. 6.7a to Fig. 6.4a, the F1 score is reduced by 15%, which is substantial. The performance is clearly affected by SNR, however, an F1 score of 0.79 could still be deemed as acceptable depending on the application. The largest drop in performance is in the fixed wing drone recall, which is reduced by 27%. Classifying on a per-file basis shows a substantial improvement of 12% in the F1 score compared to classifying single frames. The results indicate that there is sufficient information for a CNN to discriminate between drone groups even at low SNR.

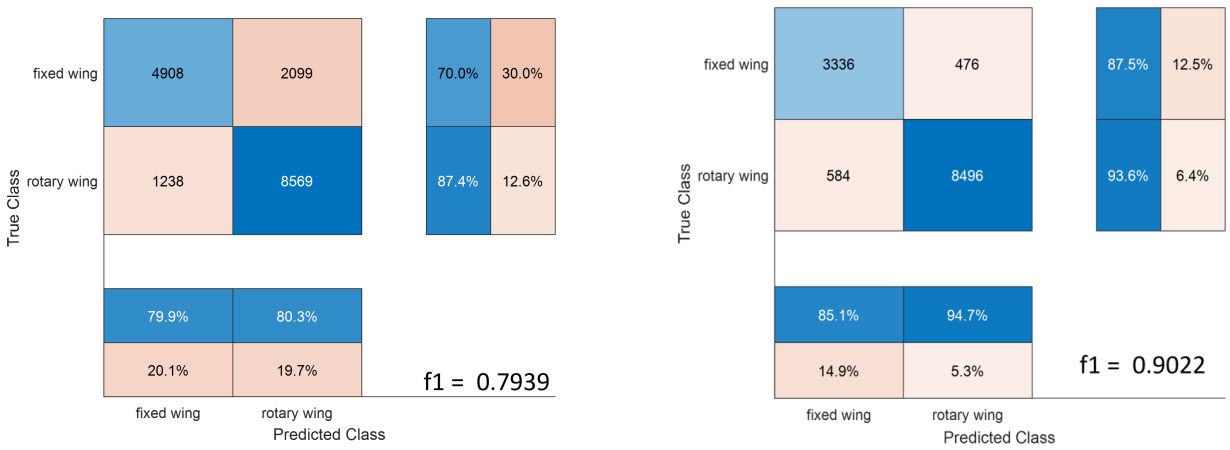
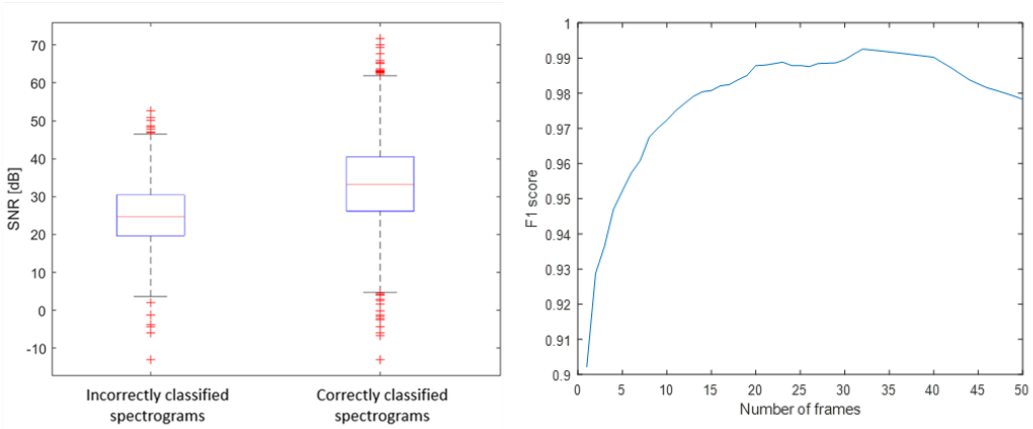


Figure 6.7: Confusion matrices displaying the result of drone group classification a) Tested on the low signal to noise ratio (SNR) data set b) Trained and tested on the full data set.

Finally, files were selected randomly for training and testing irrespective of their SNR. In Chapter 5, it was demonstrated that allowing lower SNR data into the training process improved performance on lower SNR test data. Fig. 6.7b shows the confusion matrix for this case, where training and testing is performed on a per-frame basis. Per-frame, the F1 score is 0.90, while per-file, it is around 0.97. These F1 scores are higher than the lower SNR case and lower than the baseline SNR case,

as expected, demonstrating that level of representation in the training data has a significant impact upon classification performance. Training and testing at high SNR increases the degree of separation between the classes, increasing classification performance. Training at high SNR but testing at low SNR reduces the degree of similarity between the training and test data, therefore training and testing at a variety of SNRs seems to be the intermediate output.

Fig. 6.8a shows the SNR of the test samples that were correctly and incorrectly classified, where the median SNR of the correctly classified spectrograms is significantly higher than that of the incorrectly classified spectrograms. 75% of the correctly classified targets have an SNR in the region of 26-40 dB, with outliers, marked as "+", that range from -15 dB to 70 dB. Whilst 75% of the incorrectly classified spectrograms have an SNR in the region of 20-30 dB, there are still targets of very high SNR (up to 55 dB) which are incorrectly classified, demonstrating that whilst SNR is a factor which affects performance, high SNR alone is not enough to achieve accurate classification.



(a) (b)

Figure 6.8: Drone group classification tested on the full test set a) signal to noise ratio (SNR) of correctly and incorrectly classified targets b) F1 score as a function of number of frames for drone group classification. Note that the peak corresponds to the mean track length of this data set.

Classifying on a per-file basis improves the F1 score by approximately 8%. Of course, rather than integrating the score over all frames within a file, the number of frames could be fixed. In order to understand the time needed to confidently declare a classification output, the F1 score is investigated as a function of the number of frames over which the score is integrated. As shown in Fig. 6.8b, choosing 0.95 as a reliable F1 score requires integration over approximately 5 frames, whereas increasing the F1 score to 0.97 would require almost double the frames (therefore double the observation time). It is observed that as the number of frames, N, increases, so does the F1 score, however there is an upper limit after which the F1 score remains almost stable before gradually decreasing when N exceeds the mean track length (in this case, 35 frames).

To summarise, Table C.1 shows the F1 scores obtained for drone group classification when classifying on a single frame or a single file in the three SNR regimes considered.

6.4 Drone Subgroup Classification

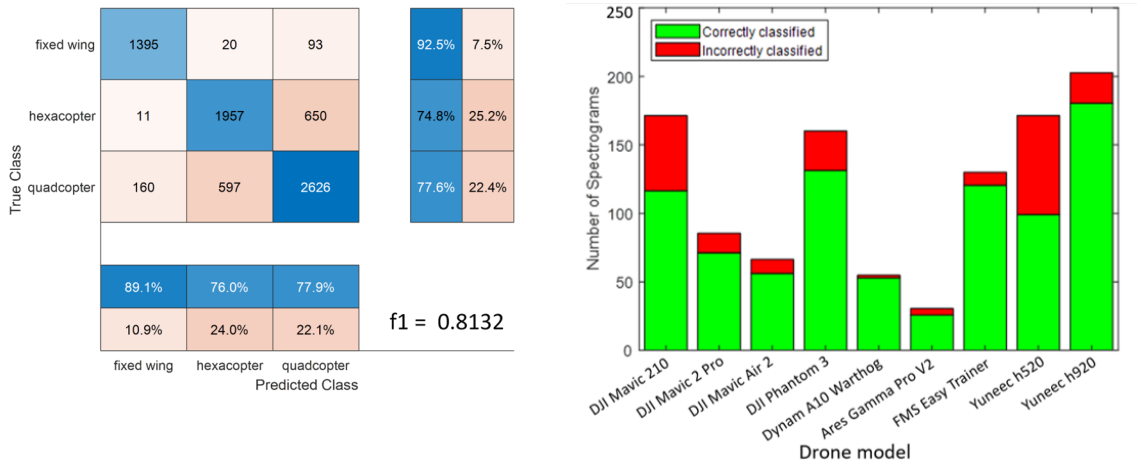
In this section, the rotary wing drones are further divided into two classes. First, the rotary wing drones are split based on the number of rotors, into a quadcopter class and a hexacopter class. In the second part of the section, the rotary wing drones are instead categorised by size.

6.4.1 Classification Based on Number of Rotors

Baseline case

Results are first reported for the baseline case, where training and testing is performed using data with SNR>30 dB. Fig. 6.9a-b show the confusion matrix and the

number of correct classifications per drone model, after seven repetitions. Fig. 6.10 tests the "Unknown" class.



(a) (b)

Figure 6.9: Drone subgroup classification, tested on the baseline data set a) Confusion matrix, b) Classification outcomes for each model of drone.

The F1 score is 0.81, which is 13% lower than that the F1 score achieved in 2-class problem (Fig. 6.4a) due to the increased complexity of the problem. The fixed wing drone recall remains very high (92.5%), and the main source of confusion arises when discriminating between hexacopters and quadcopters, with recalls of 74.8% and 77.6% respectively. This is reflected in Fig. 6.10, where high confidence is demonstrated in the fixed wing class, but the high true positive rates in the other two classes come at the price of an increased false positive rate. Fig. 6.9b shows that the source of error is largely due to the DJI Matrice 210 (a quadcopter) often being mistaken for a hexacopter, and the Yuneec H520 (a hexacopter) being mistaken for a quadcopter. This is not entirely unreasonable - the physical size of the DJI Matrice 210 (and the length of its blades) is uncommon for a quadcopter, while the dimensions of the Yuneec H520 are smaller than those of the other hexacopter for which data has been obtained.

If testing is conducted on entire files, rather than a single frame, the F1 score

is increased by approximately 6%. The increase in class recall is most notable for quadcopters, rising from 78% to 90%.

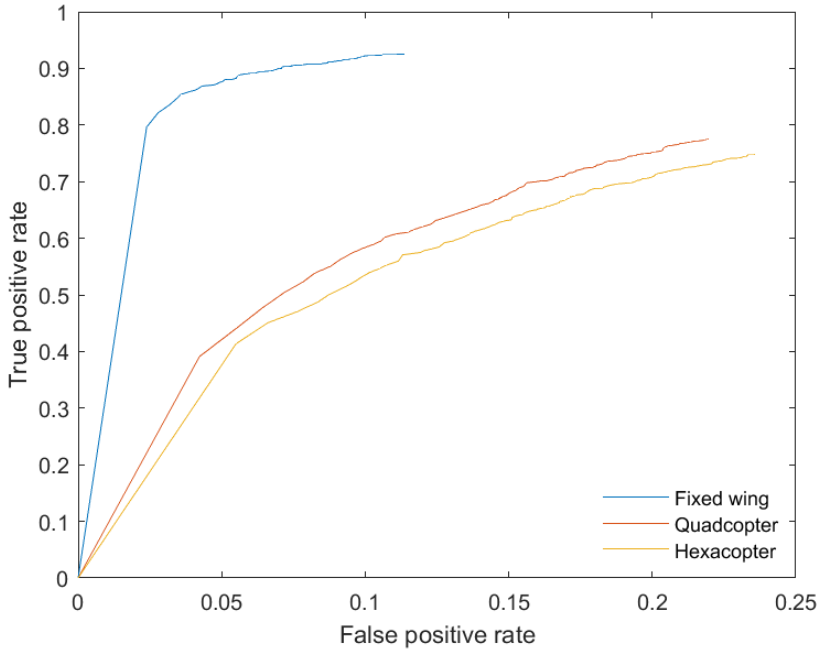


Figure 6.10: Drone subgroup classification receiver operating characteristic (ROC) curves, tested on the baseline data set.

Low SNR and All Data Cases

Fig. 6.11a shows that when the classifier is trained on higher SNR data (>30 dB) and tested on lower SNR data (<30 dB) the classifier performance decreases, similarly to the case of drone group classification. The F1 score decreases by 23%, from 0.81 to 0.62 when classification is performed on a per-frame basis, compared to the baseline case. When classifying over an entire file much of the performance can be regained, achieving an F1 score of 0.76, however, this is still 6% less than what could be obtained using a single frame and higher SNR data.

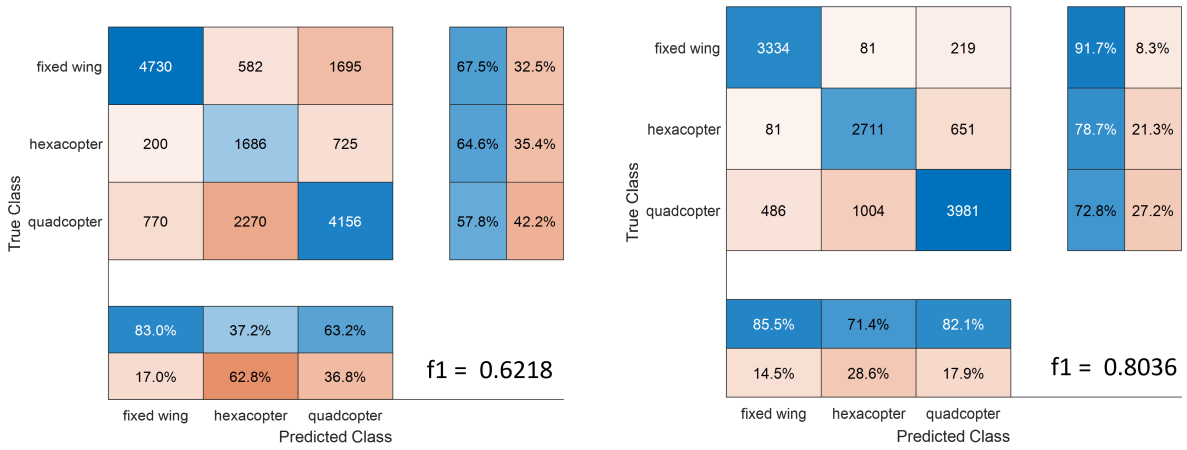


Figure 6.11: Drone rotor subgroup classification results a) Tested on the low signal to noise ratio (SNR) test data set b) Tested on the full test set.

The results obtained when training and testing samples are chosen irrespective of their SNR are shown in Fig. 6.11b. On a per-frame basis, the F1 scores obtained are similar to those in the baseline case. The general trend is also very similar- the fixed wing drone class has the highest recall, and hexacopters are mostly confused with quadcopters (and vice versa). On a per-file basis, the F1 score increases by 15% to 0.93. The performance of the drone subgroup classifier is summarised fully in Table C.2.

6.4.2 Classification Based on Size

The shape and spacing of HERM lines are influenced by drone mass, as shown in [136][137] where micro-Doppler features were used to classify loaded vs unloaded drones. Therefore, an alternative approach to subgroup classification is presented, where size was used to classify the drones instead of the number of rotors. The classes were re-assigned as “fixed wing” (Ares Gamma Pro V2, FMS Easy Trainer, Dynam A10 Warthog), “small rotary wing” (DJI Mavic 2 Pro, DJI Mavic Air 2,

DJI Phantom 3, Yuneec H520) and "large rotary wing" (DJI M210, Yuneec H920) drones. Viewing classes in this way may also make more sense from an operational perspective for defence applications. Knowing the size of a drone (and hence its capacity) could provide better actionable intelligence than knowing the number of rotors that the drone possesses.

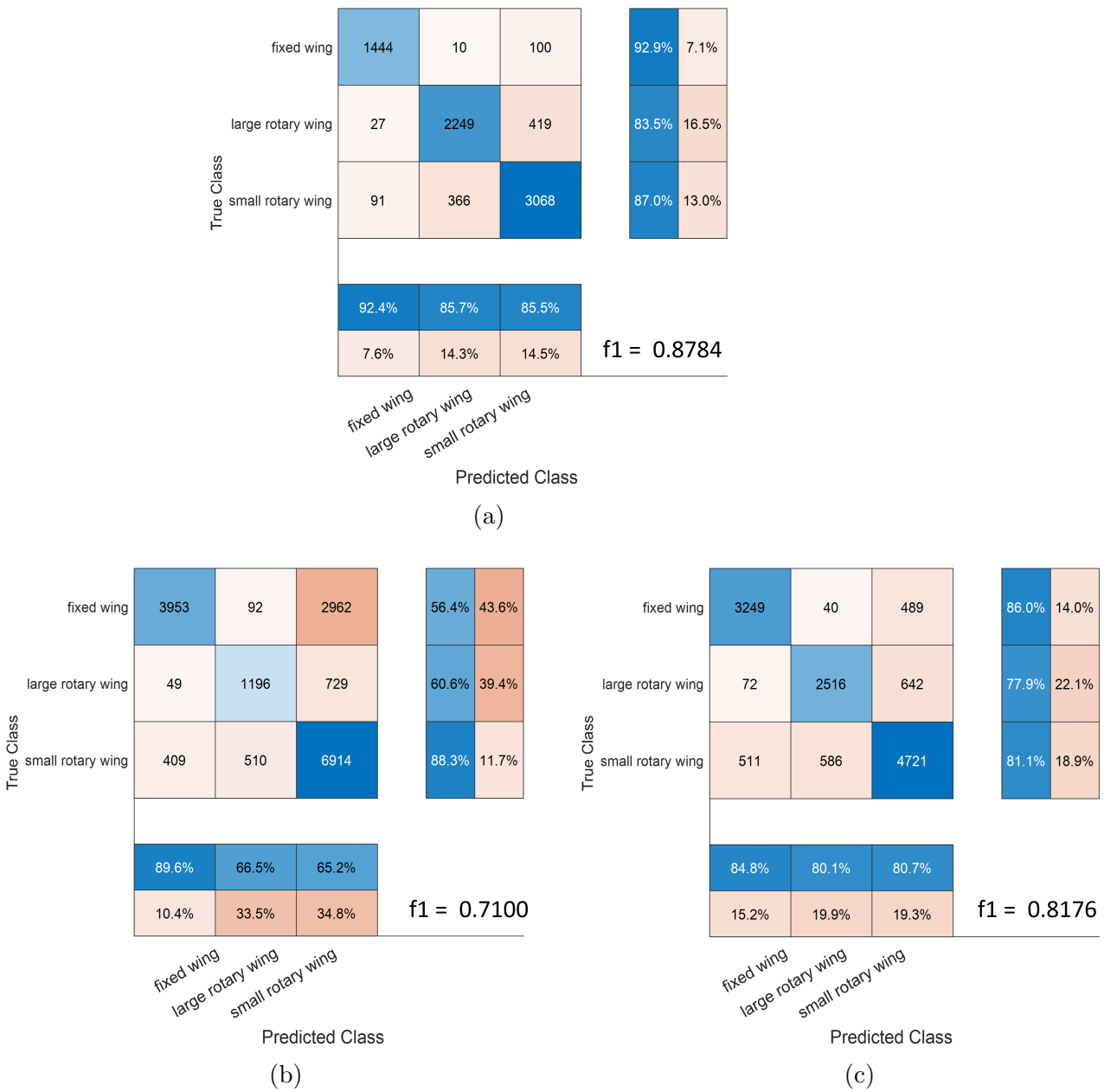
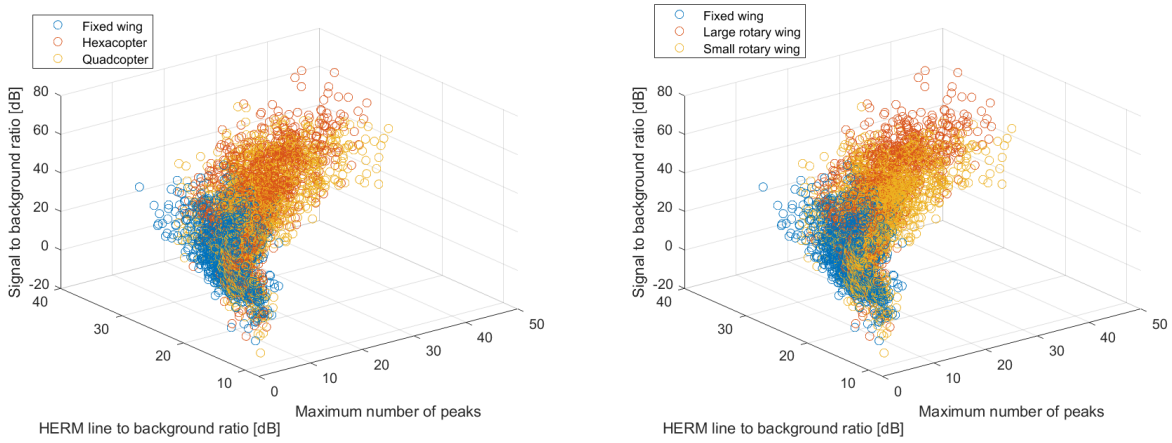


Figure 6.12: Drone size subgroup classification results a) Baseline case b) Low signal to noise ratio (SNR) case c) All data case.

The confusion matrix for per-frame classification is shown in Fig. 6.12a for the baseline case. Despite using exactly the same classifier, the F1 score can be improved by up to 8% simply by assigning the classes differently.

Fig. 6.13 shows the signal to background ratio of the drone body and strongest HERM lines, and the maximum number of Doppler peaks across all spectra comprising the spectrogram. It is demonstrated that changing the class groupings enhances the separability of the subgroups in 3-D feature space, facilitating classification. In particular, the measured SNR and HERM line to background ratio of the large rotary wing class is typically significantly larger than that of the small rotary wing class, due to the larger RCS of the drone body and propellers. The reliability of the class labels can be inferred from Fig. 6.14, where comparison with Fig. 6.10 indicates that grouping data in this way enhances performance by improving across-class feature separation.



(a) (b)

Figure 6.13: Feature plot showing the drone body signal to background ratio, strongest helicopter rotor modulation (HERM) line power to background ratio, and the maximum number of peaks in each spectrogram a) Rotor-based subgroups, b) Size-based subgroups.

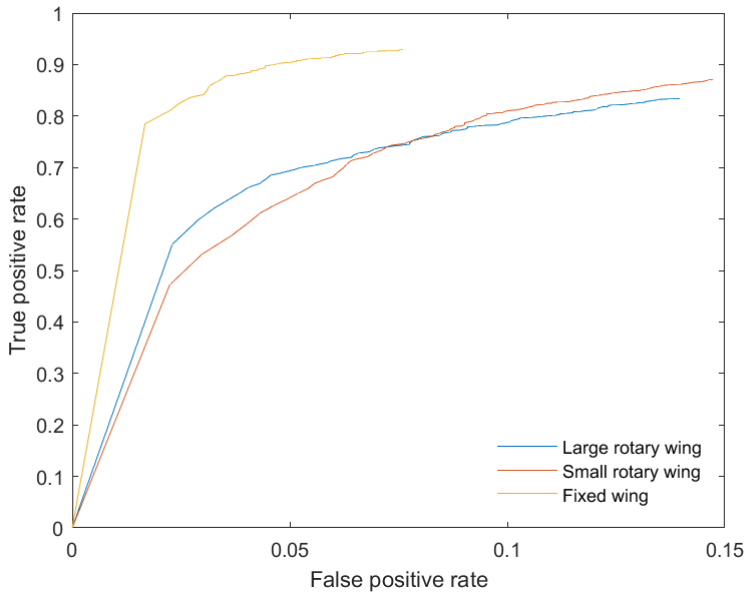


Figure 6.14: Drone size subgroup classification receiver operating characteristic (ROC) curves, tested on the baseline data set.

The performance of the size-based classifier remains high when considering the low SNR (Fig. 6.12b) and all data cases (Fig. 6.12c), compared to grouping the rotary wings based on the number of rotors. When testing on the low SNR subset, performance drops by 19%. The results are summarised in full in Table C.3.

6.5 Drone Model Classification

Lastly, AlexNet is trained to classify drone model from radar spectrograms. Training and testing is again conducted under the "baseline", "low SNR", and "all data" conditions, using a single frame or a single file.

In this work, multiple hierarchies are investigated to determine whether progressively increasing the amount of information about a target results in improved classification performance. Firstly, a single-stage nine class classifier is trained. Sub-

sequently, two 2-stage classifiers are created – in the first model, drone group classification is performed first, followed by drone model classification. In the second 2-stage model, drone subgroup is identified, before carrying out drone model classification. Finally, a 3-stage classifier is created, where a drone is first assigned to a group, then a subgroup, and finally to a drone model.

6.5.1 Single-stage Classifier

Baseline Case

The baseline results for per-frame classification are shown in Fig. 6.15. Considering that for a 9-class problem an accuracy of 0.11 could be achieved by randomly guessing the class, an F1 score of 0.55 is acceptable considering the input is a single 1 second spectrogram. It is likely that the performance may still be too low for practical use, although this will of course depend on the specific application. Classifying on a per-file basis improves F1 scores by approximately 23%, giving an overall F1 score of nearly 0.68. Fig. 6.16 tests the "Unknown" class on a per-frame basis. In addition to the mean ROC curve across all classes, ROC curves were also computed for each class to understand the confidence in declaring each of them.

Within the confusion matrix, the variability in class recall is noteworthy. For example, the Yuneec H920 has a recall of 85.7%, whereas the DJI Mavic 2 Pro has a class recall of 31.2%. A few caveats need to be noted when attempting to make a fair comparison between individual accuracies. For example, the Yuneec H920 has the highest accuracy but also contains the larger number of test spectrograms, outweighing the Ares Gamma Pro V2 class (recall of 34.8%) by factor of 5.7. Although the loss function is weighted to account for this, it may not be enough to effectively

handle such an imbalance between classes as the lack of representation may hinder the approximation of the target function.

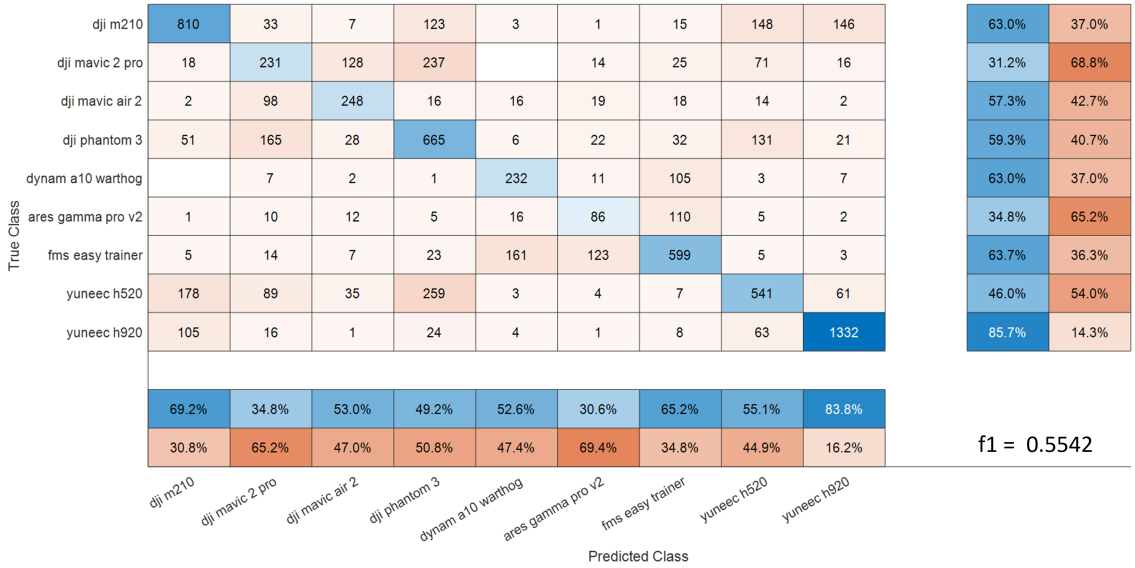


Figure 6.15: Single-stage drone model classification confusion matrix, tested on the baseline data set.

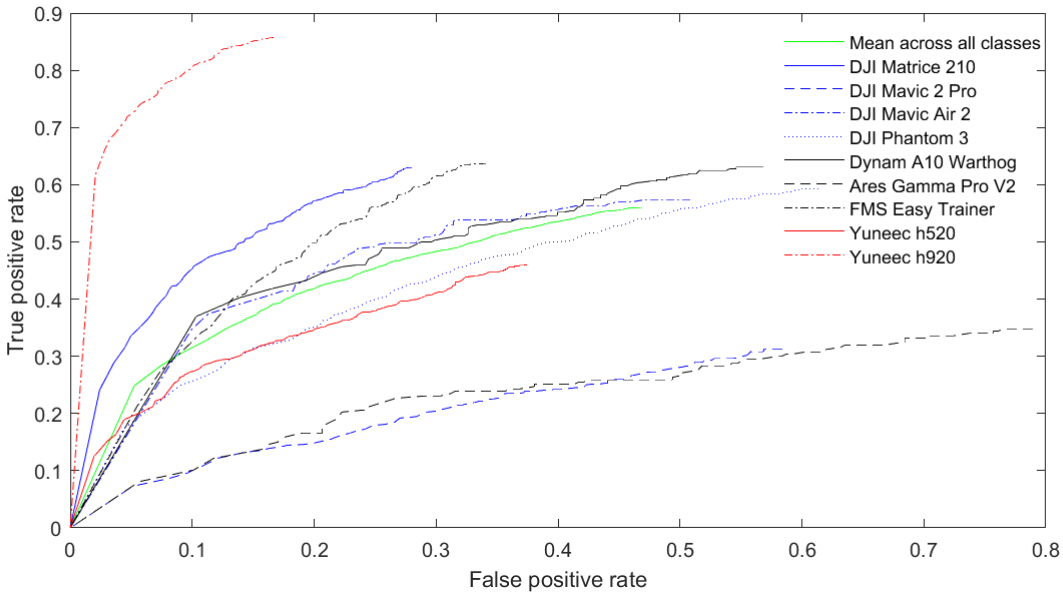


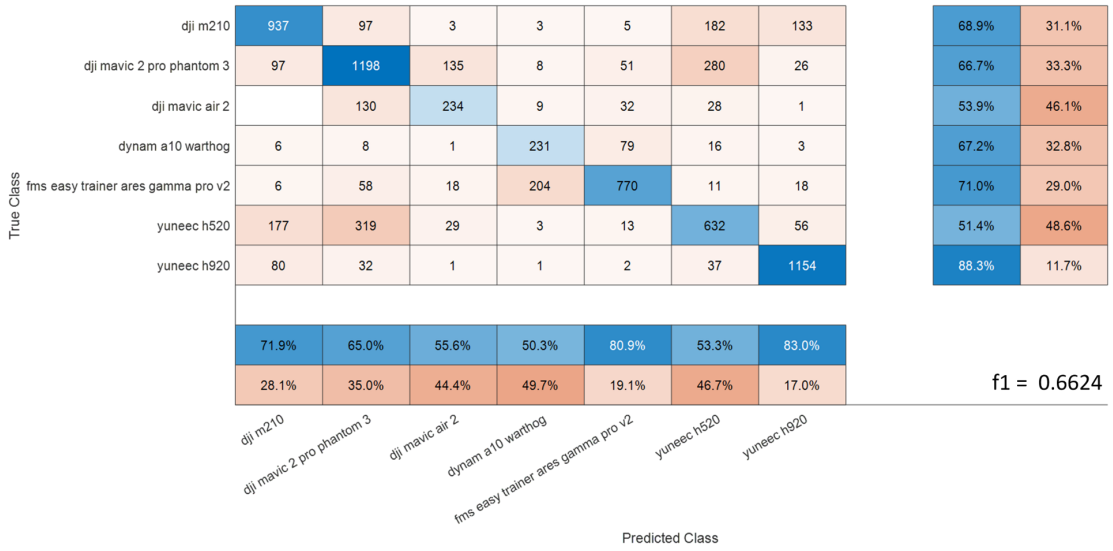
Figure 6.16: Drone model classification receiver operating characteristic (ROC) curves, tested on the baseline data set.

However, there does not seem to be a direct correlation between the class sample

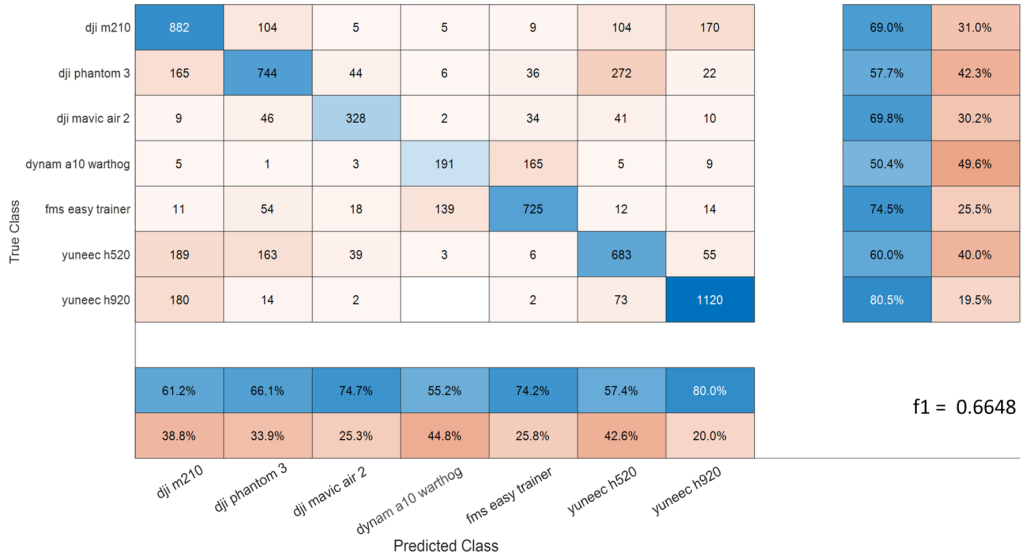
size and the classification accuracy. For example, the DJI Mavic 2 Pro class has a recall of 31.2% but a sample size that is twice larger than the Dynam A10 Warthog class, with a recall of 63.0%.

Broadly there appears to be a trend describing which classes are most commonly confused. Quadcopters tend to be more confused with other quadcopters and to a lesser extent with hexacopters. These drones are made by the same manufacturer, therefore it could be that they have common components leading to similarities in their signatures. Similarly, fixed wing drones tend to be more commonly confused with other fixed-wing drones than with quadcopters or hexacopters. On the other hand, the hexacopters more commonly confused with quadcopters than with each other. This is likely because the Yuneec H520 is more similar in weight to the DJI Phantom 3, which it is often mistaken for, whilst the Yuneec H920 is a similar weight to the DJI Matrice 210. This supports the findings from Section 6.4, where it was demonstrated that grouping drones based on their weight led to improved class separation compared to grouping them based on the number of rotors, implying that the very heavy rotary wing drones had more features in common.

The largest source of error appears to be in distinguishing between individual drone models. In particular, between the DJI Mavic 2 Pro and the DJI Phantom 3, which are very similar in terms of size and weight (Table 6.1), with the same number of rotors. Another significant source of error is confusion between the Ares Gamma Pro V2 and the FMS Easy Trainer, which are both fixed wing models of similar weights. These weaker classes were combined into two equivalent classes: a "Mavic 2 Pro/Phantom 3" class and an "FMS Easy Trainer/Ares Gamma Pro V2" class (essentially reducing the number of classes to seven). The results of repeating the training and testing process using these classes are given in Fig. 6.17a.



(a)



(b)

Figure 6.17: Single-stage drone model classification results where the DJI Mavic Pro and the Ares Gamma Pro V2 classes have been a) Combined with the classes they are most commonly mistaken for, b) Removed.

Operating in this way improves the F1 scores by about 20% and 17% for per-frame and per-file classification, respectively. It should also be noted that the combined classes also contain substantially more data than they did individually. If the weaker classes (i.e. DJI Mavic Pro and the Ares Gamma Pro V2) are removed

entirely, the F1 score for the resulting 7-class problem increases (0.66 and 0.83 for per-frame and per-file classification, respectively) as shown in Fig. 6.17b. Note that in the per-file classification, the class that performs the worst - the Dynam A10 Warthog (recall of 50.4%) - is also the class that contains the least number of files compared to the other classes.

Low SNR and All Data Cases

Fig. 6.18 shows the result of testing the classifier trained on the baseline data set using lower SNR spectrograms. As expected, F1 scores in this case are drastically reduced, with a maximum recall of 56.7% for the Yuneec H920 and a minimum recall of only 12.8% for the Ares Gamma Pro V2. Classifying on a per-file basis improves the F1 score from 0.39 to 0.56.

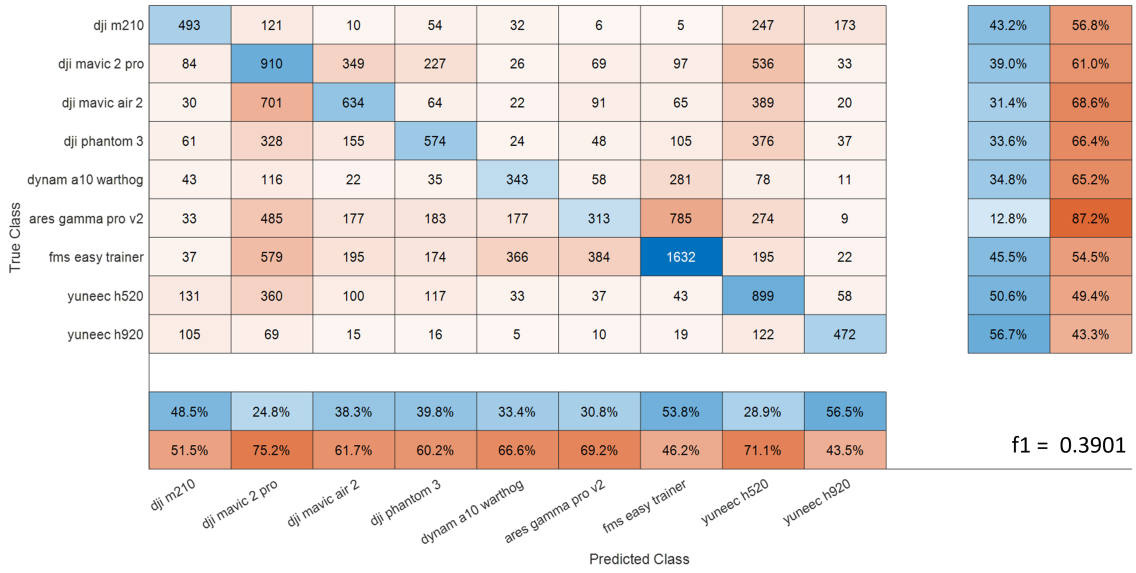
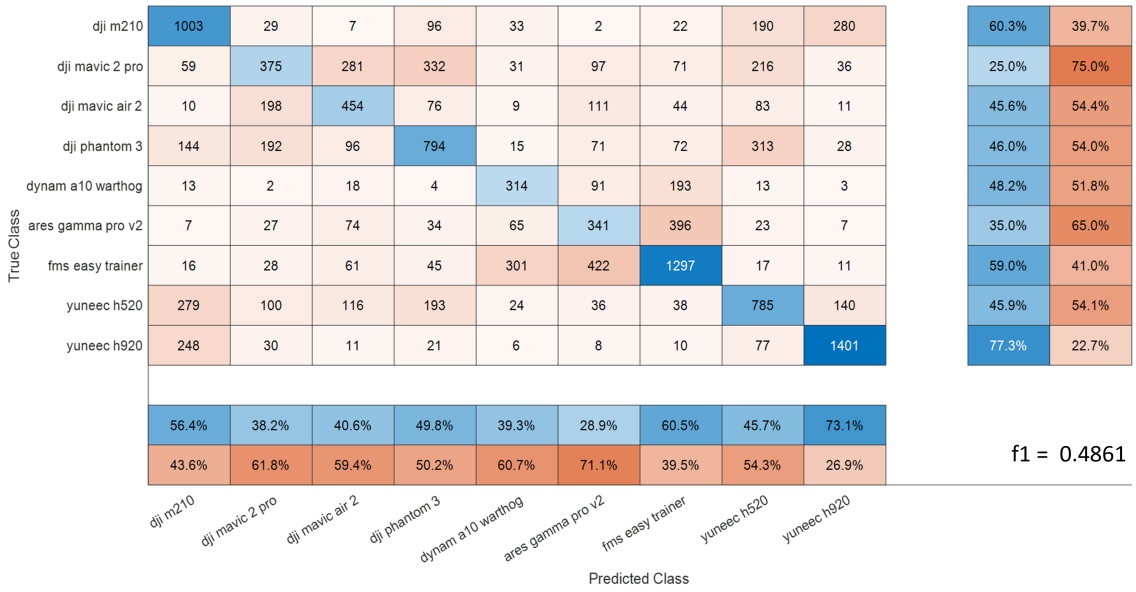


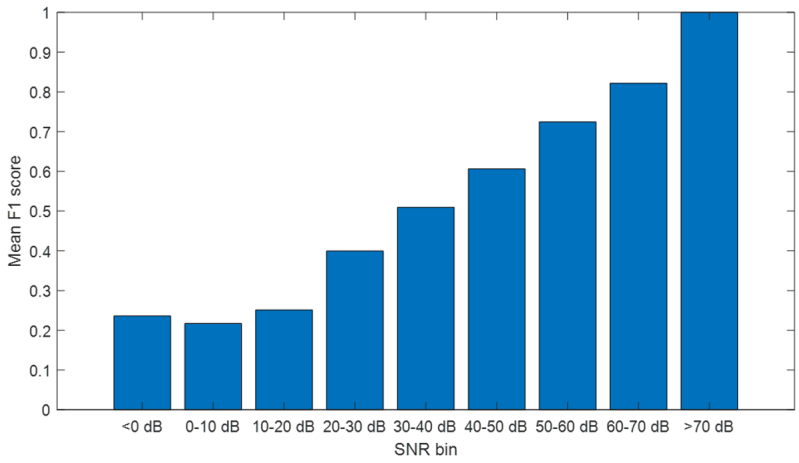
Figure 6.18: Single-stage drone model classification results, tested on the low signal to noise ratio (SNR) data set.

Fig. 6.19a shows the results of drone model classification where training and testing is performed across the whole data set. F1 scores of 0.49 and 0.67 are achieved

on a per-frame and per-file basis, respectively, with the overall performance sitting between the baseline and low SNR cases. This is in line with all equivalent results in previous sections.



(a)



(b)

Figure 6.19: Single-stage drone model classification results, tested on the full test data set a) Confusion matrix, b) Mean F1 score vs target signal to noise ratio (SNR).

In Fig. 6.19b, the mean F1 score is shown as a function of SNR bin. The performance is negatively affected by the cases where the SNR is impractically low even for detection, i.e. <10 dB, where F1 scores are around 0.20. There is a notable jump

in performance between the 10-20 dB and 20-30 dB SNR bins. This is likely due to the fact that an SNR in the 20-30 dB range is sufficient for HERM lines to be visible above the noise floor, leading to an increase in performance. The relationship between SNR and F1 score is evident: as the SNR of the test data increases, the F1 score when training with a mix of SNRs can substantially increase. However, this figure should be interpreted with caution as the target SNR bins are not equally sampled. In fact, the > 70 dB bin is occupied by a single drone model.

Fig. 6.20 shows the F1 score as a function of the number of frames used for classification. The maximum F1 score that can be achieved is 0.73. To achieve this however, up to 40 frames are needed, unlike with the two and three class problems for which high performance could be achieved using 10 frames. For the 9-class problem, performance improves significantly as the number of frames increases compared to the two and three class problems.

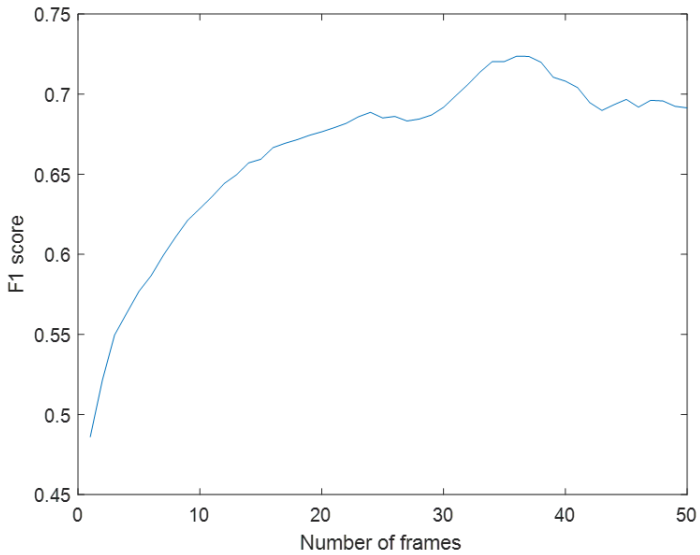
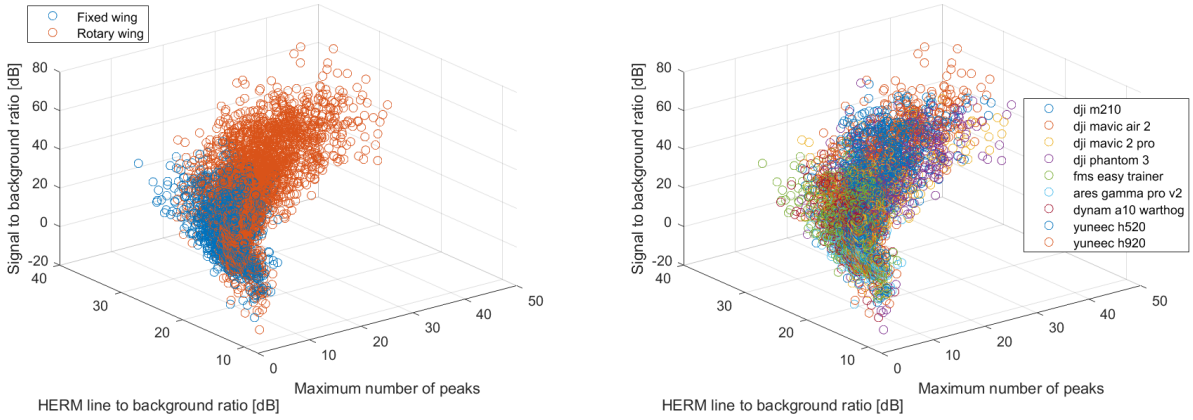


Figure 6.20: Drone model classification F1 score as a function of the number of frames, tested on the full test data set. Note that the peak corresponds to the mean track length of this data set.

Fig. 6.21 shows the position of all of the drone data frames in feature space,

showing the signal to background ratio of the drone body and strongest HERM lines, and the maximum number of Doppler peaks across all spectra comprising the spectrogram. The two and nine class problems are shown side-by-side for comparison. As the number of classes increases, it becomes considerably more difficult to separate them, placing an upper limit on classification performance. Of course, it is important to note that the feature plots only show three possible features, whereas the CNN is capable of extracting large numbers of abstract features, and can draw non-linear boundaries between clusters.



(a) (b)

Figure 6.21: A feature plot showing the body signal to background ratio, strongest helicopter rotor modulation (HERM) line power to background ratio, and the maximum number of peaks in each spectrogram a) Drone groups, b) Drone models.

6.5.2 Multi-stage Classification

In this section, rather than using a single 9-class CNN, drone model classification is performed across multiple stages. Generally, the amount of data required for robust classification increases as the number of classes increases. By splitting each stage into a reduced number of classes, an enlarged data set within each class exists at every classification stage. For example, consider a 2-stage classifier where drones are classified by group in the first stage, and model in the second stage. The second

stage network tasked with classifying the rotary wing drone models will have six possible classes, whilst the network tasked with fixed wing model classification will have three possible classes. On one hand, an increased classification accuracy in earlier stages could reduce misclassifications across the final nine classes, however, targets misclassified in the earlier stages of the classifier will not propagate through the classification stages correctly. The optimal classification hierarchy therefore depends on which of the effects dominate.

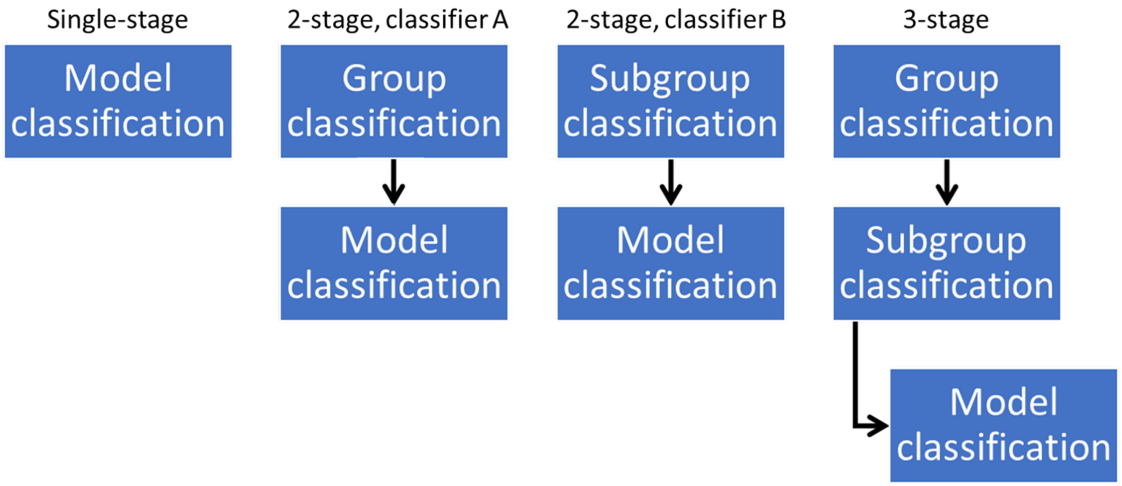
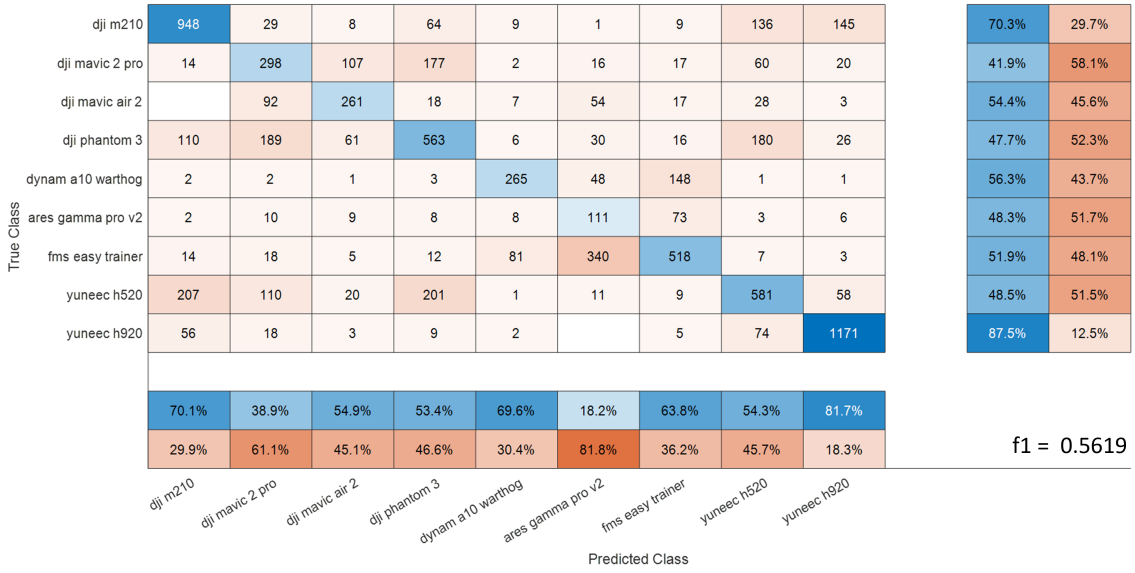


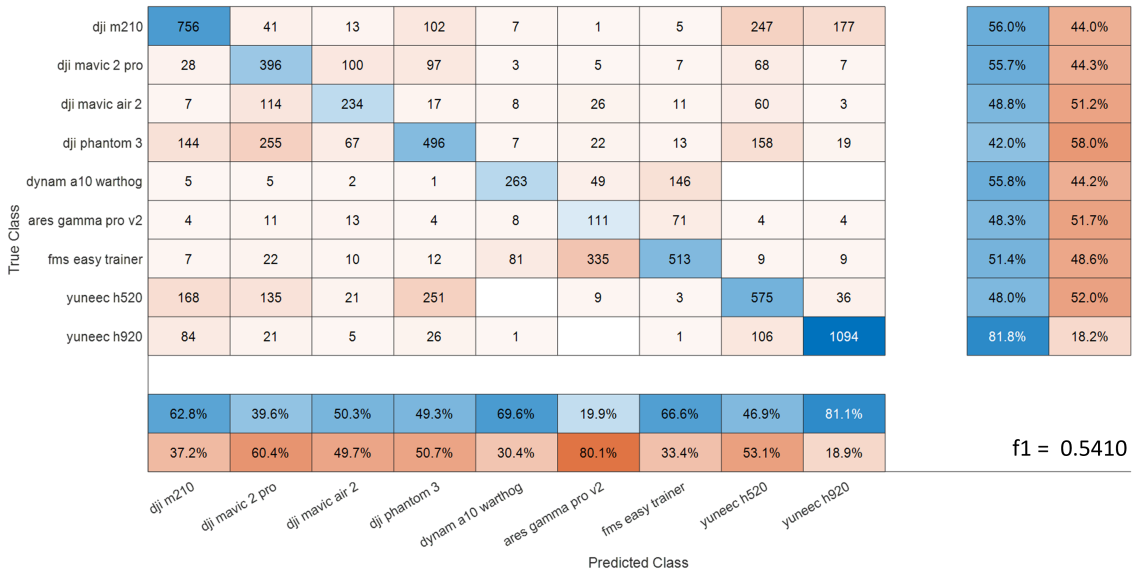
Figure 6.22: The classification hierarchies tested for drone model classification.

Fig. 6.22 shows the multi-stage architectures that were tested for drone model classification. These can be summarised as the single-stage classifier, two 2-stage classifiers (A and B), and a 3-stage classifier. In classifier A, the drones are classified by group in the first stage, whereas in classifier B, the drones are first classified by subgroup. Each individual classifier within a hierarchy is a CNN with AlexNet architecture.

2-Stage Classification



(a)



(b)

Figure 6.23: 2-stage drone model classification confusion matrices: a) Group classification followed by model classification, b) Subgroup classification followed by model classification.

The overall confusion matrix for Classifier A after seven repetitions is shown in Fig. 6.23a. The degree of improvement over the single-stage classifier is not signifi-

cant. It appears that in the case of the single-stage classifier, the percentage of fixed wing drones mislabelled as rotary wing and vice versa is not significant enough for the multi-stage classifier to provide substantial improvement.

The overall confusion matrix for Classifier B is shown in Fig. 6.23b. Compared to the equivalent single-stage classifier results of Fig. 6.15, no significant difference in overall F1 scores can be observed. The accuracy in the previously weaker classes, the DJI Mavic 2 Pro and the Ares Gamma Pro V2 classification, is vastly improved, but at the expense of poorer performance across the other classes due to errors from the first stage propagating through the model.

3-Stage Classifier

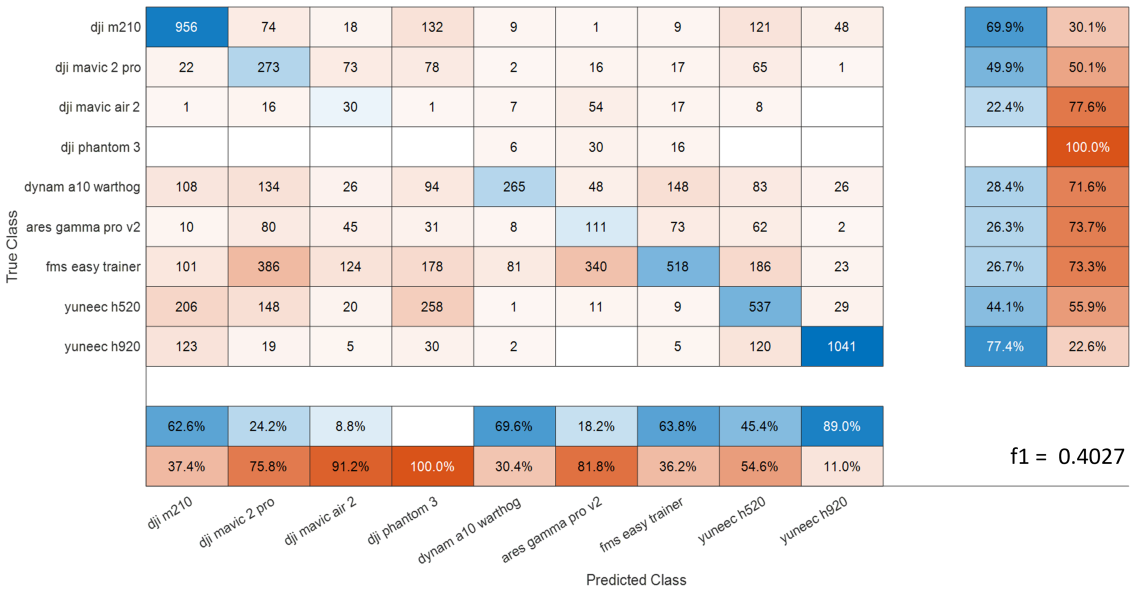


Figure 6.24: 3-stage drone model classification confusion matrix.

Fig. 6.24 shows the results from the 3-stage classifier. The F1 scores achieved using this classification method are considerably poorer than those obtained using alternative approaches. Errors due to samples that are misclassified at the first stage

propagate onward through the classifier stages, resulting in poor performance.

Overall, the multi-stage classification approaches tested here do not appear to have significant advantages over the single-stage classifier. However, if the number of classes were increased beyond nine, this may no longer be the case. A summary of results obtained for the 9-class problem is provided in Table C.4. The maximum F1 score is 0.72, when a two-stage classifier is used, performing group classification before model level classification.

6.6 Effect of Processing Interval on Drone Classification Performance

Good baseline performances for drone classification have been established, using CNNs. However, the processing length was fixed at 512 pulses, whereas in Fig. 6.1, it was demonstrated that changing the processing length has significant impact on the appearance of the HERM lines, with implications for classification performance. In [148], the authors highlighted the importance of integration time and the implications on drone classification performance, demonstrating that integration time had more of an impact on performance than the classifier used.

In this section, the integration time is varied and the resulting effect on SNR and HERM line signal strength are investigated. It is hypothesised that improving the Doppler resolution and SNR will lead to enhanced separability of the classes in feature space, improving classification performance. To investigate this hypothesis, handcrafted features are extracted from the spectrograms and a decision tree classifier is trained to perform the drone classification tasks from Section 6.3 to Section 6.5, and the impact of processing interval upon classification is examined.

Finally, methods of exploiting the gains arising from improved Doppler resolution and SNR are explored.

6.6.1 Feature Extraction

Decision trees are used rather than CNNs as they are much faster to train and showed promising performance in Chapter 5. A fast training time is needed as multiple classifiers are trained using different processing lengths, and the experiment is repeated many times to account for variance due to the stochastic nature of training.

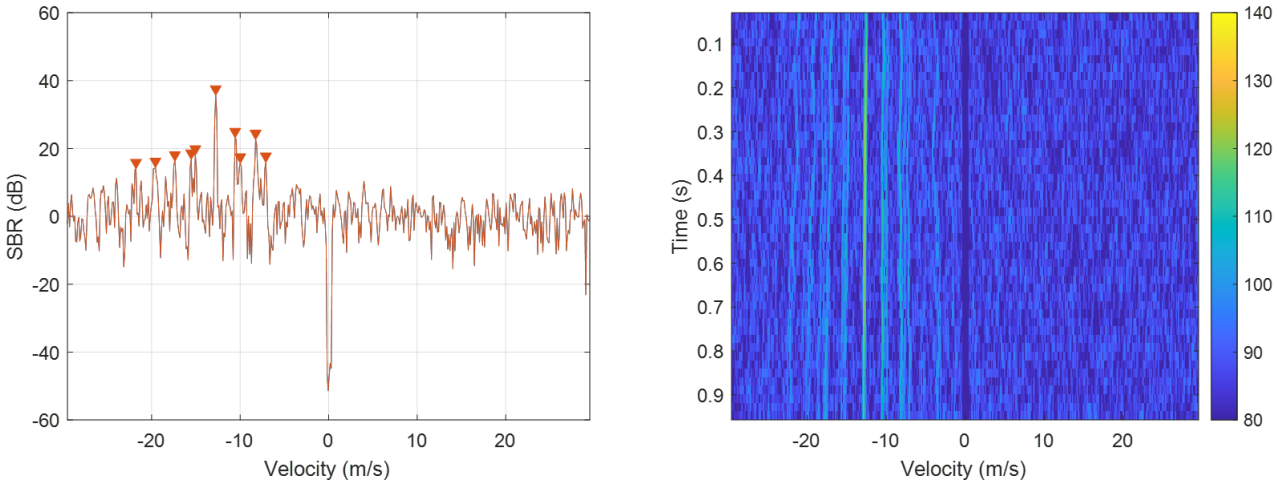
Spectrograms were produced for each frame of data, where the STFT length was varied from 128 to 4096 pulses, and the amount of overlap was modified to keep the time resolution constant. Thirteen features were then extracted from the spectrograms in the baseline data set. These features are summarised in Table 6.4.

Table 6.4: A summary of the handcrafted features extracted from the drone spectrograms.

Feature name	Description
Maximum number of peaks	The maximum number of peaks in a spectra, within a 1-second frame
Minimum number of peaks	The minimum number of peaks in a spectra, within a 1-second frame
Peak range	Maximum number of peaks – minimum number of peaks
Mean number of peaks	Number of peaks in a spectra, averaged over a 1-second frame
Total number of peaks	Number of peaks in a spectra, summed over a 1-second frame
Mean HERM line SNR	Ratio of the power of the first HERM line to the average power in the outer Doppler bins, averaged over the 1-second frame
Mean body to HERM line ratio	Ratio of the power of the body return to the power of the first HERM line, averaged over the 1-second frame
Mean peak spacing	Mean spacing between peaks within a spectra, averaged across a 1-second frame
Mean body velocity	Velocity of the target body return, averaged across a 1-second frame
Mean number of peaks to the left of the body return	Number of peaks to the left of the body return in a spectra, averaged over a 1-second frame
Mean number of peaks to the right of the body return	Number of peaks to the right of the body return in a spectra, averaged over a 1-second frame
Sideband symmetry	Abs(Mean number of peaks to the left of the body return – mean number of peaks to the right of the body return)
Mean body SNR	Ratio of the power of the body return to the average power in the outer Doppler bins, averaged over the 1-second frame

The features can be split into two categories: power-based features and Doppler-

based features. The power-based features were chosen as the drone model RCS will be set by the size, shape, and material of the body and rotors, with implications on the measured signal strengths. The Doppler-based features aim to infer information about the target's trajectory, and the symmetry, shape, and spacing of the HERM lines. The uniformity of the HERM lines across the spectrogram was noted as a potential class discriminator, as the shape of the drone and number and position of the rotors led to shielding effects by the fuselage or other rotors. Therefore, the number of peaks in each spectra is extracted, and the mean, maximum, minimum, and sum are taken across the entire 1-second frame. Fig. 6.25 shows the number of peaks extracted from a spectra, alongside the full 1-second spectrogram.



(a) (b)

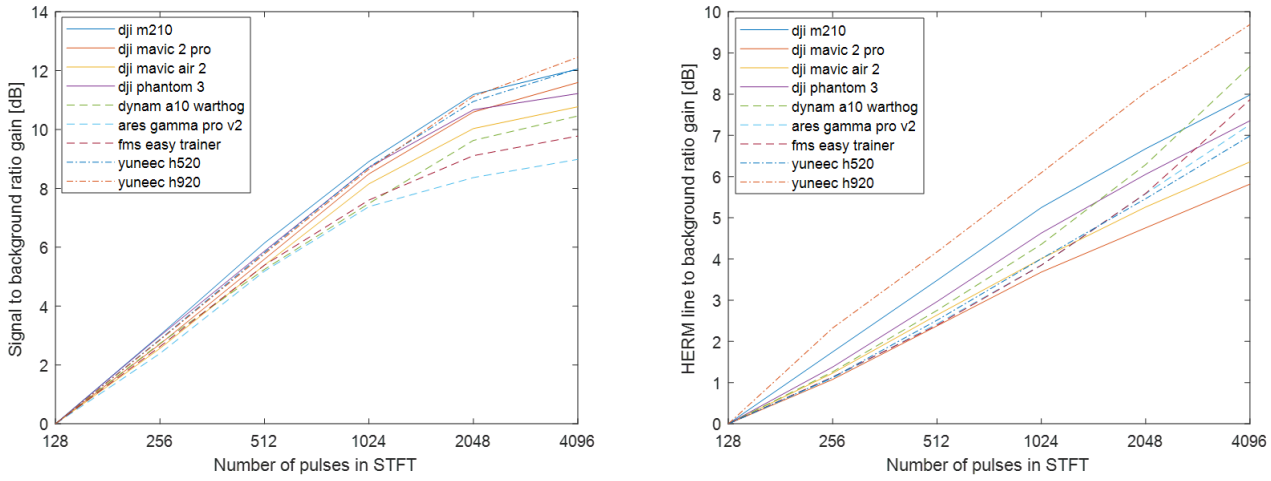
Figure 6.25: a) Peaks extracted from a single spectra, after taking the signal to background ratio (SBR) b) Corresponding DJI Mavic Air 2 spectrogram.

6.6.2 Feature Analysis

Increasing the integration length can be expected to increase SNR (∼3 dB gain per doubling of integration time for a coherent target), and increase Doppler resolution. However, the nature of drone flight, the effect of wind conditions, and the variation

in RCS due to geometry, will all limit the timescale for which the signature remains coherent. In [62], it is shown that drone signatures are not coherent over a long timescale, and that the problem is exacerbated by flight style and environmental conditions.

Fig. 6.26a shows the gain in signal to background ratio as the integration length is doubled. The SNR gain is broadly linear up to an integration length of 1024 pulses, corresponding to an integration time of 155 ms. Coherency is also a function of target type, with the DJI Matrice 210 and Yuneec H920 benefiting the most from an increased integration time, whilst the Ares Gamma Pro v2 benefits the least. Likewise, Fig. 6.26b shows the gain in the HERM line to background ratio as a function of integration length, where the HERM line power increases slowly with increasing integration length. The gain is most significant for the Yuneec H920, and least significant for the DJI Mavic 2 Pro model.

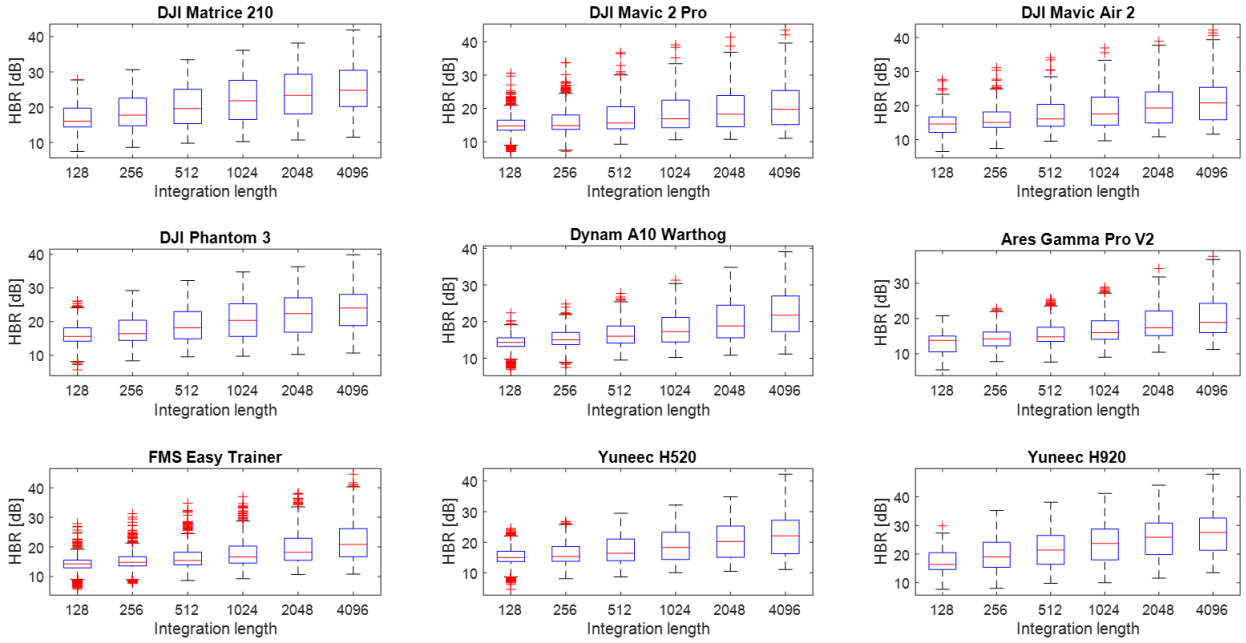


(a) (b)

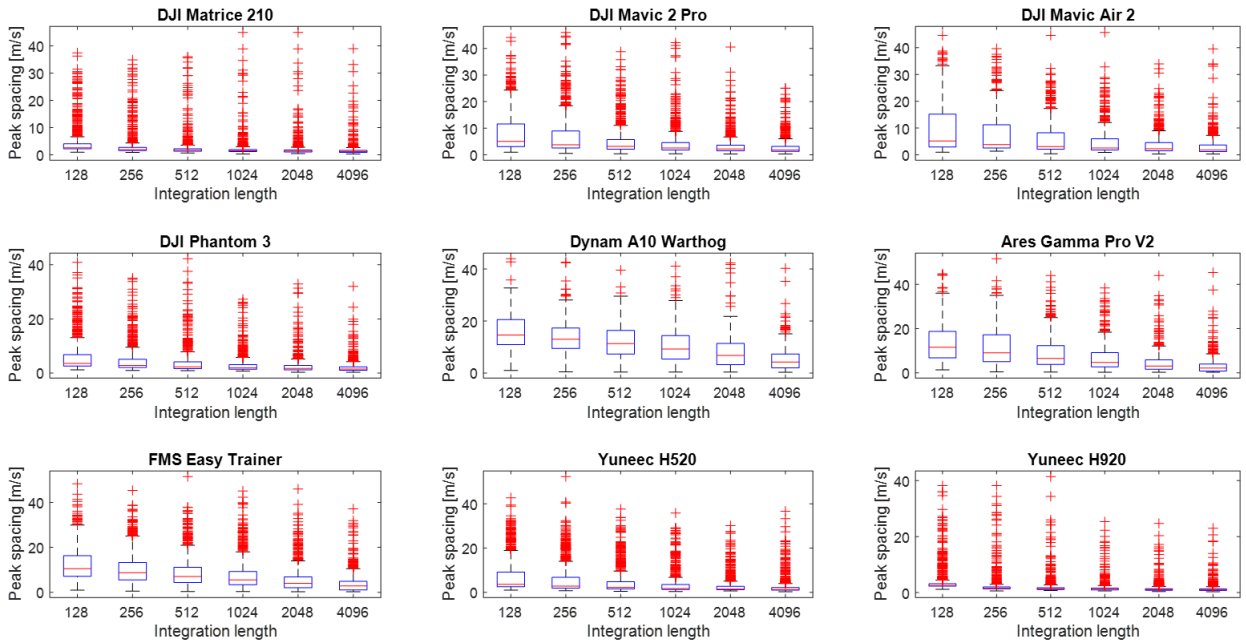
Figure 6.26: a) Gain in signal to background ratio with increasing integration length, b) Gain in helicopter rotor modulation (HERM) line to background ratio with increasing integration length.

As the HERM lines are a key classification feature, one might expect that increasing the processing length will improve drone classification performance, however, this

is not necessarily the case. More important than the signal to background ratio is the separation of the classes in feature space.



(a)



(b)

Figure 6.27: a) Spread in the helicopter rotor modulation (HERM) line to background ratio (HBR), for each drone model as the processing length is increased b) Spread in the mean spacing between Doppler peaks for each drone model as the processing length is increased.

Fig. 6.27a shows the median and range of the HERM line to background ratio as a function of integration length for each model type - note that whilst the median HERM line power increases with increasing processing length, so does the spread. This increases the amount of overlap between classes for this particular feature, which could lead to a reduction in class separability.

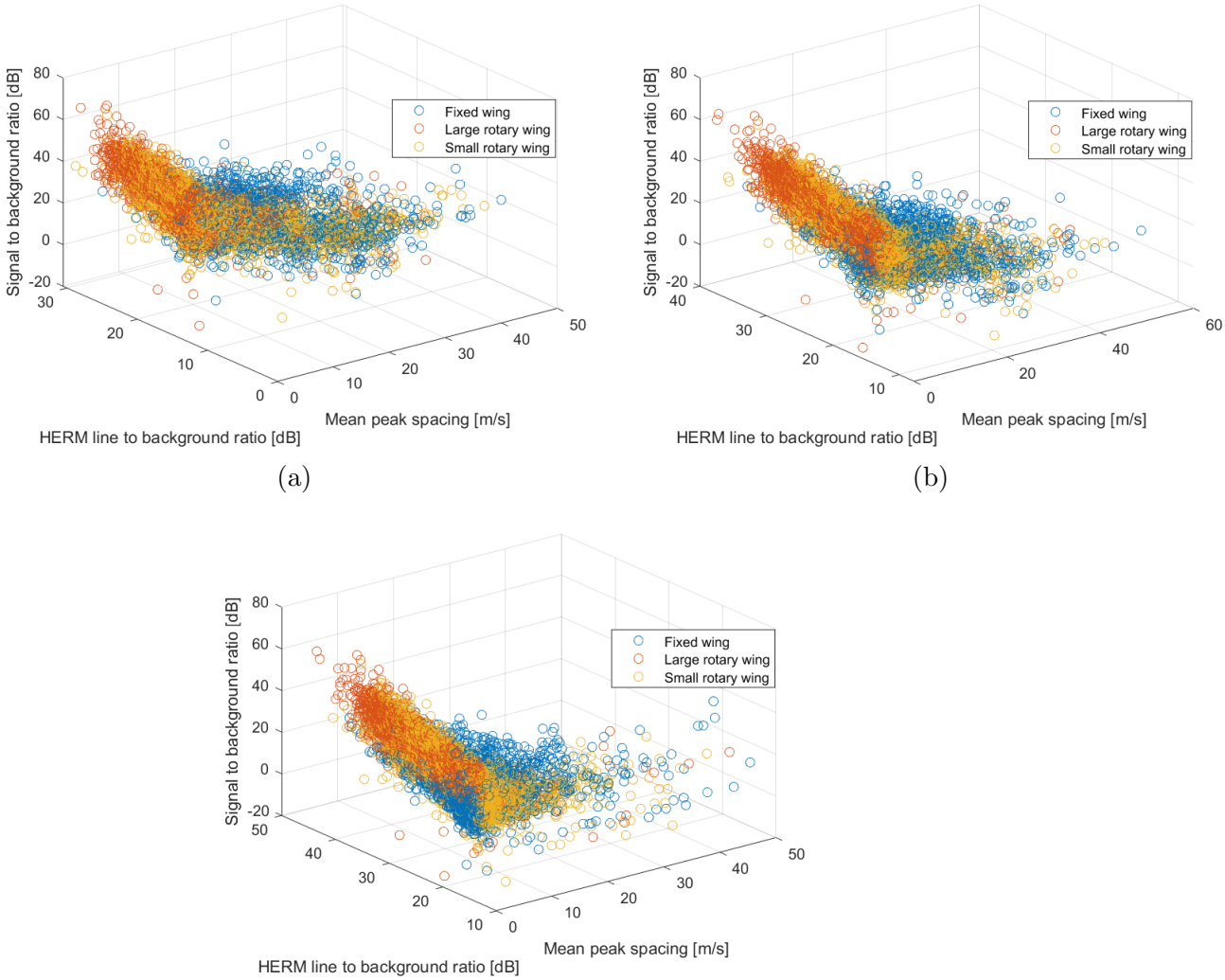


Figure 6.28: Feature plot showing the body signal to background ratio, strongest helicopter rotor modulation (HERM) line power to background ratio, and the mean spacing between Doppler peaks in each spectrogram a) Integrating over 128 pulses, b) Integrating over 512 pulses, c) Integrating over 2048 pulses.

On the other hand, the mean spacing between peaks reduces as integration length

is increased (Fig. 6.27b) due to the increased Doppler resolution and integration gain, but the spread decreases too, which could result in tighter clustering. Fig. 6.28a-c show that increasing the integration length changes the separability of the drone subgroups in 3-D feature space.

6.6.3 Classification Performance as a Function of Integration Length

A decision tree was used to perform classification, trained on the features outlined in Table 6.4. A CART algorithm was used to train the decision trees, using Gini's diversity index as the split criterion [162]. Note that in order to prevent overfitting, the minimum number of branch node observations was limited to 10. For the 9-class problem, this resulted in an average tree depth of 21 stages, where the tree depth is the number of steps between the topmost node and the lowest level leaf. A decision tree was chosen because they are fast to train and test, the results are easy to interpret, and it is possible to measure feature importance.

A decision tree was trained on features extracted from spectrograms produced with a given processing length. This was repeated 50 times, where the particular samples in the training, test, and validation subsets are randomised, to measure the median and range of the resulting F1 scores. The processing length was increased logarithmically from 128 to 4096 pulses, and the entire process was repeated six times in total. Therefore for each level of hierarchy, 50x6 decision trees were trained. Fig. 6.29 shows the F1 score with processing length for drone group, subgroup, and model classification. Overall the F1 scores achieved by the decision tree are lower than those achieved using CNNs. This is likely because the CNN extracts its own features for classification, and can combine them a non-linear fashion, whereas the decision tree is limited to a small number of handcrafted features. As the number of classes increases the F1 score decreases, similarly to when a CNN is used. For all levels of hierarchy tested, performance increases with increasing processing length up

to some optimal, and then decreases. For drone group classification using this data set and classifier, a processing length of 2048 results in the highest F1 score. On the other hand, a processing length of 512 results in the highest median F1 score when drone model classification is performed, improving performance by approximately 18%.

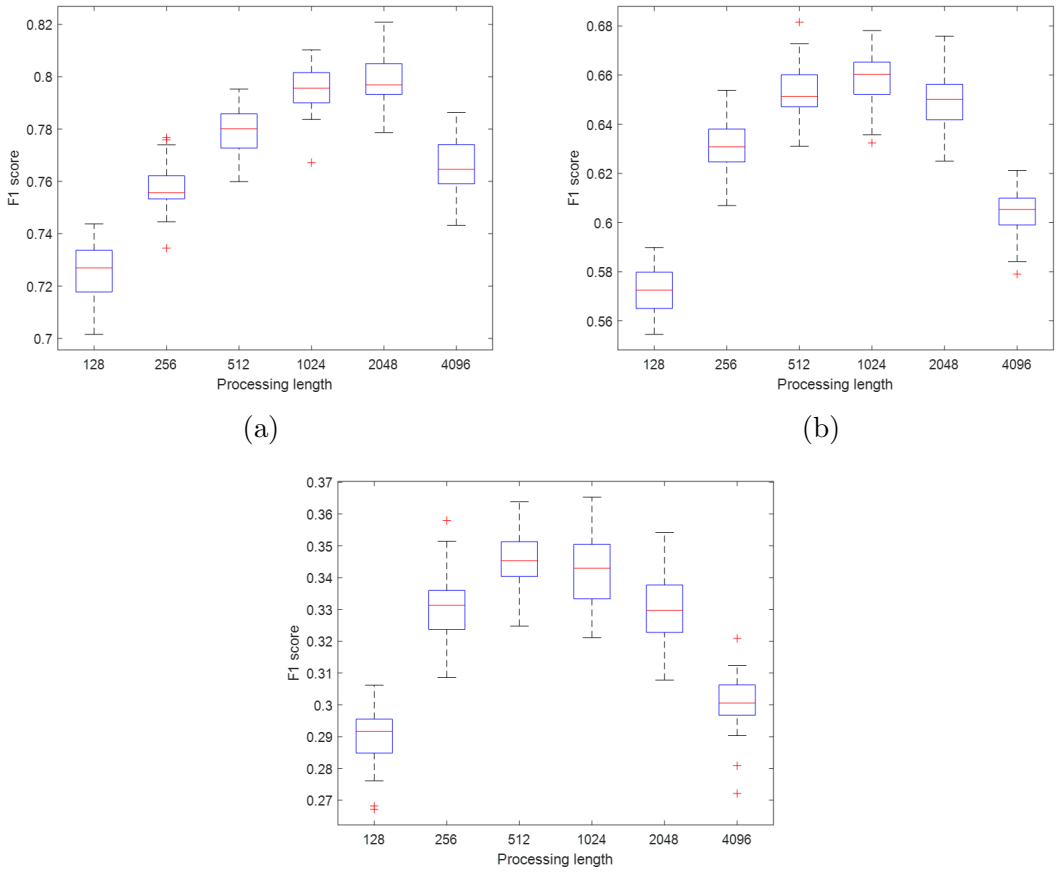


Figure 6.29: F1 score as a function of processing length, in terms of number of samples a) Drone group classification b) Drone subgroup classification (size-based), c) Drone model classification.

The F1 score can be improved by changing the processing interval, however, the optimal choice of processing interval is expected to be a function of model type and trajectory, as shown in Fig. 6.26 and Fig. 6.27. It is therefore difficult to identify a single processing length that will optimise performance over all test samples. In

[180], the authors use multiple processors in parallel, each with different integration lengths, to improve drone detection and tracking performance by maximising the information extracted from the drone signatures. It is therefore hypothesised that classification performance could be improved by training and testing multiple classifiers in parallel, each using features derived using varying processing lengths (from 128 points to 4096 points), and combining the outputs. Two approaches were considered: the first averages the scores output by the six decision trees to provide a single label (Fig. 6.30), and the second follows a majority-rules approach whereby the label declared is the modal output of the classes declared by the six trees (Fig. 6.31). Fig. 6.29 was used to rank the processing lengths for each level of hierarchy to decide which of the six trees to use should the output be multi-modal.

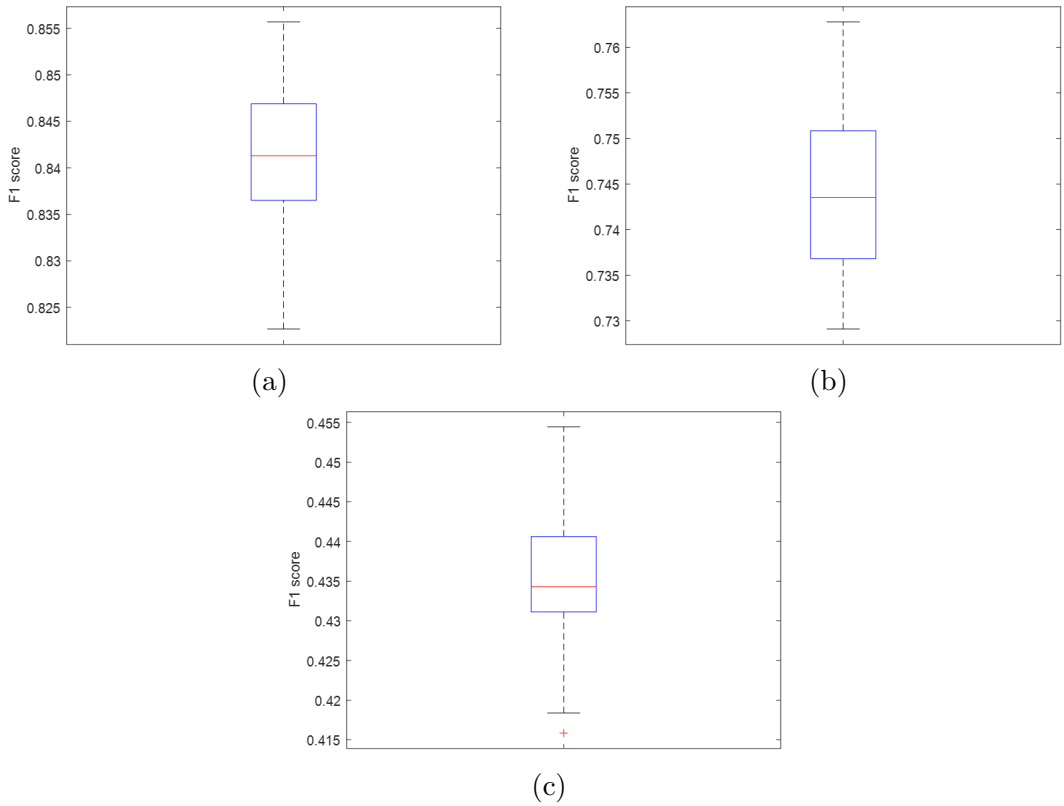


Figure 6.30: F1 score where six decision tree classifiers trained on features derived using various processing lengths are used, and the average score over all six classifiers is used to output the final label a) Drone group classification b) Drone subgroup classification (size-based), c) Drone model classification.

This experiment was repeated 50 times to record the variation in the F1 score. For all levels of hierarchy using multiple classifiers in parallel leads to a significant improvement in the F1 score compared to the case where a single processing length is used. In the case of drone model classification, the median F1 score is improved by approximately 26% when the score is averaged over six classifiers trained and tested using different processing lengths, compared to using a single processing length of 512 pulses. Averaging the score over the six parallel classifiers slightly outperforms the majority-rules approach, although the improvement is not significant.

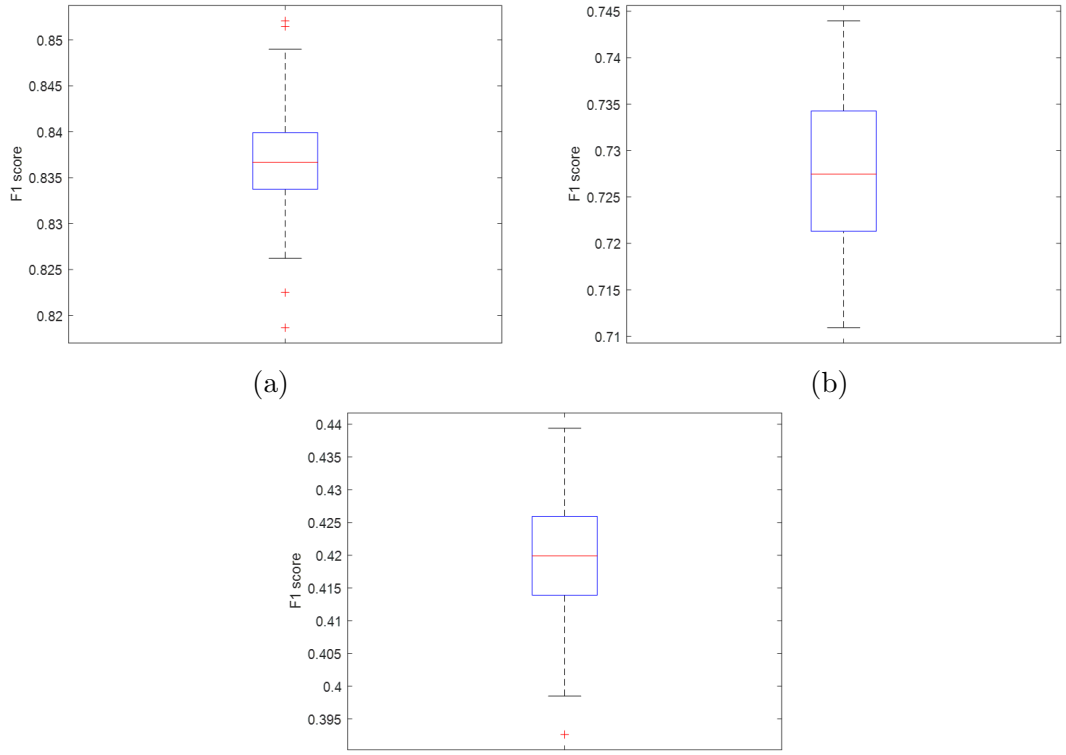


Figure 6.31: F1 score where six decision tree classifiers trained on features derived using various processing lengths are used, and a majority rules approach is used to output the final label a) Drone group classification b) Drone subgroup classification (size-based), c) Drone model classification.

6.7 Chapter Conclusions and Future Work

6.7.1 Conclusion

These findings are the result of the first comprehensive investigation of CNN use for drone-to-drone classification under real operational conditions in the HERM line regime, providing a unique contribution to the rapidly expanding domain of Doppler-based UAV classification. This work has resulted in the following findings:

- **The suitability of CNNs for UAV classification.**

  It is shown that CNNs are highly capable of handling classification between drone groups (F1 score 0.97) and subgroups (F1 score 0.95), demonstrating that it is possible to distinguish between fixed wing and rotary wing drones, and drones with varying numbers of propellers, to a high degree of accuracy. Drone model level classification is also performed, yielding an F1 score of 0.67.

- **Grouping drones based on their weight leads to improved subgroup separation compared to grouping them based on the number of rotors, enhancing performance by 8%.**

  The essence of a classification problem is the separability of target features, where tight clustering in different parts of feature space allows clear boundaries to be drawn between classes. Grouping drones by size improves feature space clustering, due to the effect of weight on the appearance of HERM lines.

- **A trade-off between accuracy and time taken to respond to threats.**

  A data-driven approach was used to recommend optimal dwell-times depending on accuracy requirements. Whilst high accuracy can be achieved using 1 second spectrogram inputs, fusing the output of a short (10 second) series of

spectrograms can improve performance significantly. As the complexity of the problem is increased, more frames are needed to perform reliable classification, and classifying whole tracks can improve performance by as much as 39%.

- **Training with a mixture of SNRs improves classifier robustness to low SNR test data.**
  This validates the results illustrated in Chapter 5 using real data.

- **Decision tree classifiers are suitable for drone group and model classification.**
  Micro-Doppler and signal strength features derived from spectrograms were used for training, as well as kinematic features.

- **The optimal processing length is shown to be a function of trajectory, target type, and the complexity of the classification problem.**
  The relationship between processing length and SNR is explored for different models of drone.

- **Integrating the result of multiple decision trees leads to a significant improvement in performance.**
  Training and testing on multiple processing lengths and fusing the result improves performance by 26%.

6.7.2 Recommendations

This comprehensive investigation has contributed to the community's understanding of drone spectrogram features, resulting in a series of specific recommendations for accurate classification of drones using radar. These are to:

- Conduct training and testing on a mixture of high and low SNR data, to improve robustness to noisy test data.

- Average the classification output over a series of spectrograms in order to boost performance. This becomes particularly important when considering model level classification, placing limits on the response time.
- Take steps to improve SNR where possible, in order to maximise model-level performance, and follow a multi-stage classification approach.
- Consider grouping similar models of drone (for example, DJI Mavic 2 Pro and DJI Phantom 3), for applications where achieving model-level classification is not of significant operational importance. The effects of these factors are expected to be more severe as the complexity of the problem increases, becoming most significant as the number of classes increases and SNR decreases.

6.7.3 Future Work

A number of avenues for potential future work have been identified, relating to the format of the input data, the approach to classification, and scenarios for further data collection.

- **A full parameter sweep to optimise classification performance.**
  The classifier design choices made in this chapter used results from previous work on bird vs drone classification. However, the previous classifiers were trained and tested on L-band radar data, with a finer Doppler resolution, therefore the same results may not hold true. A more in-depth investigation is required across all training hyperparameters and processing parameters to optimise CNN performance.
- **Investigation into the most appropriate radar parameters and design for a given classification problem.**
  To a degree, this will be set by aspects such as detection and classification

range and coverage, but a range of design choices will remain. The choice of radar parameters will set limits on the Doppler resolution, integration time, and SNR achievable, all of which will change the appearance of the drone spectrograms [88] and have profound implications on performance.

- **Acquisition of additional training data.**

  Further trials with drones performing a wider variety of manoeuvres could provide a more varied data set for classification and potentially reduce any overfitting on trajectory-based characteristics or environmental conditions such as wind speed [152] that are not inherent of the drone type.

- **Evaluation of performance under more challenging conditions.**

  Performance requirements will be application specific, and more work is needed to demonstrate trustworthy levels of performance. The radar used in this work has a maximum range of detection of only a few hundred metres. In Chapter 7, data collected using an L-band staring radar will be used for drone classification at ranges up to 5 km. Future work at Ku-band could explore the classifier's generalisation capacity to a wider range of drone models, environments, and clutter conditions.

- **Consideration of alternative data representations.**

  The CNN performs both feature extraction and classification, thus there should be an emphasis on the appropriateness of the form of input data. Potential alternatives to spectrograms include range-Doppler maps, cepstrograms, cadence velocity diagrams, kinematic parameters, and trajectories [181].

- **Fusion of multiple data representations.**

  Running multiple classifiers on different forms of pre-processed data - perhaps in a cognitive or adaptive architecture - could form an avenue for further research, and could be a way of extracting maximum performance for a given radar.

Chapter 7

Classification Under Real and Dynamic Conditions

7.1 Introduction

It has been demonstrated that CNNs show great promise for bird-drone classification. In Chapter 5, a multi-frame CNN classifier was trained using data from an L-band staring radar, where it was demonstrated that performance was comparable to traditional machine learning techniques, for both a two class (drone-vs-bird) and three class (drone-vs-bird-vs-other) problem [42]. However, this was a proof-of-concept result, obtained for a small data set consisting only of a single model of drone. In this chapter, a control CNN classifier is trained using spectrogram data provided by Aveillant from the L-band GameKeeper staring radar, and a comprehensive and step-by-step analysis of the model is carried out to identify its capabilities and investigate its limitations. Performance will be evaluated upon a comprehensive data set representing a range of environments, target ranges, and target types, in

order to investigate the classifier's ability to generalise to unseen environments and unseen models of drone. Additionally, it is demonstrated that high performance can be achieved with an update rate equal to a single frame, following the initial latency at the beginning of a track.

The signal to noise ratio (SNR) of the data set used for training in Chapter 5 was high, with clearly visible HERM lines in the DJI Inspire 1 spectrograms. A key enabler for drone-vs-bird classification is the presence of HERM lines, caused by drone rotor blade rotation. However, as the RCS of blades is usually in the order of 15-20 dB below that of the drone body, acceptable SNR margins and the target distances at which they can be obtained become narrower. It has been demonstrated in Section 5.5 through the addition of Gaussian noise that performance reduces with decreasing SNR, and that this problem is exacerbated when the classifier is trained exclusively on high SNR data [7]. In this chapter, data set A described in Section 5.2.2 is used to train a CNN, to validate the results presented in Section 5.5 (albeit using different processing parameters) on real data collected from a range of high clutter environments.

The term signal to background ratio (SBR) will be used instead of SNR in the remainder of this chapter as there are examples in the dataset where the background is no longer set by the thermal noise power, perhaps due to phase noise [182][183]. Fig. 7.3a and Fig. 7.3c show example spectrograms where the background is heightened in particular spectra, predominantly those where the drone is in a resolution cell that contains high levels of stationary clutter.

In this chapter, the performance of the control classifier is reported on an extended data set as a function of both SBR and drone model. Following this, an additional classifier is retrained on a subset of an extended data set. Performance is evaluated under increasingly demanding conditions, evaluating performance as a

function of model and SBR, as well as robustness to heavy clutter environments. First results of quantifying the false alarm rate on a bird data set are presented, and the reliability of reported scores as confidences is investigated.

A major contribution of this chapter is a proposed fusion approach for combining the low false alarm rate of the control classifier with the high true positive rate of the retrained classifier. This approach uses a dual-stage classifier, with a simple machine learning algorithm to assign the test sample to a particular CNN classifier, depending on the features present. Two approaches are considered: a fusion approach that calculates a weighted sum of the scores declared by both classifiers, and a dynamic classifier selection approach that makes a choice of classifier by drawing boundaries in feature space.

Finally, the problem of L-band drone-vs-drone classification is considered, whereby a classifier is trained to distinguish between birds and small, medium, and large drones. Short range drone model classification was investigated in Chapter 6, using a Ku-band radar and spectrograms with a duration of a single second. In this chapter, the results are validated using a different radar system with a lower carrier frequency and a longer-dwell time, attempting drone size classification with target ranges of up to 5 km. This could have significant consequences for drone surveillance applications where it may be beneficial to be able to estimate the size of a drone and therefore assess the risk associated with it with sufficient time to formulate a response.

The layout of the chapter is as follows: Section 7.2 gives an overview of the methodology, including the spectrogram parameters, multi-frame CNN implementation details, and an overview of the training and test data sets. Section 7.3 presents the results of training a control classifier on the high SBR data set (henceforth referred to as data set A), and an investigation of the robustness under increasingly

demanding conditions on an extensive test data set (henceforth referred to as data set B). Section 7.4 contains the results of retraining the classifier on additional bird and drone data. The classifiers are then compared in Section 7.4.4, where the false alarm rate is quantified on an extensive set of bird data. In Section 7.5.1, a novel fusion approach between two CNNs is proposed, whereby a dual-stage classification approach is used to make an intelligent choice of CNN classifier. Section 7.6 presents the results of training a drone size classification algorithm. The chapter conclusions are then given in Section 7.7.

7.2 Methodology

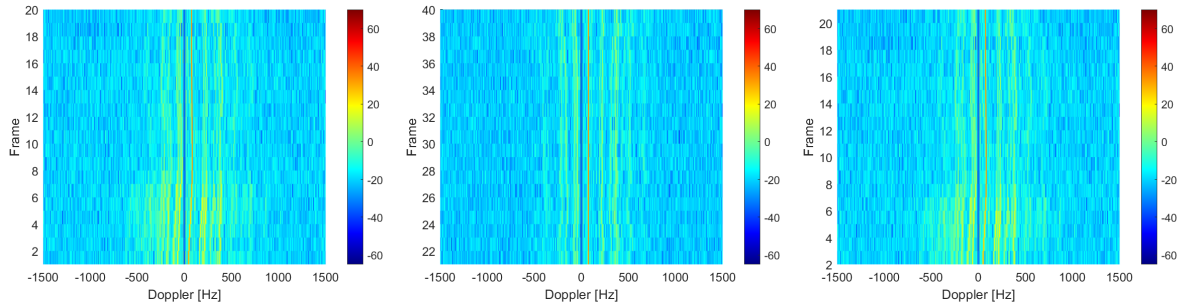
A CNN classifier (AlexNet) is tested on both an existing data set (data set A, Section 5.2.2) and a supplementary data set (data set B, Section 7.2.2) provided by Aveillant. Data set A consists of DJI Inspire 1 and DJI Inspire 2 drones at ranges in the order of 0.5 km to 2.5 km, under similar low-clutter environments. On the other hand, data set B consists of a wide variety of drone models, covering a spread of ranges (from 0.3 km to 5.0 km) and rural, urban, and industrial environments. Performance is evaluated as a function of dwell time and SBR, with the primary aim of assessing the classifier's ability to generalise to a more complex data set. The implications of this for counter-UAV applications will be considered, and recommendations will be formulated for training a robust classifier.

7.2.1 Spectrogram Parameters

Each frame of data provided contains data from 4096 pulses with a 50% overlap between adjacent frames. Data was provided not only for the resolution cell where

the target is present, but also for an additional 6 adjacent cells (in range and angle) per frame. Only the data from the central resolution cell was selected to form the spectrogram, providing a time series matrix of size Nx4096, where N is the number of frames belonging to the track. A high pass filter with a cut-off frequency of 15 Hz was used to remove stationary and low velocity clutter, a Blackman-Harris window was applied to the time series data, and a 4096-point FFT was used to transform the time series data into the frequency domain. As the PRF of the radar is 7353 Hz, this equates to a CPI interval of 0.5571 seconds giving a Doppler resolution of 1.8 Hz. The 50% overlap between frames equates to an update rate of 0.279 seconds.

There is no overlap between the spectrograms used for training the classifier. For a spectrogram length of L, a file containing N frames will produce $\lfloor N/L \rfloor$ spectrograms, discarding the remaining frames. Two example consecutive spectrograms are shown in Fig. 7.1a and Fig. 7.1b, where L is 20. Generally, there is no overlap between the spectrograms used for testing the classifier, to reduce computational load and correlation. However, in Section 7.3.2, a 19 frame overlap is used to demonstrate that the classifier is capable of operating with an update rate equal to a single frame. Fig. 7.1c shows an example spectrogram with a 19 frame overlap relative to the spectrogram shown in Fig. 7.1a.



(a) (b) (c)

Figure 7.1: Example I1-D spectrograms, each 20 frames long a) The first spectrogram in a track b) The second spectrogram in a track where a 20 frame update rate is used c) The second spectrogram in a track where a single frame update rate is used.

Control target (drone) labels are assigned using the onboard GPS and correlating this with the radar tracks. Opportune targets are manually labelled using ground truth information [161], and using features such as track length, height above ground, and RCS.

Target information files for each track were extracted from the tracker, and provided along with the time series data. These files contain the target's positional coordinates relative to the radar, range gate, beam, Doppler bin, and radial speed. This information is provided for every frame of data within a track. The target information file is used to provide the Doppler bin of the target, for calculation of the body power return.

In this chapter, the definition of SBR is the ratio of the target power at the output of a 4096-point FFT to the background power (usually but not always due to noise) averaged over the first and last 300 Doppler bins of the spectra. The Doppler bin output in the target information file is used to select the target power, taking the maximum signal within 15 bins either side of the tracker bin.

7.2.2 Data Set B Coverage

Data set A (Section 5.2.2) consists of data from two models of drone recorded at semi-rural sites. A subset of this data was used to train classifiers in Chapter 5 and Section 5.5, and the need to validate the reported performance on an extended data set was noted. Additional real data measurements were provided for new radar sites, additional drone platforms, and birds. Some of the data in the original data set contained artefacts, making them inappropriate to use for training and testing purposes. Three major types of artefact were found: areas of high local stationary clutter where even the body return is lost below the high background, gaps in the spectrogram where the target is not present, and clutter lines at constant Doppler.

Each track was considered on a case by case basis, and discarded from the data set if it contained severe artefacts that may have the potential to impact on performance.

The amount of data retained for each target is summarised in Table 7.1, where the drone models have been anonymised for confidentially purposes, but an estimate of diameter is given when available. The drone models provided span a range of diameters and weights, from smaller (e.g. P4-D) rotary wing drones to larger (e.g. AV-D) rotary wing drones, as well as a fixed wing model (Z1-F). One can expect the smaller models of drone to be more difficult to detect, and to provide a significant challenge to distinguish from birds due to a lack of micro-Doppler [88]. In order to investigate performance as a function of model size the drone models were grouped into three categories: small, medium, and large drones, based on their diameters relative to the I1-D and I2-D. The groupings are shown in Table 7.2.

Table 7.1: Summary of the drone models included in data set B, their diameter, the amount of data recorded for each model, and the approximate span of ranges at which data was collected.

Class	Diameter (m)	Number of frames	% of Total Frames	Minutes of data	Range span (km)
Il-D/I3-D/I4-D	0.5	23810	24.1%	111	0.3-5
P3-D/P4-D	0.4	14861	15.1%	69	1.5-5
G-D	0.9	17796	18.0%	83	1.4-3
O1-D	unavailable	14820	15.0%	69	1-1.8
DJIM200-D	0.9	5574	5.6%	26	1.5-4.3
S900-D	0.9	11947	12.1%	56	0.5-4.3
ZI-F	unavailable	3502	3.5%	16	0.8-1.5
AV-D	2.14	4375	4.4%	20	1.2-5
Car	N/A	2006	2.0%	93	N/A

The sites at which the data were collected span a range of environments, including airports, rural environments, and dense urban and industrial areas. Table 7.1 shows an approximation of the spread in range of the targets in the data set provided. A wide spread of ranges is provided for the AV-D, I1-D, P4-D, and DJI-S900 models, where for the I1-D a substantial amount of data is beyond 3 km in range. A wide range in Doppler velocity (from 0 to 25 m/s) is also exhibited, suggesting varying blade rotation rates hence a spread of micro-Doppler signature bandwidths.

Table 7.2: Size groupings of the drone models represented in data set B, based on diameter.

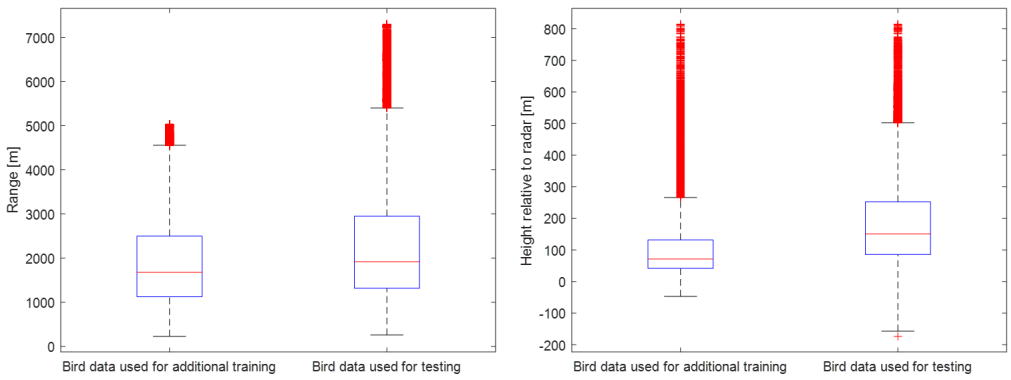
Small (<0.5m)	Medium (0.5 to 0.75m)	Large (>0.75m)
P3-D	I1-D	G-D
P4-D	I3-D	DJIM200-D
M2-D	I4-D	AV-D
	RC-F	S900-D
		Z1-F
		O1-D

Additional bird data were used to balance the bird and drone classes for training the retrained classifier, and for balancing the classes in the test data set for a more reliable measure of accuracy. A staring radar installed on a rooftop at the University of Birmingham campus is used to conduct routine trials for gathering data of control test targets and opportune birds [64]. Approximately 5 hours of opportune bird data were recorded using this radar, collecting data over a section of Birmingham city, resulting in 3134 bird spectrograms used to balance the bird and drone classes when training the classifier presented in Section 7.4. As shown in Fig. 7.2, the birds fly at various ranges and heights within the surveillance volume.

A radar trial using an Aveillant staring radar located at Cranfield Airport obtained data for a control drone and birds of prey (an African Peregrine Falcon, a Saker-Gyrfalcon hybrid, a Common Buzzard, and a Harris's Hawk) [184][64], along with recordings of opportune birds. Opportune birds included species such as Car-

rion Crow, Common Wood Pigeon, Western Jackdaw, Red Kite, and Lesser Black-backed Gull, among others.

From this trial, approximately 4.5 hours of opportune bird data were recorded, resulting in 110 different bird tracks and 1631 spectrograms. In addition to this, the data from control birds amounts to around 20 minutes of recorded data, producing 195 spectrograms belonging to known species of birds, corresponding to 5 individual tracks. These data were used for balancing the test subsets, in order to give a more reliable measure of accuracy and false alarm rate. As shown in Fig. 7.2, the opportune birds fly at various ranges and heights within the surveillance volume.



(a) (b)

Figure 7.2: Coverage of the bird data in data set B in a) Range, b) Height relative to the radar.

7.2.3 Example Spectrograms

Fig. 7.3a-c show some example near range spectrograms (<1 km) for the Z1-F, S9-D, and I1-D models, respectively. Note that for visualisation purposes, these spectrograms are produced over the entire track, and the stationary clutter has not been filtered. All three spectrograms exhibit strong HERM lines at this range, simplifying the bird-vs-drone classification problem when this is the case. However, note the presence of decreased signal to background ratio (SBR) when the drone sits in a resolution cell containing high power in the central Doppler bin, particularly

in Fig. 7.3a where the background power is raised in frame 130 to frame 330. It is expected that classification will be more challenging for these parts of the track, as the HERM lines are barely visible above the background, with SBR decreased by approximately 20 dB in Fig. 7.3a.

HERM lines are shaped by the radar parameters (frequency, dwell time etc.) and by the target parameters [81]. Note the differences in the HERM lines between the three models in Fig. 7.3a-c –the number of blade returns, the width of the HERM lines, and the spacing between harmonics are all a function of model type as well as flight manoeuvre, since the spacing between harmonics can be used to estimate the rotor rotation rate, which is dependent on drone size [82].

The difference in power between the body return and the first rotor harmonic is also a function of model type, as shown in Section 6.6, as the power returned by the blades depends on their shape, size, and material. These features could potentially be exploited to distinguish between different models of drone.

Fig. 7.3d-e show example spectrograms where the HERM lines are less visible. Fig. 7.3d is an example P1-D spectrogram with a mean range of 2.9 km – the P1-D is considerably smaller than the I1-D platform, and no rotor returns are visible at this range, with a body SNR of approximately 33 dB. Cases such as these are expected to pose significant classification challenges, as it has been demonstrated that HERM lines are a key feature for robust classification. Fig. 7.3e is an example I1-D spectrogram at a mean range of 3.3 km. Comparison with Fig. 7.3c demonstrates a lower drone body SBR and HERM line SBR, with fewer HERM lines visible above the noise floor. Fig. 7.3f shows an example opportune bird spectrogram used for retraining. The recorded data covers a range of SBRs and radial velocities.

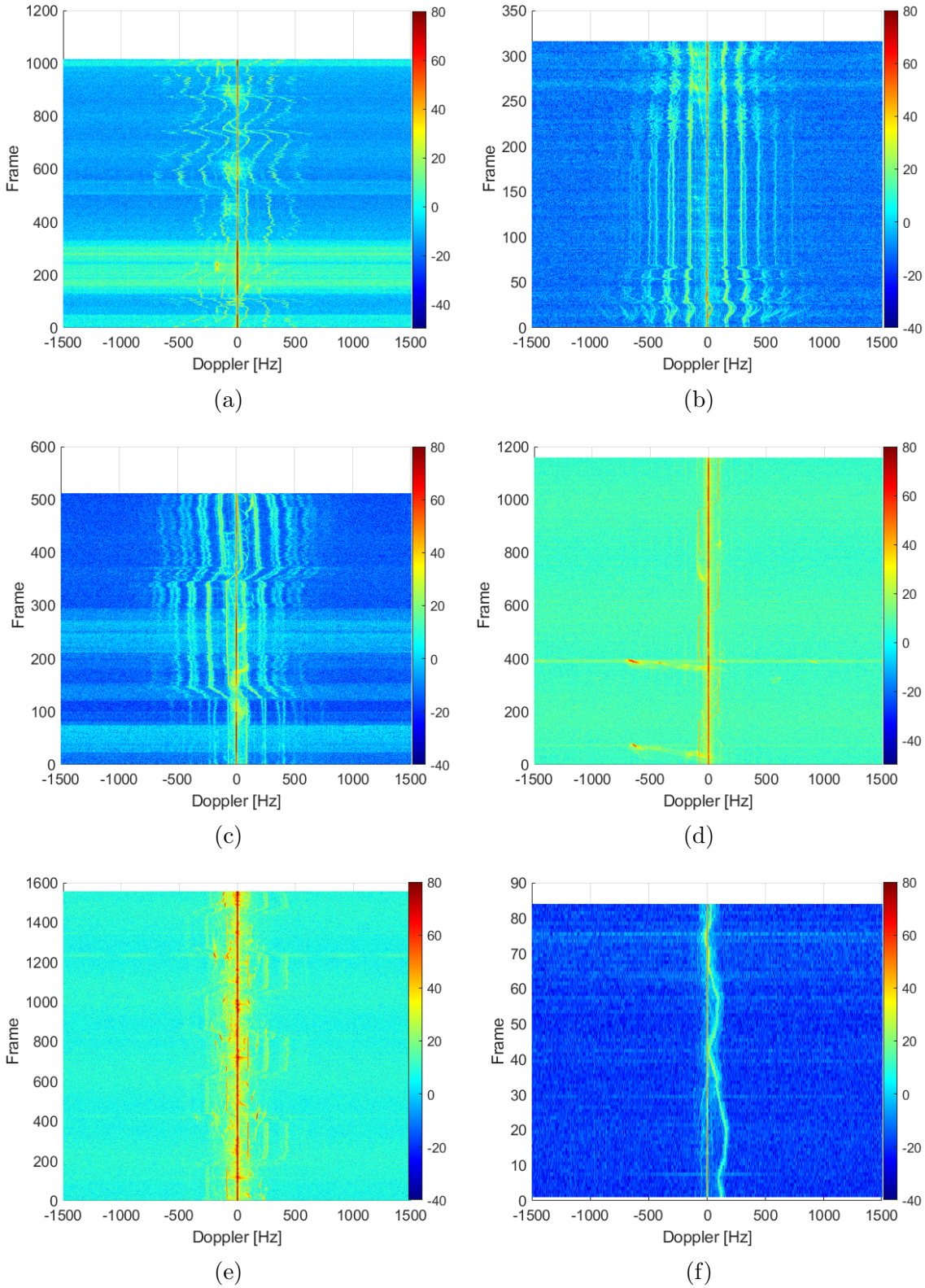


Figure 7.3: Example spectrograms a) Z1-F (946 m) b) S9-D (753 m) c) I1-D (553 m) d) P1-D (2.9 km) e) I1-D (3.3 km) f) Opportune bird spectrogram.

7.2.4 Implementation Details

The CNN algorithm used in this chapter is the same as in Chapter 5 and Chapter 6. As in Chapter 6, performance is reported in confusion matrices and summarised using F1 scores. AlexNet [8] has been utilised, using transfer learning to initialise the network weights and biases, as this algorithm has been shown to perform well in real (although high SBR) conditions, and to be particularly robust to noisy test data [43]. Raised background levels may impact upon classification performance as HERM lines could be masked, and also the background has different underlying statistics compared to thermal noise.

As described in Section 2.7, time series data is extracted from the range, azimuth, and elevation bin containing the target, and is used to plot spectrograms that form the input to the multi-frame CNN classifier.

7.3 Control Classifier Results

First, a classifier was trained exclusively on data from data set A to establish a baseline performance on a reduced data set. High performance is expected upon this data set as a subset was used (albeit with different processing parameters) to train high performing classifiers in Chapter 5. The classifier's capacity for generalisation is then evaluated by testing the classifier on data set B and identifying any limitations.

7.3.1 Training the Control Classifier

A classifier with the architecture discussed in Section 3.4.3 was trained on data set A. Performance was investigated as a function of spectrogram length in order to

determine the dwell time to use throughout the work.

Choice of Dwell Time

A simple parameter sweep was used, where the number of frames was varied from 10 to 70 frames. As shown in Fig. 7.4, a dwell time of 20 frames seems to be a good trade-off between high performance and low initial latency. Note also that as the spectrogram length is increased there are fewer spectrograms produced per track, which could potentially lead to a decrease in performance due to lack of training data. For consistency, the same parameters were used for all classifiers presented in this chapter.

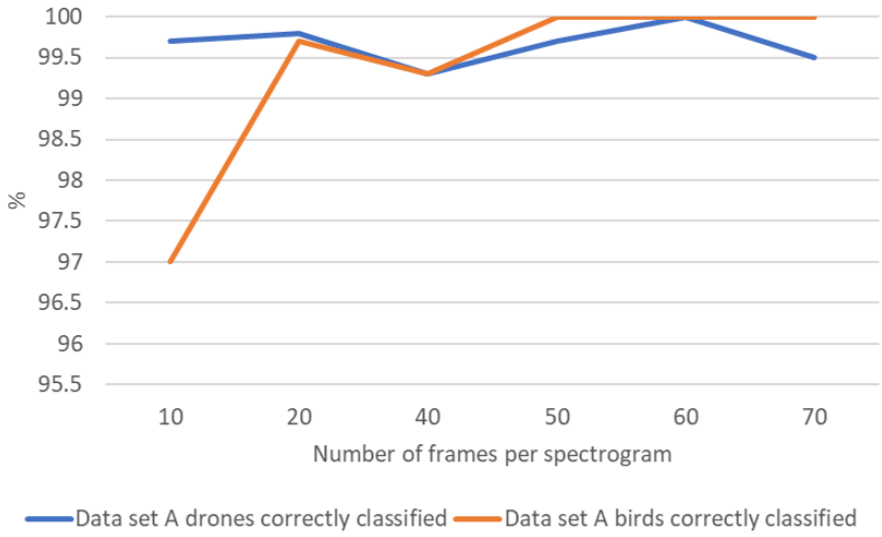


Figure 7.4: Performance on data set A as a function of spectrogram length.

7.3.2 Testing the Control Classifier on Data Set A

The chosen spectrogram length of 20 frames corresponds to a 5.5 second duration per spectrogram. In total there were 1096 bird spectrograms and 2669 drone spectrograms. 60% of the tracks were used for training, 30% for testing, and the remaining

10% for validation during training. Here, spectrograms are not overlapped, such that there is a 20-frame latency between outputs. Care was taken to ensure that test and training spectrograms were taken from different files, to avoid overfitting. A summary of the number of spectrograms used for training is given in Table 7.3.

Table 7.3: A summary of the data used for training the control classifier.

		Number of files	Number of spectrograms	Maximum SNR [dB]	Minimum SNR [dB]	Mean SNR [dB]
Target class	Bird	53	683	44.2	21.6	30.7
	I1-D	46	865	64.2	9.7	52.7
	I2-D	77	713	64.1	22	51.0

Fig. 7.5 demonstrates high performance, where all but one bird spectrogram and two drone spectrograms were correctly classified, giving a very high recall and precision. It is important to understand how these results have been achieved, by investigating why the classifier may correctly or incorrectly classify a given spectrogram.

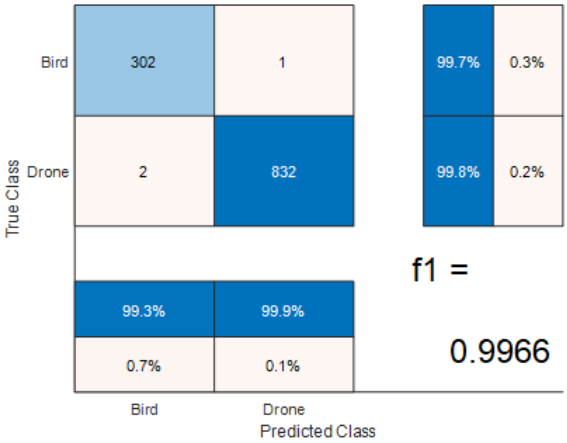


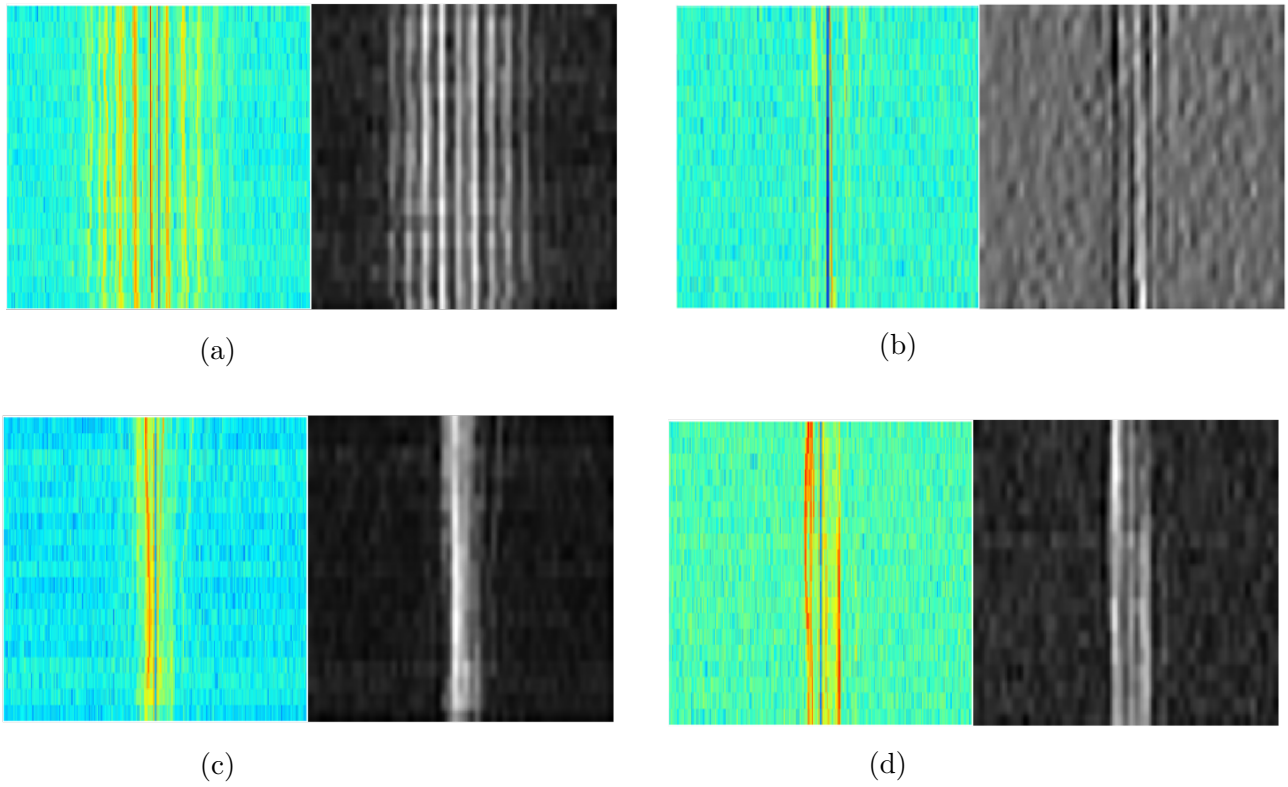
Figure 7.5: Performance of the unweighted control classifier on data set A, using a 20-frame update rate.

Viewing Network Activations for Test Spectrograms

Due to the 'black box' nature of CNNs it is not possible to assert exactly why particular decisions were made, however, many CNN visualisation techniques have been proposed in the literature [185][186]. A simple visualisation approach is chosen, whereby a test image is compared with the network activations, to indicate which features the network has learnt [187]. In this work, only the channels in the first convolutional layer are considered, since in deeper layers features become increasingly non-linear thus less explainable. There are 96 channels in the first convolutional layer, and rather than viewing all 96 feature maps only the channel that results in the maximum activation for a particular input image is shown. Fig. 7.6a is an example of a correctly classified I1-D spectrogram, along with the corresponding activations in the maximum activated channel in the first convolutional layer. Areas of high activation are shown in white, whilst low activation is black. The classifier acknowledges the HERM lines in this drone spectrogram, since they are all highlighted bright white, suggesting that the classifier views them as significant for high classification performance. This supports the hypothesis that the presence or absence of HERM lines is a feature that the classifier has learnt to look for, and that poor performance may be expected when these are not present (e.g. when SBR is low, at long ranges, and for small models of drone).

Fig. 7.6b shows an example of a misclassified I1-D spectrogram, along with the maximum activated channel in the first convolutional layer. Note that the misclassified target has a relatively low body SBR, and the HERM lines are hardly visible above the noise floor. The classifier is at least looking in the target region, picking up most of the body return and highlighting the same areas that we as a human would be looking at. In this case, there is sufficient evidence to suggest that the classifier has not learnt sufficiently robust features to classify drone spectrograms in

the absence of strong HERM lines.



(a) (b)

(c) (d)

Figure 7.6: Input spectrograms and their corresponding maximum activations a) Correctly classified drone spectrogram b) Misclassified drone spectrogram c) Correctly classified bird spectrogram d) Misclassified bird spectrogram.

Fig. 7.6c is an example of a correctly classified bird spectrogram, where the body of the bird appears bright white in the activation map, indicating that the classifier is strongly activated by the body return. The spectrogram in Fig. 7.6d is incorrectly classified – here there is evidence of multiple targets within the spectrogram and the classifier is activated by clutter as well as the body return. Perhaps multiple birds are flying within the same resolution cell, resulting in approximately parallel lines that almost giving the appearance of micro-Doppler. The channel that is maximally activated for this image is the same channel that was activated very strongly by the I1-D HERM lines in Fig. 7.6a, suggesting that there is a filter in the classifier that looks for HERM lines. For the example shown in Fig. 7.6d, this filter is activated by the bird spectrogram, leading to a misclassification. These findings imply that the

classifier could have difficulties when it comes to classifying flocks of birds, suggesting that this should form an avenue of future work.

Performance Using a Weighted Loss Function

Only 30% of the spectrograms used for training the control classifier belong to the bird class. If the training data is skewed towards a particular class, this could lead to bias in the classifier. Data imbalances could be compensated for by over or under sampling the classes, recording or simulating additional data, or modifying the loss function to give equal weight to the classes overall. The classifier was retrained using a modified loss function (Section 6.2.5). Test spectrograms were overlapped by 19 frames, such that a classification label is given every frame.

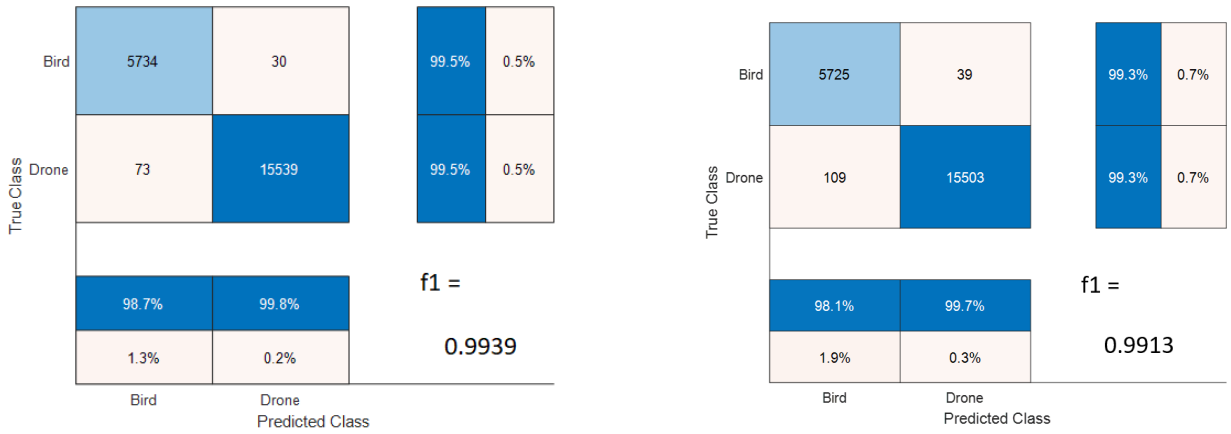


Figure 7.7: Performance of the control classifier on data set A, using a single frame update rate a) Unweighted loss function b) Weighted loss function.

Fig. 7.7 shows the performance of the classifier using the original loss function (Fig. 7.7a) and a weighted loss function (Fig. 7.7b). An F1 score of 0.99 is achieved by both classifiers, comparable to the performance of the same classifier on the test

data set without any spectral overlap. In this case, it seems that the imbalance between the bird and drone class is not severe enough to require weighting the loss function, therefore an unmodified loss function will be used throughout the remainder of the chapter.

Fig. 7.7 demonstrates that it is possible to produce classification results for every frame of the track, reducing the time taken between consecutive declarations from 5.5 seconds to 0.279 seconds. As it is significantly more computationally demanding to produce results for each frame, the remainder of results presented in this work will consider only the case where the update rate is equal to the duration of a spectrogram.

7.3.3 Testing the Control Classifier on Data Set B

The control classifier achieved very high performance on data set A, however this only covered two very similar models of drone at intermediate ranges. A robust classifier ought to be able to classify a wide variety of drone models, ideally having the ability to generalise to classify models of drone that were unseen during training. Data set B was therefore used to test the control classifier to assess its ability to generalise to unseen sites, clutter environments, ranges, and models of drone.

Fig. 7.8 combines the performance on data sets A and B. Note that only 30% of data set B was used for testing here, to allow comparison with Fig. 7.13. A low false alarm rate of 0.5% is maintained, however, less than 62% of drones are correctly classified as such. The classifier was trained on relatively high SBR returns from medium sized drones at intermediate ranges, where the spectrograms all exhibit strong HERM lines.

The poor generalisation is likely due to the fact that there is more variation in data set B compared to the training data, and in the majority of cases the rotor returns are not visible whether due to range, model of drone, or geometry.

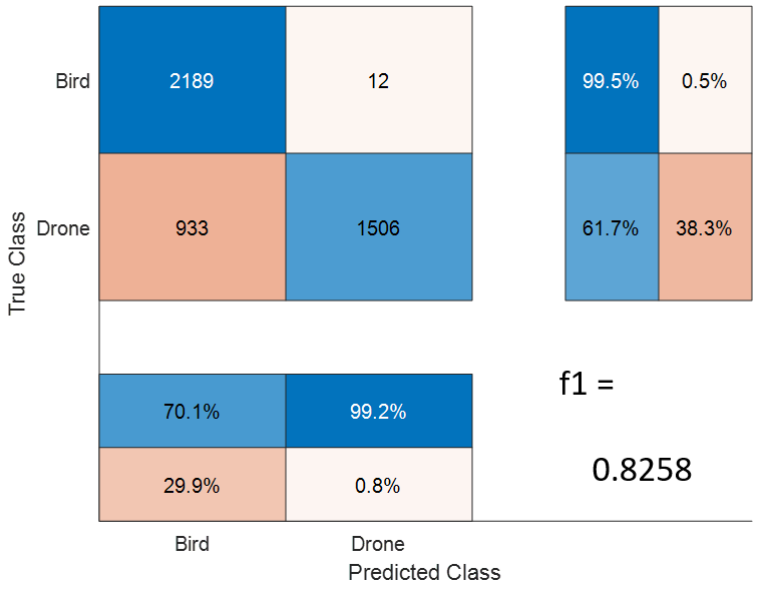


Figure 7.8: Performance of the unweighted control classifier on test samples from data set A and data set B, using a 20-frame update rate.

Performance of the Control Classifier as a Function of Drone Model

Performance was then tested on the entirety of data set B, to investigate performance as a function of model type. Performance is particularly poor for the smaller models of drones, as shown in Fig. 7.9 where only a single P3-D spectrogram and 1.7% of P4-D spectrograms are classified as drones. In addition, the RC-F model is only classified as a drone in 3% of spectrograms, as it has very small rotors hidden behind plastic cladding, resulting in negligible HERM lines. Even for the larger models of drone performance is relatively poor, due to the wide coverage in range.

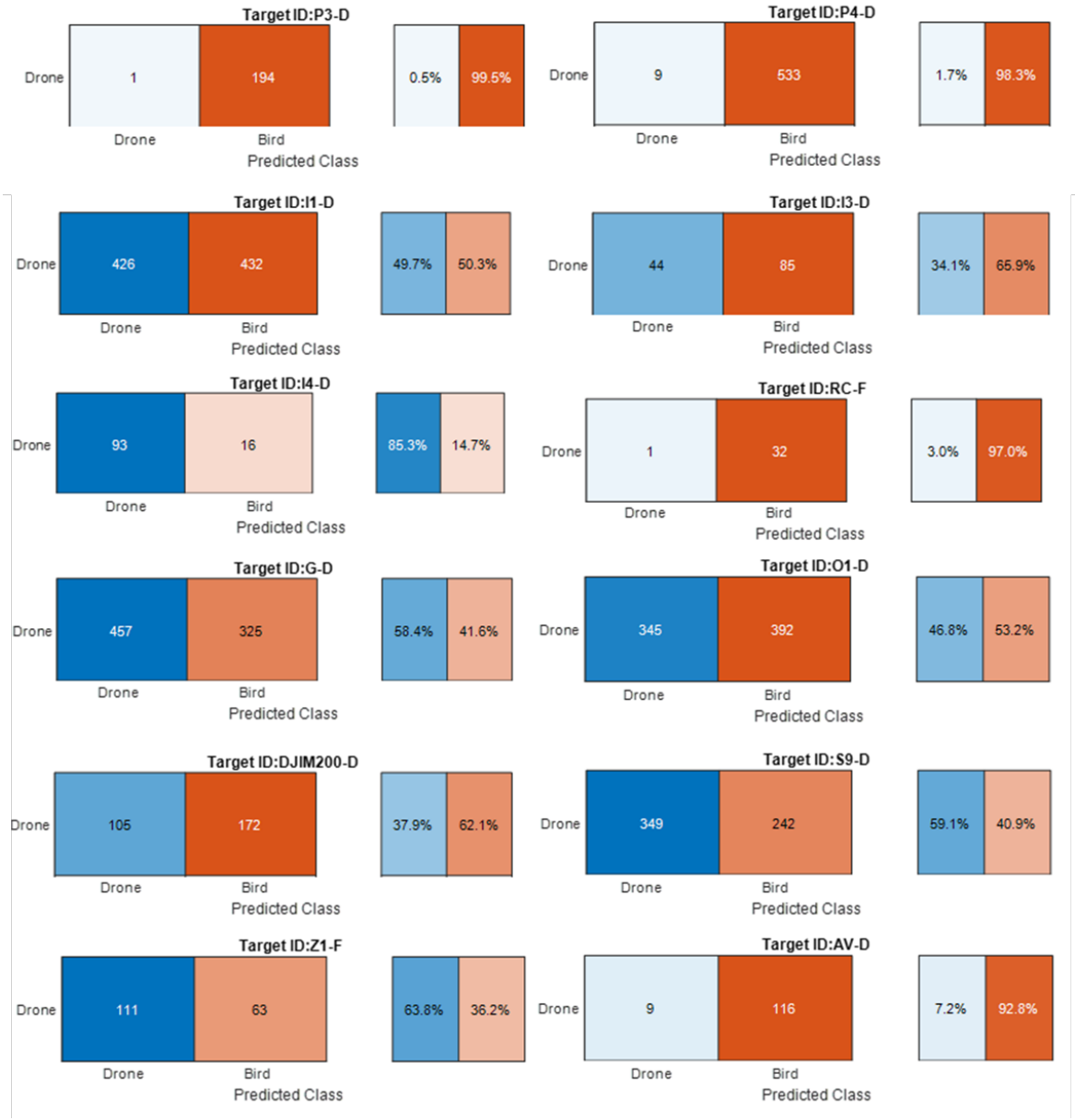


Figure 7.9: Performance results for the control classifier on each drone model in data set B (recall 42.8%).

Performance of the Control Classifier as a Function of Range

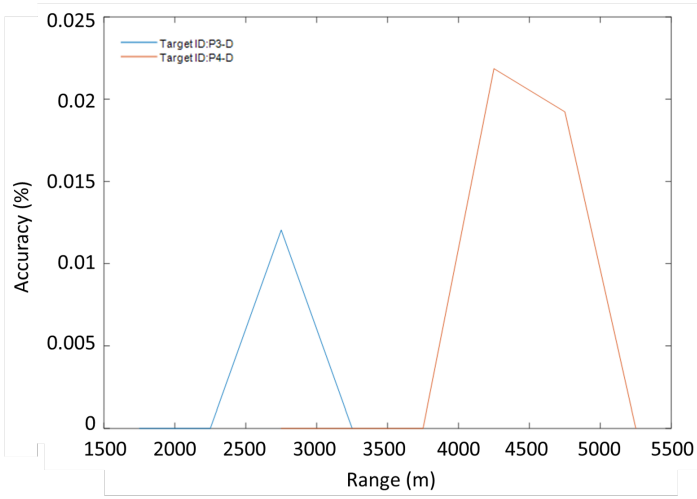
The range of the target was calculated from the target information files provided by the tracker, and was used to plot performance as a function of range as shown in Fig. 7.10a to Fig. 7.10c, where range bins of length of 500 m were used. Fig. 7.10a

shows the results for the two small drone models – one may expect to see performance decreasing with range due to beam broadening and decreased target power, however, this is not the case, and performance is better (although still poor) at the intermediate ranges. As range (R) increases, the power of the target return will decrease whilst the thermal noise stays the same, typically resulting in a decrease in SBR. However, when the clutter within a particular resolution cell results in a background that exceeds the power of the thermal noise, the SBR may not follow the same relationship with range. It is therefore suggested that fluctuations in signal background mean that performance does not necessarily decrease with increasing range.

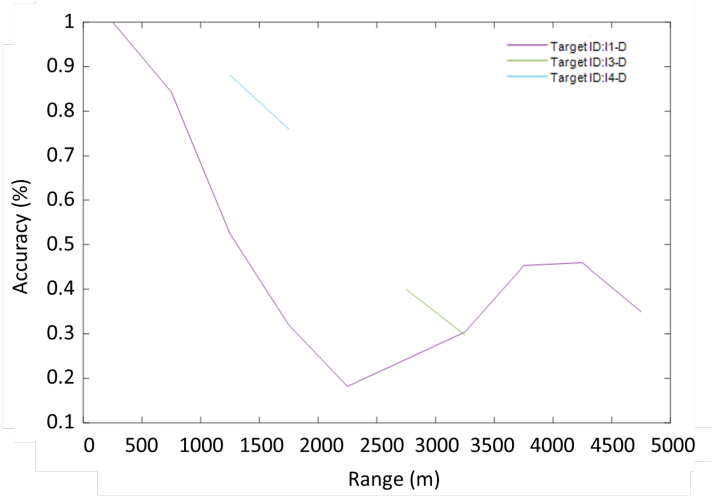
Fig. 7.10b displays accuracy as a function of range for the medium sized drones, with the exception of RC-F since all data for this model was within a single range bin. Broadly performance decreases with increasing range as expected, although again there is some variation possibly due to local background levels.

Fig. 7.10c displays accuracy as a function of range for the large sized drones. Again, performance largely decreases with increasing range with the exception of the Z1-F drone.

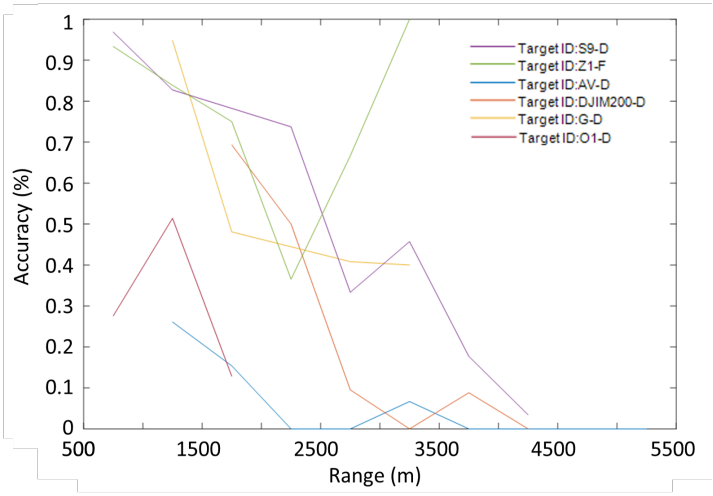
These results suggest that local clutter environments [64] and background levels may impact upon classification performance. In order to investigate this further, performance was evaluated as a function of SBR, exploring the extent to which size of drone, range, and background may affect performance.



(a)



(b)



(c)

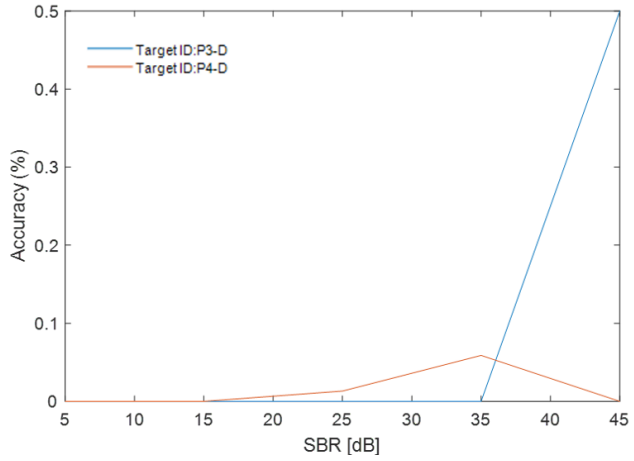
Figure 7.10: Control classifier performance as a function of range for the a) Small drones in data set B b) Medium drones in data set B c) Large drones in data set B.

Performance of the Control Classifier as a Function of Signal to Background Ratio

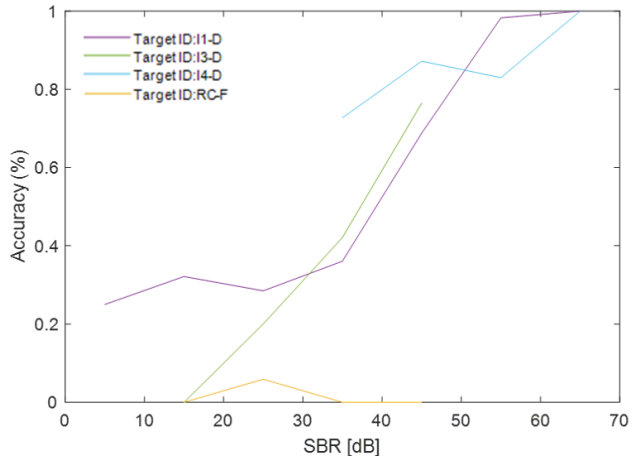
The SBR was calculated using the method outlined in Section 7.2.1. Performance is shown as a function of SBR in Fig. 7.11a to Fig. 7.11c. Fig. 7.11a shows the results for the two small drone models where, as expected, performance increases with increasing SBR.

Fig. 7.11b displays accuracy as a function of SBR for the medium sized drones. Performance and SBR are strongly correlated, with the accuracy increasing with increasing SBR with the exception of the RC-F drone, where performance was very poor regardless of SBR due to the size of the blades. The results imply that an SBR of 50-60 dB is sufficient to achieve high classification performance for the I1-D drone. Recall that in Section 5.5 it was demonstrated that an SBR of 29 dB was sufficient to correctly classify 90% of I1-D spectrograms, however, this was a lower limit only, as the SBR was reduced artificially through the addition of Gaussian noise, without taking into account the effects of beam broadening or increased clutter. In Section 5.5, the training and test data sets are collected under the same conditions, at the same site, with the drone flying along the same trajectory whilst in this work there is a greater deal of separation between the training and test data sets. In addition to this, different processing parameters were used to generate the spectrograms used for training and testing in Section 5.5, further complicating comparison.

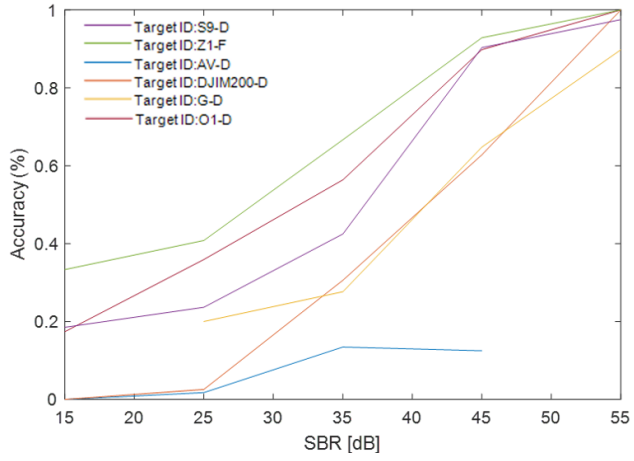
Fig. 7.11c displays accuracy as a function of SBR for the large sized drones. Again, it is the case that performance is heavily impacted by SBR. High accuracies can be achieved for certain models of drone (S9-D, Z1-D, and O1-D) at an SBR of 45 dB and above.



(a)



(b)



(c)

Figure 7.11: Control classifier performance as a function of signal to background ratio (SBR) for the a) Small drones in data set B b) Medium drones in data set B c) Large drones in data set B.

Scores Output by the Control Classifier

In order to improve stability of the classification output over time, successive outputs can be combined sequentially such that the score is propagated over an entire track. Fusion approaches could also be used to combine the output across multiple classifiers. This will improve stability as each classifier approximates the target function differently, thus errors may cancel out leading to a correct output. There are many methods of performing fusion at the level of classifier outputs: one can fuse the class labels themselves, use techniques based on class ranking, or fuse the class probabilities [188]. Bayesian fusion methods are commonly used to combine the likelihoods across multiple classifiers, resulting in a single posterior probability [188][189]. Of course, fusing the outputs in this way requires a measure of confidence in the classification scores as confidences.

In order to qualitatively assess the reliability of the classifier scores, the scores declared by the control classifier are presented in histograms. Fig. 7.12a shows the case where the true class is bird and the spectrograms are correctly classified, whereas Fig. 7.12b shows the case where the true class is bird and the spectrograms are incorrectly classified as 'Drone'. Fig. 7.12c-d show similar for the case where the true class is drone, where the spectrograms are correctly classified in Fig. 7.12c and misclassified in Fig. 7.12d. The results demonstrate that the classifier is generally very confident in its decision, even when it is misclassifying a target, with the classifier reporting a score of over 0.95 for 58% of the birds falsely declared as drones, and for 80% of the drones falsely declared as birds. In [190], it is demonstrated that CNNs regularly overestimate the scores, and the outputs need to be calibrated before interpreting the scores as confidences. This suggests that a fusion approach that uses the scores as a measure of prediction confidence is unlikely to be suitable for the multi-frame CNN classifier, and that class label fusion may be more appropriate.

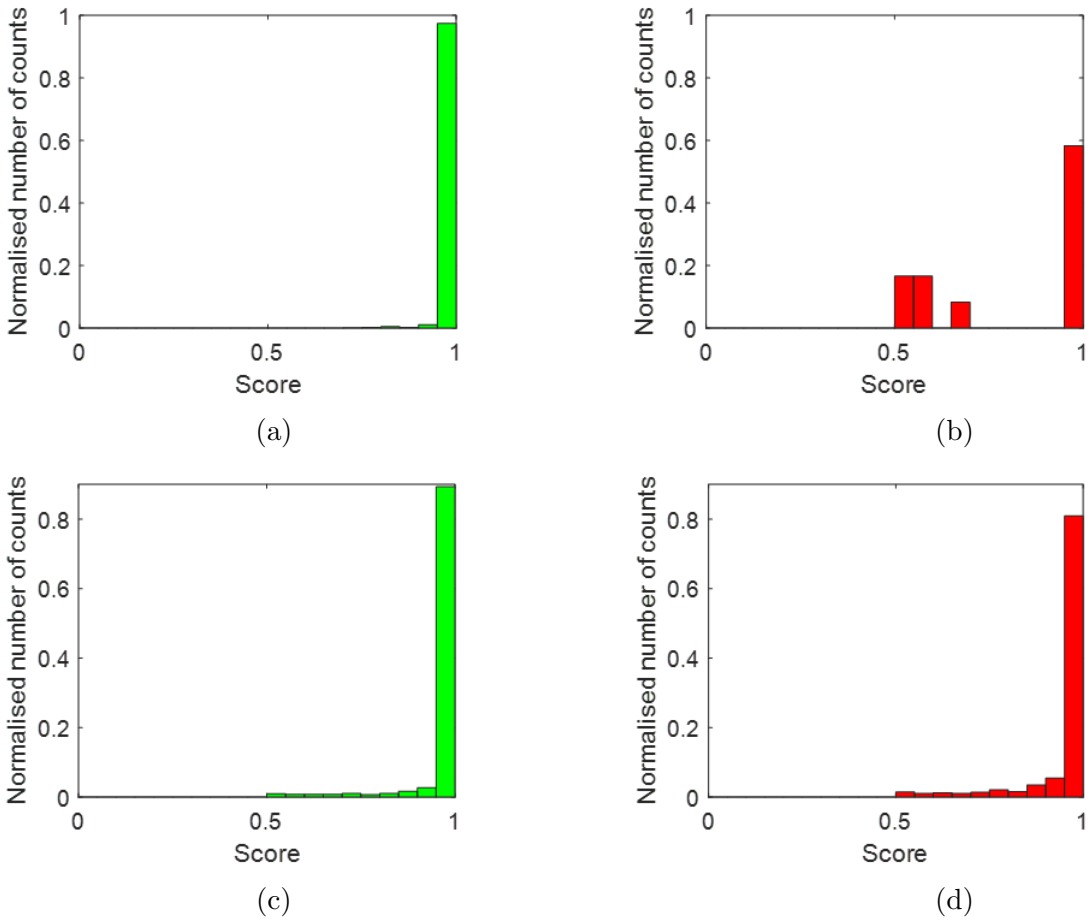


Figure 7.12: Control classifier scores a) The spectrogram is a correctly classified 'Bird' spectrogram b) The spectrogram is a 'Bird' spectrogram misclassified as 'Drone' c) The spectrogram is a correctly classified 'Drone' spectrogram d) The spectrogram is a 'Drone' spectrogram misclassified as 'Bird'.

7.4 Retrained Classifier Results

7.4.1 Training the Retrained Classifier

In order to improve the classifier's robustness to RCS and range, 60% of the drone files in data set B were randomly selected and added to the training subset of data set A, to retrain the classifier on some of the images that were more difficult to classify. The idea being that the classifier would realise that HERM lines are not visible for many drones, encouraging it to learn more robust classification features

such as variation in SBR, Doppler spread, or the rate of change in Doppler. A set of supplementary bird measurements were added also to the training subset, to balance the training classes and to provide more representation of bird features, to avoid raising the false alarm rate.

Table 7.4: Summary of data used for training the retrained classifier, across data sets A and B.

Data source	Target type	Number of files	Number of spectrograms	Maximum SBR [dB]	Minimum SBR [dB]	Mean SBR [dB]
Data set A	Bird	53	683	44.2	21.6	30.7
Data set A	I2-D	77	713	64.1	22.0	51.0
Data set A	I1-D	46	865	64.2	9.7	52.7
Data set B	Bird	357	3134	42.5	9.6	25.2
Data set B	P3-D	2	109	40.2	24.0	34.8
Data set B	P4-D	6	315	40.5	4.9	22.8
Data set B	I1-D	6	410	60.0	7.5	34.7
Data set B	I3-D	1	59	43.1	10.5	26.1
Data set B	I4-D	1	46	62.9	46.7	53.6
Data set B	RC-F	1	33	48.2	14.3	28.5
Data set B	AV-D	1	52	40.7	16.7	27.9
Data set B	DJIM200-D	2	88	40.0	9.0	28.2
Data set B	G-D	4	443	55.6	27.4	45.3
Data set B	O1-D	4	445	52.7	12.8	34.8
Data set B	S9-D	6	211	54.5	12.0	37.6
Data set B	Z1-F	1	124	41.4	10.3	31.8

Table 7.4 provides a summary of the spectrograms used for training the retrained classifier.

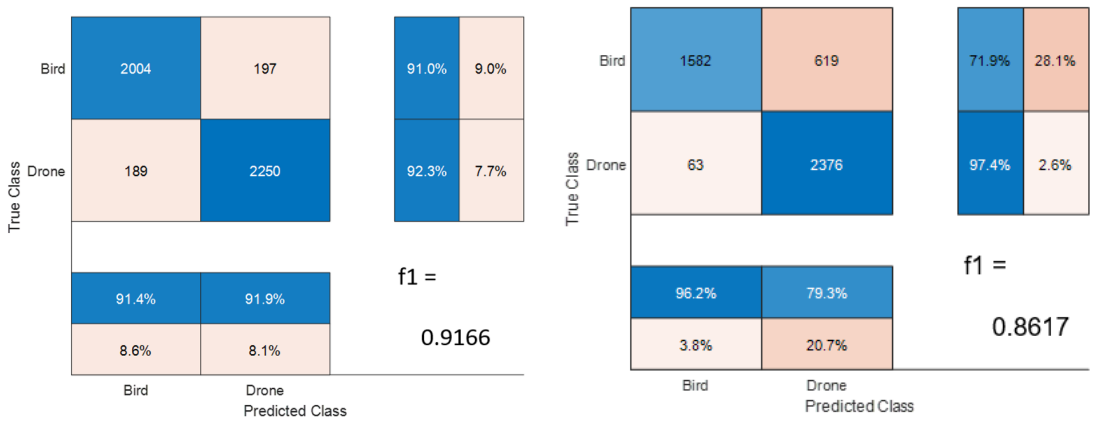
7.4.2 Testing the Retrained Classifier on Data Set A

The retrained classifier is tested on the test subset of data set A, achieving a high drone recall. The false positive rate increases from 0.7% to 5.3% compared to the control classifier, giving a lower F1 score of 0.98. The false positive rate may still

be acceptable for some applications, particularly if the number of false alarms can be reduced by fusing the result across a track.

7.4.3 Testing the Retrained Classifier on Data Set B

Fig. 7.13a shows the results of testing the retrained classifier on data set A, and data set B.



(a) (b)

Figure 7.13: Performance of the retrained classifier on data set A, and data set B a) Balanced bird and drone classes b) Unbalanced bird and drone classes.

The drone recall (92.3%) is significantly higher than for the control classifier (61.7%). 9% of bird targets are misclassified as drone, which is a significant deterioration over the 0.5% of bird targets misclassified by the control classifier. This is due to a reduced separation between the bird and drone class in the training data. The F1 score of the retrained classifier is 0.91, compared to 0.82 for the control classifier, suggesting that this classifier provides the best balance between precision and recall overall.

An additional classifier was trained without using the bird data in data set B to account for the class imbalance - in this case, only 15% of the training data belonged

to the bird class, resulting in the misclassification of 28% of bird spectrograms (Fig. 7.13b). This is unacceptably high, especially when one considers how the number of bird tracks can be expected to vastly outnumber the drone tracks under most operational environments. In addition to balancing the classes, training on additional bird data provided more coverage of bird tracks. This likely allowed the classifier to extract more robust classification features, facilitating discrimination between drone and confuser classes.

Performance of the Retrained Classifier as a Function of Drone Size

On the whole, drone recall is significantly improved for each model of drone in data set B. The most significant improvement is for the P3-D model, where 97.7% of spectrograms were correctly classified, despite the absence of blade returns. Table 7.5 allows a direct comparison between the control and retrained classifier performances upon each group of drones. The retrained classifier vastly outperforms the control classifier, classifying 61.8% of the small drone spectrograms compared to just 2.7%. Similarly, the retrained classifier achieves good performance on the medium and large drones, achieving accuracies of 92.2% and 94.4%, respectively.

Table 7.5: Performance comparison of the control and retrained classifiers, grouped by size of drone.

Drone group	Drone recall [%]		
	Control classifier	Retrained classifier	Difference
Small	2.7	61.8	59.1
Medium	59.3	92.2	32.9
Large	44.9	94.4	49.5

Performance of the Retrained Classifier as a Function of SBR

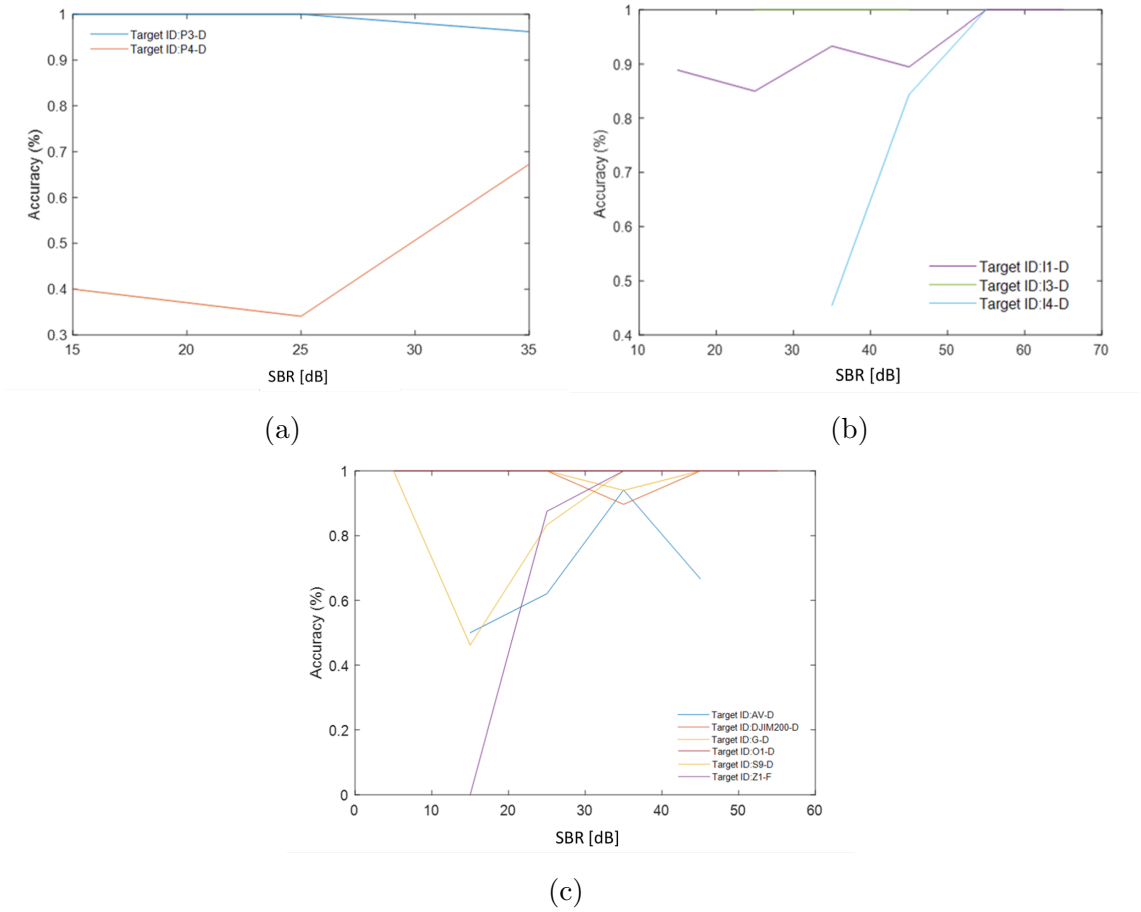


Figure 7.14: Retrained classifier performance as a function of signal to background ratio (SBR) for the a) Small drones in data set B b) Medium drones in data set B c) Large drones in data set B.

Performance is shown as a function of SBR in Fig. 7.14a to Fig. 7.14c. Fig. 7.14a shows the results for the two small drone models where, as expected, performance predominantly increases with SBR. Fig. 7.14b displays accuracy as a function of SBR for the medium sized drones, where again performance increases with increasing SNR for the case of the I1-D and I4-D. Fig. 7.14c displays accuracy as a function of range for the large sized drones. High accuracies can be achieved with an SBR above 30 dB for all but the AV-D, whilst for the control classifier a SBR of 50 dB was required for the accuracy to exceed 90%.

Generally, SBR has less of an impact on classification performance for this classifier compared to the control classifier, as it is not relying on the presence or absence of HERM lines, which is a feature that is heavily dependent on SBR. Similar results were shown in Section 5.5, where it was demonstrated that retraining on a subset of lower SBR improved classifier robustness and reduced the dependence of performance on SBR.

Scores Output by the Retrained Classifier

The scores declared by the retrained CNN are presented in Fig. 7.15a for the case where the true class is bird and the spectrograms are correctly classified. Fig. 7.15b presents the scores when the 'Bird' spectrograms are misclassified as 'Drone'.

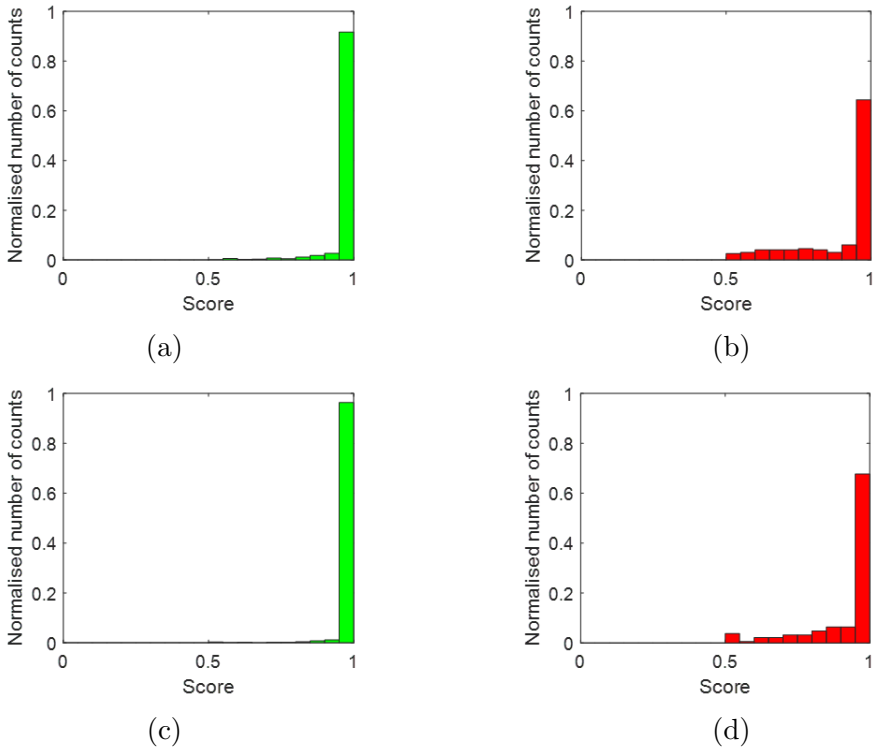
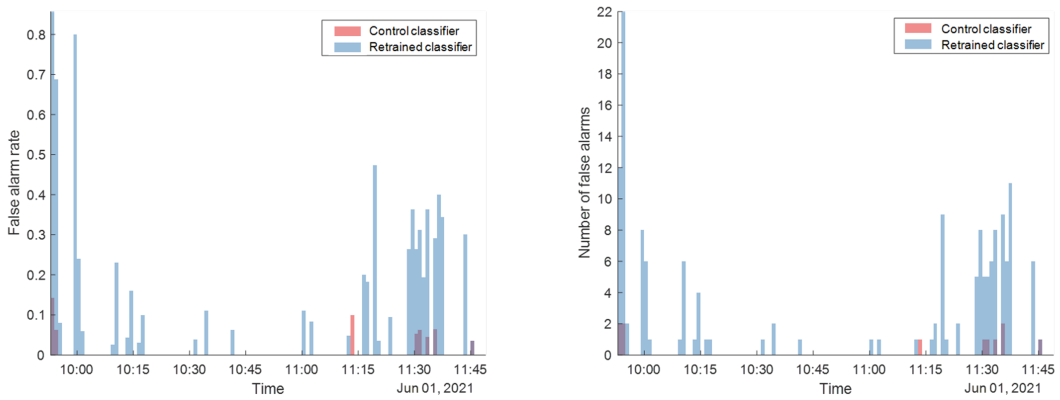


Figure 7.15: Retrained classifier scores a) The spectrogram is a correctly classified 'Bird' spectrogram b) The spectrogram is a 'Bird' spectrogram misclassified as 'Drone' c) The spectrogram is a correctly classified 'Drone' spectrogram d) The spectrogram is a 'Drone' spectrogram misclassified as 'Bird'.

Fig. 7.15c-d show similar for the case where the true class is drone. Similarly to the control classifier, the results demonstrate that the classifier is generally very confident in its decision, even when it is misclassifying a target, with the classifier reporting a score of over 0.95 for 68% of the birds falsely declared as drones, and for 64% of the drones falsely declared as birds.

7.4.4 Evaluating the False Alarm Rate of the CNN Classifiers

The time between false target reports is a vital metric for many applications, and is not reported in the confusion matrices. The false alarm rate is therefore quantified in order to assess the implications for operational effectiveness.



(a) (b)

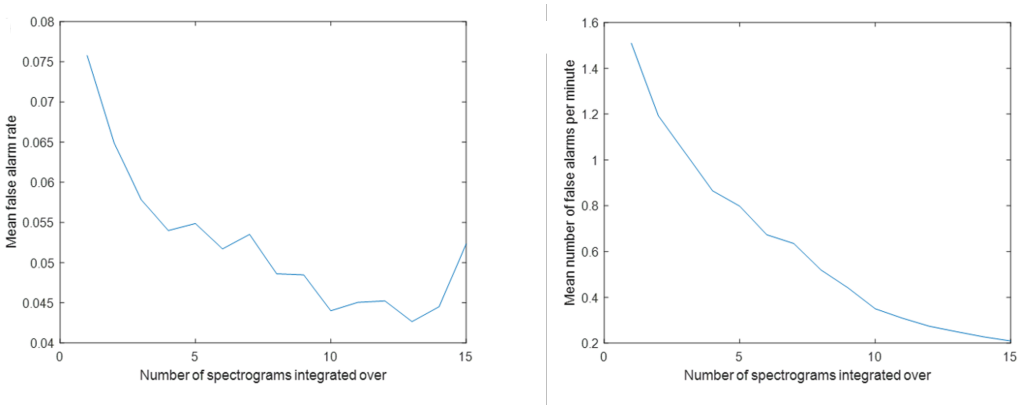
Figure 7.16: Histogram showing the a) False alarm rate and b) The number of bird spectrograms misclassified per minute by the control classifier (red) and retrained classifier (blue).

The opportune bird test data (Section 7.2.2) was used to quantify the false alarm rate per minute of both the control and retrained classifiers. Each spectrogram was assigned to a time bin, corresponding to the timestamp of the final frame in the spectrogram. The classifiers were then tested on the spectrograms in each minute time bin, and the mean false alarm rate per minute (Fig. 7.16a) and the mean number

of false alarms per minute (Fig. 7.16b) were calculated. It is important to note that the tracks for testing have already been subjected to some filtering to ensure reliable ground truth labels, and therefore these results do not form a complete description of the expected false alarm rate across the entire radar field of view.

The average false alarm rate per minute of the control classifier is 0.5%, corresponding to 0.11 false alarms per minute for the tracks extracted from this particular trial. The average false alarm rate of the retrained classifier is 8%, resulting in a mean of 1.51 false alarms per minute. This may be slightly high for some applications, however, recall that a single false alarm is defined as a misclassification of a single spectrogram, therefore the false alarm rate can be reduced by averaging the classification label over multiple spectrograms.

The score was averaged over N successive spectrograms, and the output label was declared as bird if the mean score over N spectrograms was above 0.5, else the target was labelled as drone. For the first N-1 spectrograms, the score is averaged over all spectrograms belonging to the track. As demonstrated in Fig. 7.17, integrating over multiple spectrograms leads to a significant decrease in the mean false alarm rate. Extending N beyond 13 frames leads to a rise in the false alarm rate (although the number of false alarms continues to fall), as when N exceeds the number of spectrograms in the track there are fewer outputs within a particular histogram time bin, increasing the false alarm rate if the number of outputs decreases faster than the number of false alarms decreases. The shortest track in the test data set consists of 10 spectrograms. Integrating over 10 spectrograms results in a mean false alarm rate of 4%, and an average of 0.35 false alarms per minute. This corresponds to a false alarm every three minutes, which is reasonable. No overlap was used between successive spectrograms, therefore integrating over 10 spectrograms results in a latency of approximately 55 seconds for the reported level of performance.



(a) (b)

Figure 7.17: a) The mean false alarm rate of the retrained classifier, as a function of the number of spectrograms the output is integrated over b) The mean number of false alarms per minute of the retrained classifier, as a function of the number of spectrograms the output is integrated over.

7.5 Fused Classifier Approach to Reducing the False Alarm Rate

A fusion approach was developed with the aim of combining the high drone recall of the retrained classifier with the low false alarm rate of the control classifier. Ideally, the control classifier would be used to classify birds and high SBR drones that exhibit clear HERM lines, whilst the retrained classifier would be used for the classification of smaller and longer-range drones. Hypothetically, if one knew the truth label, one could choose which of the two classifiers to use, depending on which maximised the score in the true class. This is an idealised case of dynamic classifier selection [191][188].

The problem can be re-framed as a classification problem, where the task is to assign a test image to one of two categories: the collection of spectrograms that are classified with a greater score in the correct class using the control classifier, and the

collection of spectrograms that are classified with a greater score by the retrained classifier. As demonstrated in Section 7.3.3, the effectiveness of the control classifier is a function of range - there is a boundary beyond which the performance of the classifier drops, for a particular model of drone. Take the simplified case where performance is a function of range only - the task of the classifier would be to draw the boundary in range, whereby the retrained classifier is used for longer range spectrograms, whilst the control classifier is used for shorter range spectrograms to take advantage of the low false alarm rate. In reality, performance is not just a function of range, but also of size and model of drone, the local background level, and trajectory, to name just a few factors. The problem becomes multi-dimensional with boundaries that are not clearly defined, therefore a simple handcrafted boundary is not enough, and it makes sense to look to machine learning for a solution.

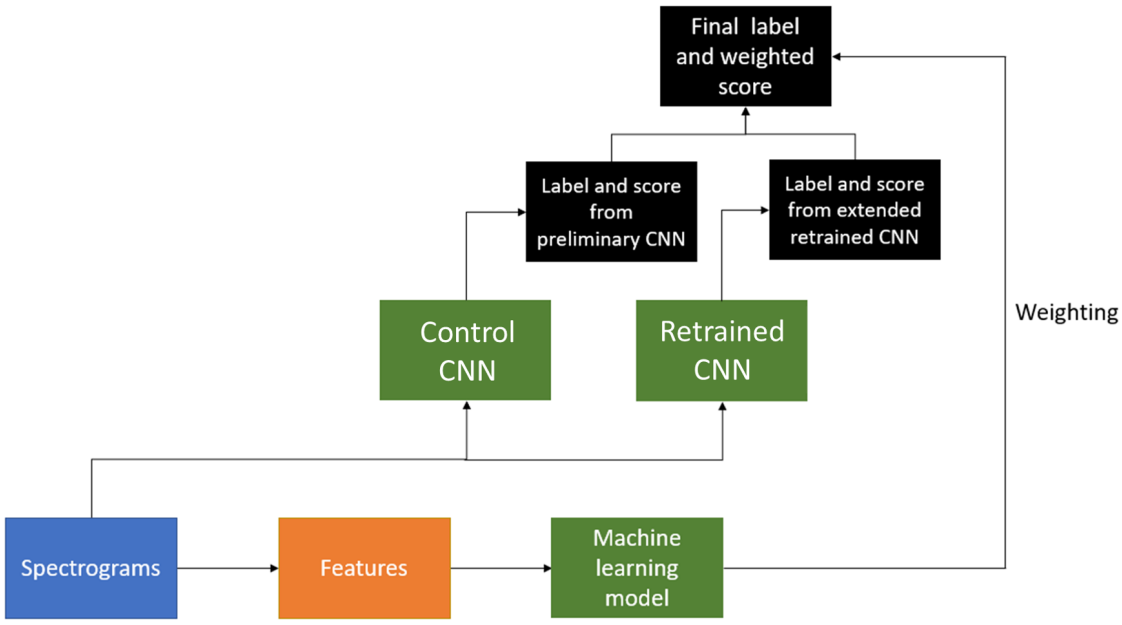


Figure 7.18: A flowchart outlining the fused classifier approach, where a machine learning model is used to form a weighted output from two convolutional neural network (CNN) classifiers.

The proposed approach is to test both the control and retrained CNNs on a particular spectrogram and to calculate a weighted sum of the scores declared by

both classifiers. If the score in the 'bird' class exceeds 0.5, the spectrogram is declared as a bird, else the spectrogram is declared as a drone. A machine learning model is used to learn the appropriate weightings depending on the features present in the test data. It is trained on a set of spectral features to output the optimal classifier and a score, which is then used to weight the scores of the two CNNs and declare the target class. Fig. 7.18 outlines the fusion approach to bird-vs-drone classification.

7.5.1 Fused Classifier Methodology

Six simple features are extracted from each spectrogram (average range, average body SBR, average radial velocity, standard deviation of the range, standard deviation of the body SBR, and standard deviation of the radial velocity). These 6 features form the input to a machine learning algorithm. Two algorithms were chosen for investigation: a logistic regression model, and a decision tree.

Truth labels (either 'control' or 'retrained') were taken to be the classifier that correctly classifies spectrograms with the highest score or, in the case where the spectrogram is misclassified by both CNNs, misclassifies the data with the lowest score. The data used for training the machine learning model therefore can not have been seen by either of the CNNs during training. Routinely, 60% of the data set is used for training the CNN. This would leave the remaining 40% of the data set to be split between training the machine learning model and testing the overall classification system. If a 60/40 split is chosen for training the machine learning model, then only 16% of the full data set can be used for evaluating performance (40% of 40%). It was decided that this would not give a sufficiently representative test data set, therefore, an alternative training approach was taken to maximise the amount of data available for testing the model. Data set A was split into two halves for training. One half (henceforth referred to as subset A1) was used to train one

control classifier, whilst the other half (henceforth referred to as subset A2) was used to train a second control classifier. Similarly, data set B (Section 7.2.2), was also split into two halves (subsets B1 and B2), and used to train two retrained classifiers. Each of the four classifiers were then tested on the subsets that were unseen during training, as shown in Table 7.6. In this way, the full data set can then be used for training and testing the machine learning model. 60% of this feature matrix was then used for training a machine learning algorithm, and 40% was used for testing the entire model.

Table 7.6: The subsets used for training and testing four convolutional neural networks for a fusion model.

Classifier	Training data	Testing data
Control	A1	A2+B2
Control	A2	A1+B1
Retrained	B1	A2+B2
Retrained	B2	A1+B1

Fig. 7.19a shows the result of testing the two control classifiers on the full data set. The control classifiers achieve excellent performance on the bird data set, but poor generalisation to the drone class, correctly classifying only 60.7% of drone spectrograms. Fig. 7.19b shows similar for the retrained classifier, where the drone recall is very high, at the expense of a raised false alarm rate relative to the control classifier.

Fig. 7.19c shows the hypothetical obtainable performance on the Aveillant data set if the 'optimal' classifier was always used for testing, where prior knowledge of the target's true identity is used to choose the optimal classifier. In this case, it is possible to correctly classify all of the bird spectrograms and 97.3% of the drone spectrograms, simply by making an intelligent choice between the two CNNs, setting a theoretical upper bound of 0.95 on the F1 score. The aim of this work

was to achieve an F1 score comparable to this upper limit, whilst using a machine learning model to predict the optimal CNN classifier without any prior knowledge of the class label.

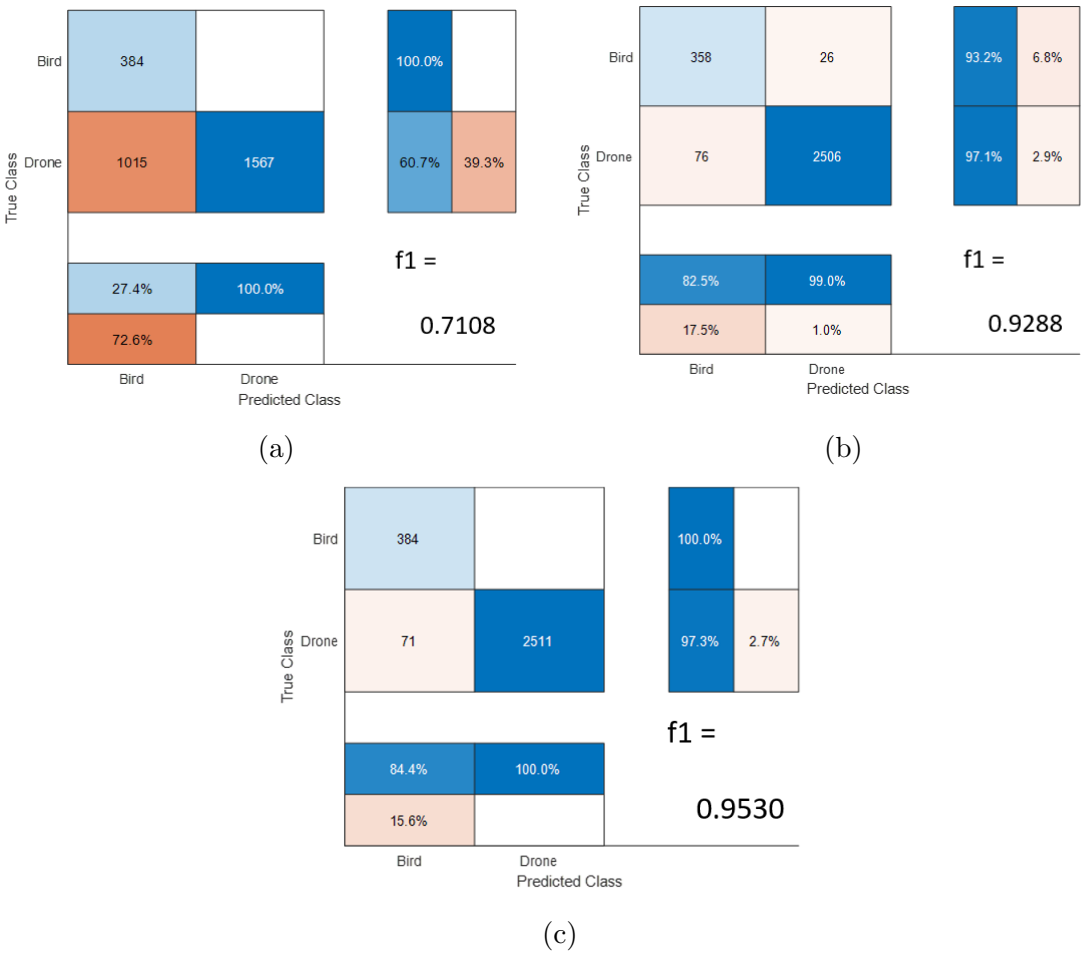


Figure 7.19: Performance of the classifiers used in a fused classification scheme a) Control classifier b) Retrained classifier c) Theoretical optimal classifier.

Fig. 7.20 is a plot showing the optimal CNN classifier for the entire data set, as a function of SBR and standard deviation on SBR across 20 frames of data. The control classifier results in the best performance for the higher SBR data whilst the retrained classifier is preferable for lower SBR data. This is to be expected, since it was demonstrated in Section 7.3.3 that the performance of the control classifier is heavily impacted by SBR. Note however that there is not a clear threshold above which the control classifier is more suitable, therefore it is essential to consider

multiple dimensions and turn to a machine learning algorithm rather than a small number of handcrafted features.

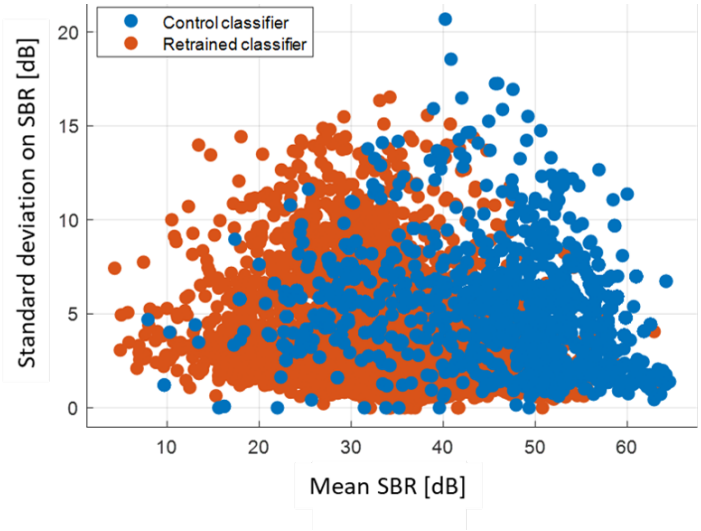


Figure 7.20: A plot showing the optimal convolutional neural network (CNN) classifier, as a function of mean signal to background ratio (SBR) and SBR variance across a spectrogram.

7.5.2 Decision Tree for Classifier Fusion

A decision tree was trained to make the choice of CNN classifier, using the features described in Section 7.5.1. A standard CART algorithm was used to select the predictor at each node, using Gini's diversity index as the split criterion [162]. The maximum tree depth was chosen to be 75, in order to prevent overfitting the model on the training data.

Fig. 7.21 shows the results of testing the decision tree classifier on the test subset. The decision tree correctly identifies 85.2% of the test samples that are classified more confidently by the control classifier, and 84.7% of the test samples that are classified more confidently by the retrained classifier, with an overall F1 score of 0.85.

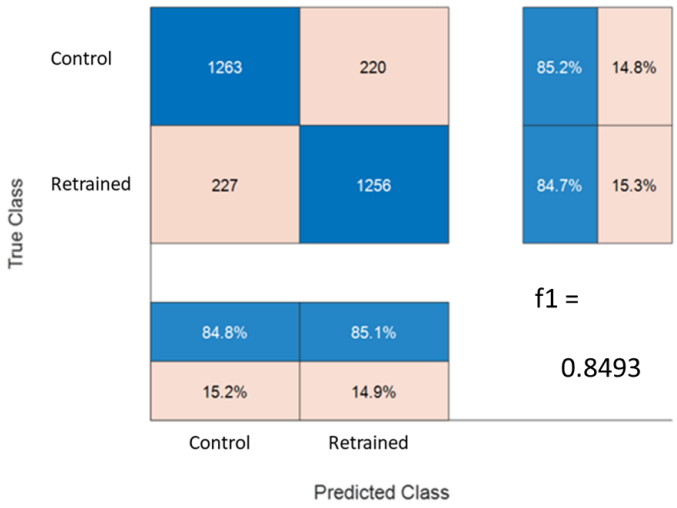


Figure 7.21: Performance of the decision tree used for classifier fusion.

Fig. 7.22 shows the estimates of predictor importance, where the most significant classification feature is shown to be the mean SBR across the spectrogram, followed by the standard deviation in SBR.

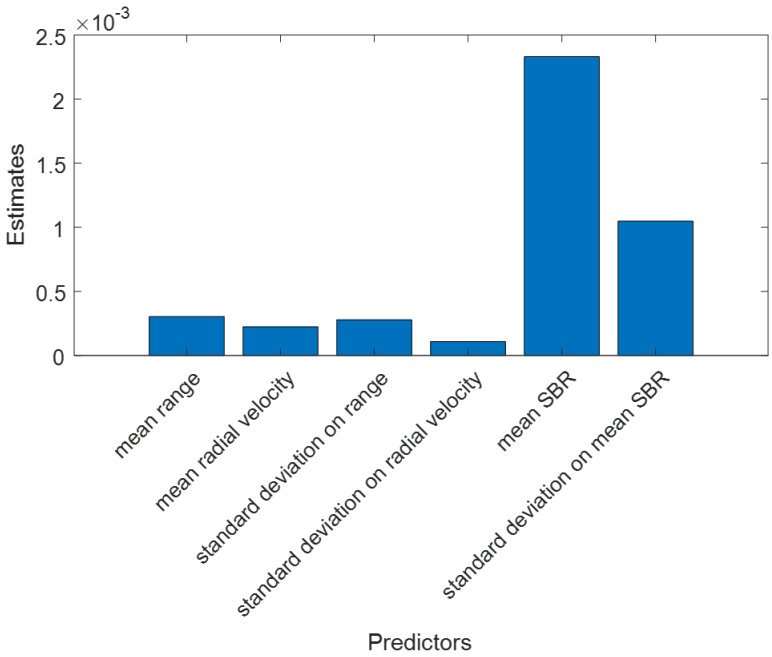


Figure 7.22: Estimates of predictor importance for the decision tree used in the fused classifier scheme.

Fig. 7.23 shows a histogram of the weightings on the control classifier, using the

scores output by the decision tree. For a particular test spectrogram, the weighting on the control classifier and the weighting on the retrained classifier sum to 1. Generally, the decision tree appears to favour the control classifier, although the retrained classifier is weighted very strongly in almost 20% of cases.

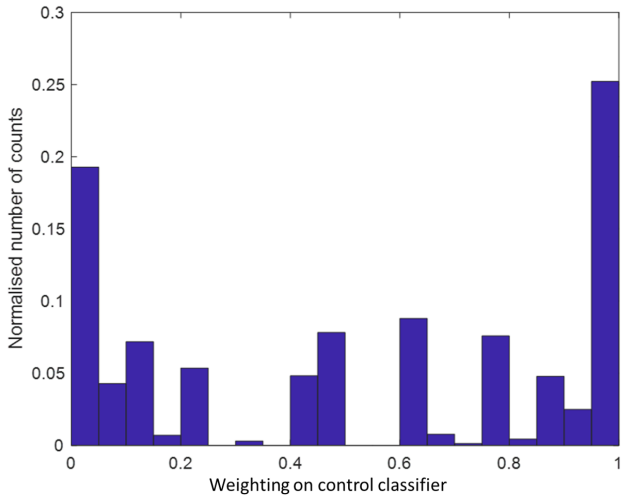
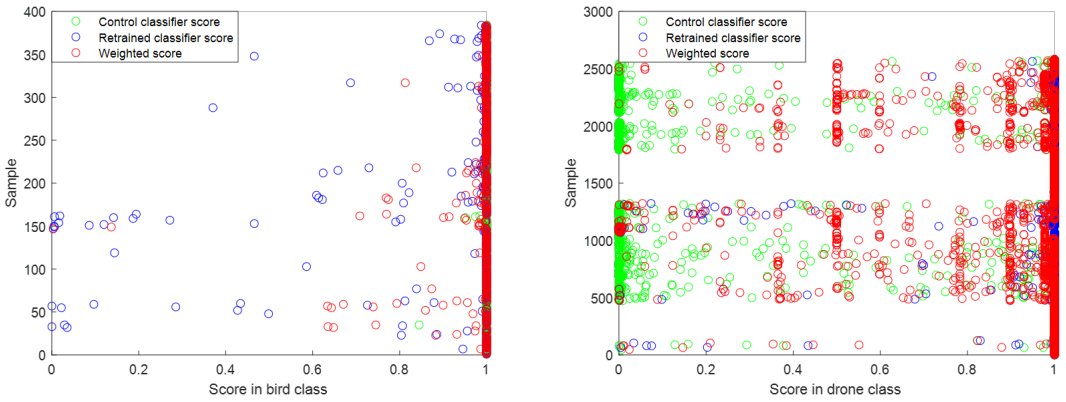


Figure 7.23: A histogram of the weightings on the output of the control classifier, set by a decision tree model.

Fig. 7.24 demonstrates the effect of the weighting upon the bird-vs-drone scores. Fig. 7.24a shows the results for the bird spectrograms, where it is demonstrated that the score in the bird class is generally raised by the control classifier, resulting in a correct classification and lowering the false alarm rate compared to the retrained classifier. Fig. 7.24b shows the results for the drone spectrograms, where it is shown that generally the score in the drone class is raised by the retrained classifier, resulting in correct classifications and raising the true positive rate. Misclassifications still occur when the control classifier is very confident that the test target is a bird, particularly since the control classifier is generally weighted more strongly than the retrained classifier, as demonstrated in Fig. 7.22. Note that there are many examples where the weighted score is very close to 50%, resulting in misclassifications that could perhaps be eliminated by fine tuning the model, or by adjusting lowering the threshold above which 'drone' is declared, although this will of course have an

impact upon the false alarm rate.



(a) (b)

Figure 7.24: The scores output by the control classifier (green), retrained classifier (blue), and the overall weighted score (red) where the weighting was set by a decision tree model. The model was tested on a) Bird spectrograms b) Drone spectrograms.

Fig. 7.25 displays the results of testing, where the output of the decision tree algorithm is used to produce a weighted fusion between the control and retrained classifiers. The algorithm misclassifies only 0.5% of bird spectrograms, comparable to the false alarm rate of the control classifier, whilst 92.4% of the drone spectrograms are correctly classified – compared to just under 60.7% for the control classifier.

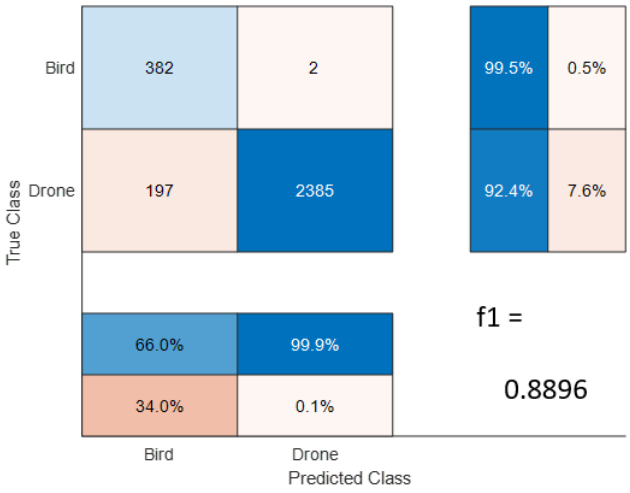


Figure 7.25: Performance of the decision tree enabled bird-vs-drone convolutional neural network (CNN) classifier, where a fusion approach is used.

It is therefore demonstrated that this approach can be used to combine the high performance of the retrained classifier with the low false alarm rate of the control classifier, resulting in a more robust classifier. One drawback with this approach is that the score of the CNN classifier has been demonstrated to be an unreliable measure of confidence, as the classifier generally declares labels with a high score, regardless of whether it is correct. Therefore an alternative approach was also considered, using the label output by the decision tree to perform dynamic classifier selection.

7.5.3 Dynamic Classifier Selection

Fig. 7.26 displays the results of the bird-vs-drone algorithm, following the passing of the spectrograms through the CNN declared by the decision tree.

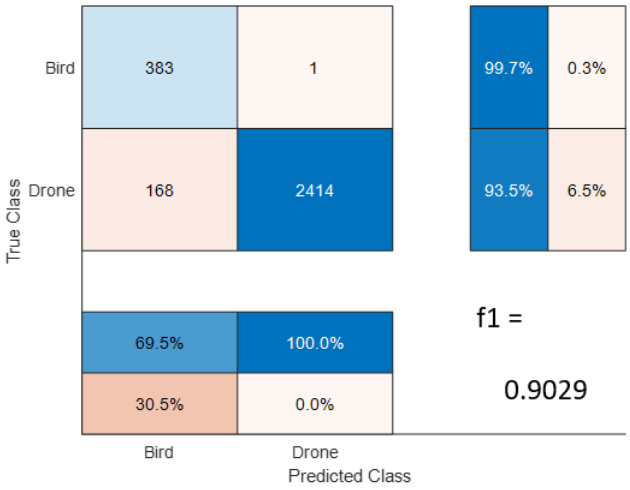


Figure 7.26: Performance of the decision tree enabled bird-vs-drone convolutional neural network (CNN) classifier, where a dynamic classifier selection method is used.

In this case, only 0.3% of the bird spectrograms are misclassified as drone – comparable to the control classifier - whilst 93.5% of the drones are correctly classified, compared to just under 60.7% for the control classifier. This method slightly out-

performs the fusion approach, with an overall F1 score of 0.90, but the improvement in performance is not significant.

For comparison, a logistic regression model was also considered. The model was trained using the MATLAB function *fitclinear* with a logistic solver [192]. Ridge (L2) regularisation was used, with strength equal to $1/n$, where n is the training sample size. Stochastic gradient descent was used to minimise the objective function. Table 7.7 compares the results of using a logistic regression classifier and a decision tree classifier for both fusion and dynamic classifier selection. The decision tree classifier outperforms the logistic regression classifier, although all four results demonstrate that a fusion approach could be used between the control and retrained classifiers, resulting in a robust drone classification algorithm whilst minimising the false positive rate. A decision tree network with dynamic classifier selection is the most appropriate model for this purpose out of those considered.

Table 7.7: Summarised results of the fused classification scheme approaches considered.

Classifier	Approach	F1 score
Decision tree	Fusion	0.89
Decision tree	Dynamic classifier selection	0.90
Logistic regression	Fusion	0.85
Logistic regression	Dynamic classifier selection	0.80

7.6 Drone-vs-Drone Classification

Following a thorough analysis of the bird-vs-drone classification problem, the problem of drone-vs-drone classification was considered. Rather than attempting model-level classification, performance was established using four classes. Drones were grouped based on size, in order to establish whether drone size could be inferred

from the spectrogram as this method was shown to result in a high performing classifier in Chapter 6. This could be useful for assessing the risk of drone-aircraft collisions, since larger drones are capable of causing more damage in a collision, and for defence purposes since larger drones could carry heavier payloads, for example chemical agents or weapons.

A classifier (based on the architecture of AlexNet, using the same spectrogram processing parameters as in Section 7.2.1) was trained both to perform bird-drone classification and to estimate the drone size, where the drone size groupings are given in Table 7.2. 60% of the bird, small drone, medium drone, and large drone files were used for training, 10% for validation, and 30% for testing.

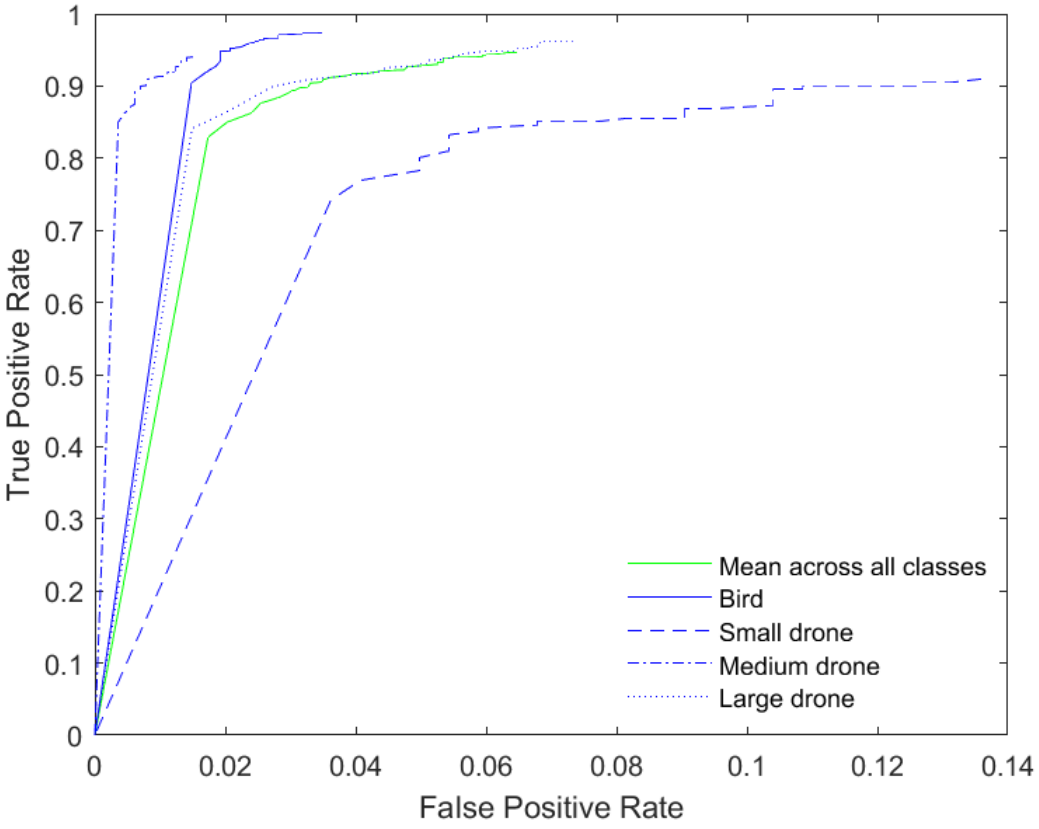
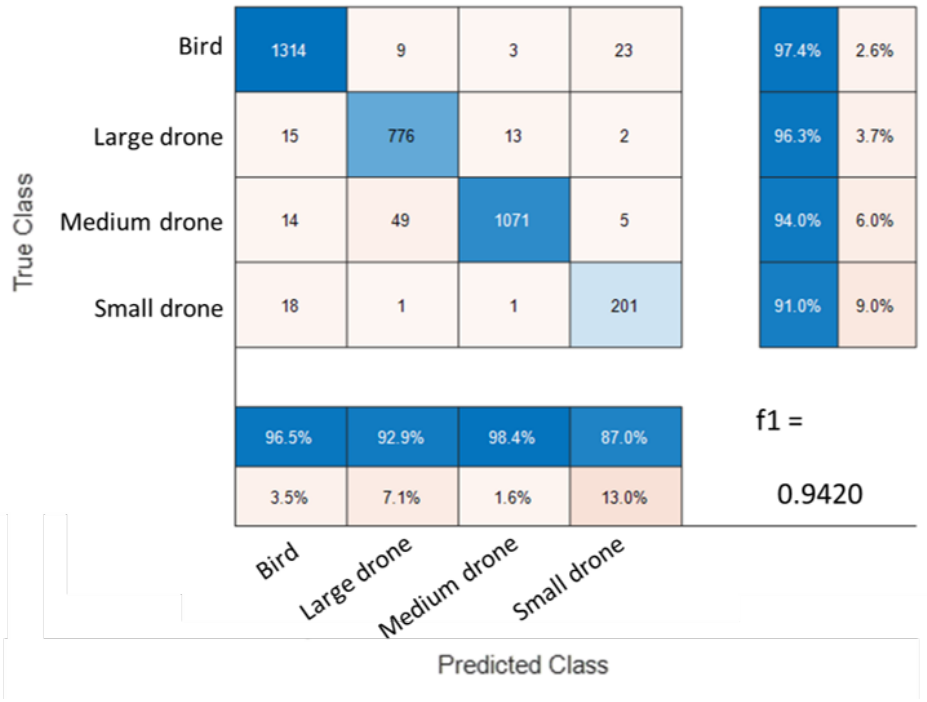


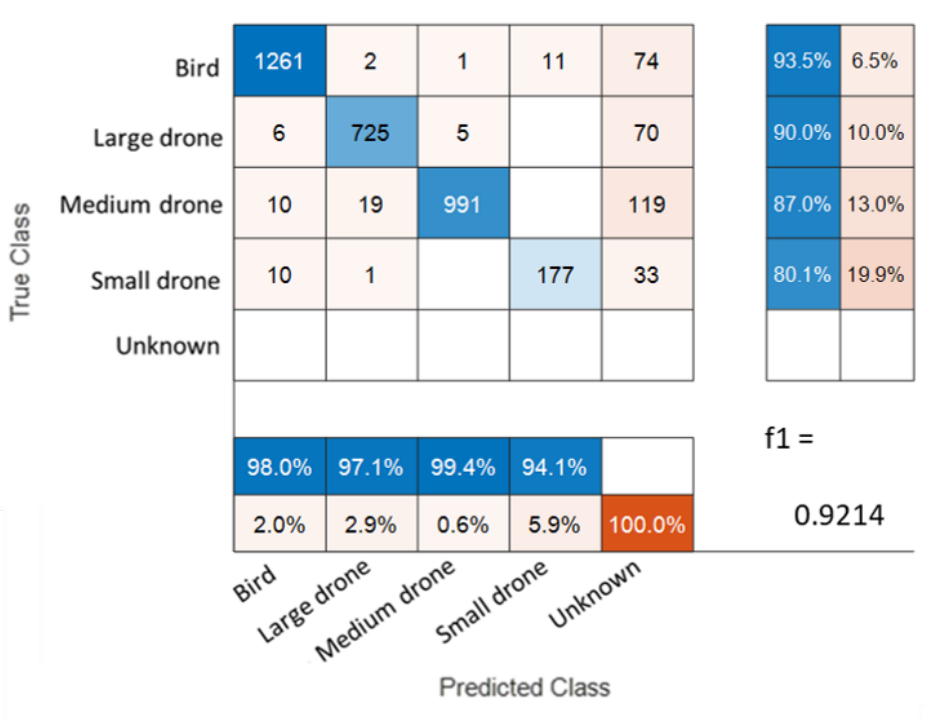
Figure 7.27: Receiver operating characteristic (ROC) curve for a drone size classifier, showing true positive rate vs false positive rate.

Since the problem is no longer binary, an unknown class was introduced by enforcing a threshold on the classification scores whereby if the score is below the threshold, the classifier declares the class as 'Unknown'. Fig. 7.27 is a ROC curve showing the true positive and false positive rates, as the threshold ranges from 0 to 1. The small drone class has the largest false positive rate, meaning that these false declarations can be removed by including an unknown class without significantly affecting the true positive rate. On the other hand, it is clear that introducing a threshold may have a significant effect on the true positive rate for the medium drone class. A threshold of 0.96 was selected, by choosing the threshold that corresponds to the elbow of the curve for the mean true and false positive rates across all four classes.

Fig. 7.28a shows the results of testing the size-based classifier without an unknown class (F1 score of 0.94). 96% of the large drones are identified as such, and 94% of the medium drones are also correctly classified, with the majority of misclassifications being 'large drone'. 91% of the small drones were correctly classified, however, 9% were misclassified as birds. Fig. 7.28b shows the confusion matrix where 'Unknown' is declared for scores below 0.96 (F1 score of 0.92). When an unknown class is used, many of the misclassifications between classes are eliminated, at the expense of a small percentage of targets now being undeclared. There is an improvement in the precision for each class, at the expense of the recall.



(a)



(b)

Figure 7.28: Drone size based classifier results a) 4 classes b) Including an unknown class.

Of the targets classified as small drone, 94% of these actually are small drones, compared to 87% without the unknown class. The inclusion of an unknown class has made the classifier more reliable since it is no longer being forced to decide on a label. The targets that are declared as unknowns ought to be monitored until it is possible to classify them with confidence.

Note that one cannot apply a direct comparison between the results of this classifier and the results of the bird-vs-drone classifier, since the particular samples in the training and test subsets were not the same in both cases in order to maintain balance in the training classes. However, combining the drone classes into a single class results in good performance for the bird-vs-drone problem, that is at least comparable to the performance of the binary classifier.

This work has raised questions about why the classifier is so confident, even when it is wrong. Although the softmax maps the output onto (0,1), and softmax scores have been shown to perform reasonably well when identifying out-of-distribution samples [193][194], CNN scores can not reliably be interpreted as a measurement of model confidence. The existing domain may be orders of magnitude larger than the region occupied by training samples, but the network extrapolates with 'unjustified high confidence' for points far from the training data [195][196]. Secondly, the softmax layer is sensitive to perturbations, meaning that CNNs can be easily fooled by imperceptible changes to the input image resulting in high scores in an incorrect class [197]. In [190], the authors demonstrate the importance of calibration for treating CNN scores as confidence measures, outlining a scaling method that prevents overestimation of confidence. Another alternative is to consider a replacement to the softmax function for the declaration of classifier confidence. In [148], an auto-encoder is trained adversarially to identify anomalous input spectrograms that fall outside of the expected distribution. In [198], a Competitive Over-complete Output Layer (COOL) is used to train a more robust network for the classification of

SAR images, resulting in a set of confidences with more variation than the softmax output. Finally, alternative network architectures such as Bayesian neural networks could be considered [195][196].

7.7 Chapter Conclusions and Future Work

To summarise, a multi-frame CNN classification algorithm has been developed, using training and test data from an L-band staring radar. Classifier performance has been evaluated under increasingly demanding conditions [41], with a range of environments, models of drone, and target ranges represented in the test data set.

7.7.1 Conclusion

In this chapter, a multi-frame CNN has been trained and tested on two Aveillant data sets. Data set A consisted of only two (similar) models of drone, at intermediate ranges (<2.5 km). Most of the drone signatures within this data set had clearly visible HERM lines, making the drone spectrograms easily distinguishable from bird spectrograms to the human eye. Data set B was significantly more challenging from a classification perspective consisting of a wide variety of drone models (including small drones), a wide variety of clutter environments, and extensive coverage in range. The precise results outlined in this work are likely to be data set dependent, but it is expected that the following findings will hold true.

- **Classification performance is dependent on range and on the local SBR.**
  A control CNN was trained on a high SBR data set and tested on both a similar data set (Data set A, F1 score of 0.997) and a more challenging data set (Data set B, F1 score of 0.826). Poor performance was observed when the micro-Doppler returns were not visible above the noise floor or clutter,

indicating that they are a key feature for classification. This validates the work presented in Section 5.5.

- **Bird-drone classification performance is a function of drone model.**

  It is demonstrated that CNNs can separate birds from larger drones more accurately (94%) than from smaller models of drone (62%).

- **High false alarm rates were shown to be exacerbated by imbalanced training data.**

  Opportune bird data collected at The University of Birmingham was used for balancing the data sets to prevent bias towards the drone class.

- **When training a classification algorithm, it is vital that the training data is sufficiently representative of the data that the classifier will be exposed to during deployment.**

  A retrained classifier was developed, where training was conducted on samples from both data set A and data set B. The resulting classifier achieved excellent performance on the drone class with an overall F1 score of 0.917.

- **Using a decision tree classifier for dynamic classifier selection resulted in a robust drone-vs-bird discriminator.**

  Two bird-vs-drone classifiers were trained in this work, both with the same CNN architecture, but with differing amounts of representation within the training data. Ultimately, a single classifier was not optimal for all situations, therefore classifier fusion between two CNNs was explored. It was demonstrated that using either a decision tree classifier or a logistic regression model for dynamic classifier selection resulted in a robust drone-vs-bird discriminator.

- **CNNs are suitable for classification of UAV size at L-band.**

  A CNN correctly classified 96.3% of large drones, 94% of medium drones, and 91% of small drones, whilst also correctly classifying 97.4% of birds. Inferring

drone size will better inform risk assessments associated with potential collisions with manned aircraft, and allow an operator to assess whether the drone has the potential to be carrying a heavy payload such as a weapon.

- **CNNs may report high confidence even for misclassified spectrograms.** The application of a threshold on the classifier score is not a sufficiently robust method of declaring unknowns as this leads to an overestimation of confidence.

7.7.2 Recommendations

Based on the results of this work, recommendations are to:

- Ensure that the training data is sufficiently representative of the data that the classifier will be faced with operationally. This could involve simulation, augmentation (e.g. injecting Gaussian noise in to spectrograms) and collecting data from the site at which the classifier will be deployed.
- Produce a clutter map before deploying the model at a new location, in order to better understand the potential limitations of the classifier, since it has been demonstrated that the model is sensitive to clutter.
- Restrict the input to the classifier where possible, using filters on the height and threshold of the tracks, in order to reduce the number of false declarations due to confuser targets.

7.7.3 Future Work

A number of avenues for future research have been identified, across several key areas: the approach to training, testing, and pre-processing.

- **More consideration needs to be given to the treatment of targets outside of the expected domain (drone and bird), and the declaration of 'Unknown' targets.**

  Alternative approaches may be to include an 'other' class during training, (although it may be difficult to populate this class with sufficiently varied examples), to train a large number of networks and calculate a confidence based on the distribution of outputs, or to consider alternatives to the softmax classification layer.

- **The use of simulated drone data [84] for training would allow additional models of drone and more complicated flight scenarios to be represented.**

  More work is needed to develop a realistic drone simulation model, considering how to model the body return and the blades (as point or line scatterers) as well as other effects such as SBR variation with increasing range and changing aspect angle. To prevent overfitting on simulated noise, it may be beneficial to inject the simulated targets into real radar backgrounds.

- **A thorough understanding of the performance metrics, particularly the false alarm rate, is required prior to classifier deployment.**

  Operationally, one expects birds to vastly outnumber drones, therefore a rigorous measurement of the false alarm rate is needed to assess whether an algorithm is suitable for a particular application. In addition to this, it is important to consider how the false alarm rate varies according to the bird population – for example, is there a heightened false alarm rate along migratory pathways at certain times of the year, and does this impact the effectiveness of the algorithm?

- **A comparison of classifier performance across frequency bands.**

  In Chapter 6, drone model classification was performed using data collected with a Ku-band staring radar operating at very short range, whereas the data

in this chapter was collected at longer ranges using an L-band radar. A comparison of performance in the two regimes would be interesting but lies beyond the scope of this work since there are few shared platforms across the two data sets.

- **Fusion of multiple radar systems into a single surveillance system.**

  The L-band radar could be used alongside a long-dwell classifier for long range drone classification, whilst the Ku-band radar could be used alongside a CNN that takes single frame inputs for short-range classification.

- **An investigation of performance using different processing parameters.**

  Since a key classification feature is a Doppler-based signature, better classification performance may be achieved using finer Doppler resolution (therefore a higher degree of micro-Doppler detail and better separation between the target and stationary and slow-moving clutter). However, fast drone manoeuvres conducted within the coherent integration time may cause energy to spread along multiple Doppler bins, reducing SBR.

- **Investigation into system noise characteristics and the influence of noise upon classification performance.**

  As high SBR is required for accurate classification, system noise (such as phase noise) may affect performance. Methods of reducing noise, such as matched filtering, could be explored.

Chapter 8

Conclusion and Future Work

8.1 Summary

The increase in drone use for commercial and recreational purposes and the subsequent potential for misuse necessitates non-cooperative UAV surveillance methods. Radar is a promising solution, allowing round-the-clock surveillance, unaffected by light levels or weather conditions. However, drones pose numerous detection challenges due to their low RCS, low flight altitudes, and relatively low velocities, requiring high sensitivity for detection. Birds occupy airspace in a similar way to drones, and have a similar size and RCS, therefore presenting as false alarms and hindering the effectiveness of drone surveillance systems. Therefore, the ability to distinguish between drones and non-drones is crucial for the development of an effective C-UAS system. Further, the ability to classify drone type is useful for security purposes and forming risk assessments. In this thesis, CNNs have been used to classify drones from non-drones (primarily birds), and to perform drone model-level identification. CNNs were chosen as they extract complex non-linear features directly from the

input data, and are high-performing, efficient, and invariant to changes in position.

Experimental data collected using an L-band staring radar were used to demonstrate key differences between bird and drone signals, particularly the presence of characteristic HERM lines in drone spectrograms. This thesis began by proving CNNs to be suitable for discriminating between drone and bird spectrograms, establishing baseline performances comparable to various machine learning models trained on kinematic features (Chapter 5). CNNs learn directly from the data, and are capable of learning abstract features, with a great deal of potential for classification under more challenging scenarios.

This thesis has investigated the relationship between performance and SNR, contributing to the understanding of how range and RCS limit reliable classification (Chapter 5). SNR was reduced artificially through the injection of noise into the target spectrograms, reducing the visibility of drone micro-Doppler features. Whilst accuracy decreased with SNR, shallower networks were shown to be more robust to noisy test data.

Augmentation of the training data was shown to improve classifier robustness, where injecting noise into the signals used for training reduced the likelihood of the classifier overfitting on micro-Doppler features (Chapter 5). One of the key challenges of machine learning is the trade-off between bias (error caused by the incorrect approximation of the underlying target function) and variance (sensitivity to fluctuations within the training set). The aim is to approximate the target function closely enough to make accurate predictions, whilst avoiding learning the random noise in the training data (overfitting).

A methodology was developed for optimising the proportion of training data to augment, using Bayesian optimisation to tune augmentation hyperparameters

(Chapter 5). This methodology could be used to mitigate the reliance on SNR, extending the space in which reliable classification is possible, and permitting classification of smaller drone models at longer ranges.

A further major contribution of this thesis was the use of CNNs to classify drone spectrograms in which drone micro-Doppler features appeared as HERM lines. Experimental data collected using a Ku-band FMCW staring radar prototype were used to systematically classify drones from coarse groupings down to identification of nine drone models, building a more complete picture of airspace occupancy (Chapter 6). Drone groups were also classified at L-band, classifying size of drone as well as distinguishing them from birds, providing a novel contribution to the field as the bulk of the literature where spectrograms were used for classification used higher frequency radars for data collection. Classification of drone group or model could allow for greater understanding of the risk associated with drone sightings, allowing estimation of the potential damage caused by collision with a manned aircraft, and assessment of whether the drone could carry a heavy payload.

It is demonstrated that drone weight is more discriminative than the number of rotors, with significant implications for classification strategies (Chapter 6). These findings are the result of the first comprehensive investigation of CNN use for drone-to-drone classification under real operational conditions, investigating multiple classification strategies and exploring the effects of SNR and dwell-time on performance. The trade-off between accuracy and time taken to respond to threats has been systematically explored, providing recommendations for optimal dwell-times depending on accuracy requirements.

This thesis has also investigated the relationship between processing length and SNR, and the subsequent implications for performance (Chapter 6). Feature space is explored across various drone models, providing a further contribution to the field.

Decision tree classifiers are trained for drone group and model classification, using micro-Doppler features as well as kinematic features and features related to signal strength. The optimal processing length is shown to be a function of trajectory, target type, and the complexity of the classification problem. A classification pipeline is developed whereby parallel classifiers are trained using different integration lengths, and the outputs from each classifier are fused, leading to a significant improvement in performance. The systematic assessment of the effect of processing interval on performance contributes to the wider understanding of drone spectrogram features that can be exploited for classification.

Most of the existing research considered idealised scenarios, whereas this thesis explored classification upon an extended data set under increasingly demanding conditions (Chapter 7). This work has considered 20 different drone models (with diameters from 0.3 m to 2.14 m), extensive range coverage (up to 5 km), and a variety of clutter environments (urban, rural, and industrial), contributing to the understanding of performance limitations and progressing the solution space for drone classification.

Two classifiers were developed for drone-bird discrimination – one of which was optimal for classifying drones with clearly visible micro-Doppler, whilst the other was more robust to noise and had the ability to generalise to unseen scenarios. Dynamic classifier selection was used to make an intelligent choice of CNN classifier based on features extracted from the tracker and spectrogram, reducing false positives whilst maintaining a high true positive rate (Chapter 7). The fusion of two CNNs using a machine learning classifier for drone classification is a novel contribution of this work, and both hard and soft fusion approaches are presented using a decision tree classifier and a logistic regression model. This work lays the foundations for a cognitive approach to classification, with the potential to be scaled up in the future.

8.2 Future Work

Whilst strands of future work have been highlighted throughout the thesis, this section provides an overview of potential future research avenues focusing on wider extensions.

Future work could investigate the use of cognitive radar systems for adaptive selection of parameters for the optimisation of classification performance. For example, it has been shown that increasing the integration time can improve SNR and Doppler resolution provided the target remains coherent, and an adaptive system could make an intelligent choice of integration time to maximise the probability of correct classification for a given target. The parameters used for data collection or processing have a profound effect on spectrogram appearance, and this could be exploited to maximise the separation of classes within feature space.

Further investigation into CNN architectures is strongly recommended, as the pretrained networks considered in this thesis are not necessarily optimised for the classification of radar images. Perhaps custom networks could achieve better performances than the networks considered in this thesis, with wider implications for target classification using radar. Of course, there are many other deep learning algorithms besides CNNs that could be employed for drone signal classification. Algorithms such as recurrent neural networks, long short-term memory networks, gated recurrent units, transformers, and attention models can capture spatial and temporal features, and could form the basis of future research.

Future work could consider alternatives to the traditional order of detection, tracking, and classification. Classifying before detecting would decrease the number of false targets, reducing the processing overheads. Classifying a target before

tracking would result in fewer false tracks and false associations, and could assist with repairing track breaks [199].

Future work could further explore fusion approaches to maximise the information available to the classifier. Fusion of data from multiple nodes within a multistatic radar system could allow the classifier to benefit from different aspect angles and enhanced RCS. Additionally, information gathered from multiple radars could be processed in a single classification pipeline, providing features from different scattering regimes. Sensor fusion could allow the classifier to benefit from multiple modalities, for example, combining radar with optical methods.

Whilst CNNs have demonstrated high performance when trained on spectrograms, spectrograms are a single representation of the data. Other radar images such as range-Doppler maps, cadence velocity diagrams, and cepstrograms contain additional information about the target and could be used as the input to a CNN. Further representations such as track features, polarimetric features, and the time series signal could also be considered. Future work could seek to find an optimal combination of data representations, whether at the level of signal, feature, or decision fusion.

This thesis has only considered single targets, and classification performance has not been formally evaluated upon flocks of birds or swarms of drones. Flocks of birds may cause potential problems for the classifier due to multiple bird body returns within a resolution cell almost giving the appearance of HERM lines. Future work could investigate the effect of multiple targets on drone and bird spectrograms, and the subsequent implications for classification.

Finally, whilst a range of drone models, manufacturers, and sizes are considered in this thesis, this does not wholly represent the full spectrum of drones available on

the market. Drones are becoming smaller and increasingly manoeuvrable, with the development of micro-drones and 'smart dust' [19]. Further, stealth drones made of radar absorbing materials and blades with deliberately low RCS would provide a particular challenge for classification. Therefore, classification techniques must continually develop in order to keep pace with the evolution of drone technology.

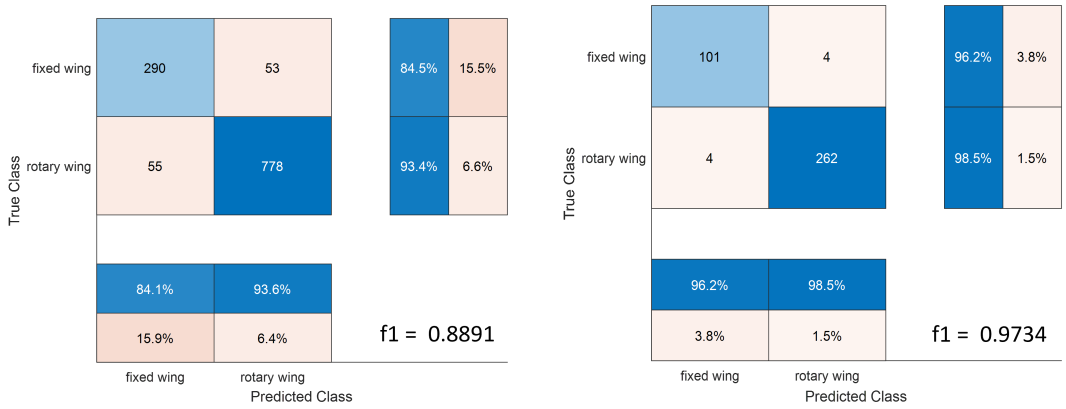
8.3 Conclusion

The aim of this thesis was to investigate the use of deep learning approaches for the classification of drones using radar, and to build on the existing knowledge base by considering UAV classification as a continuum of problems and achieving classification under increasingly difficult conditions. Data driven and systematic approaches to drone vs bird discrimination and drone identification have been conducted, developing convolutional neural networks capable of accurate classification under a range of scenarios. Deep learning algorithms show great potential for the automatic classification of UAVs, facilitating the use of radar for monitoring low-altitude airspace occupancy as drone numbers proliferate, ensuring safety in the skies.

Appendix A

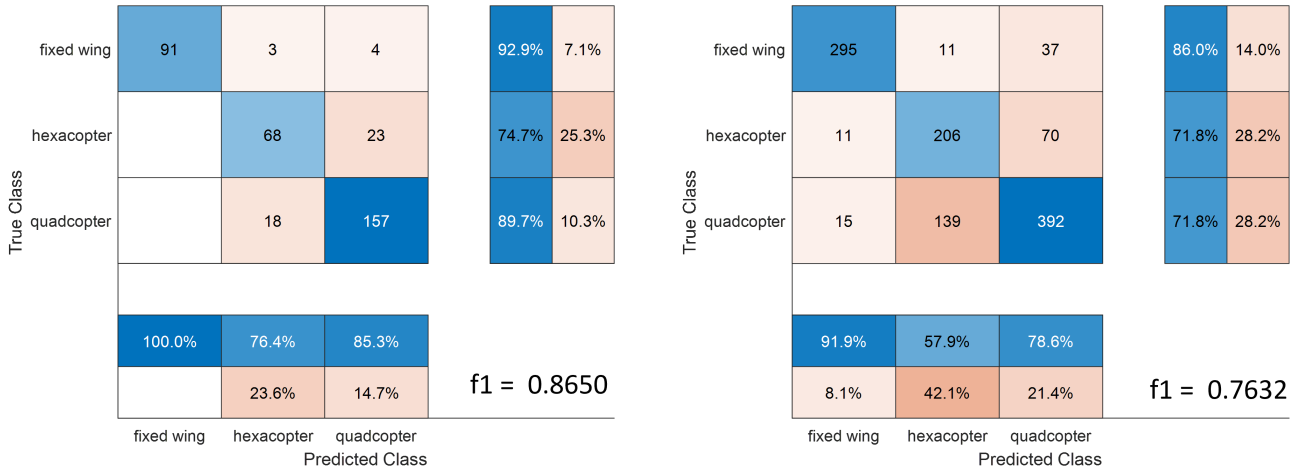
Drone Classification Per-file Confusion Matrices

This appendix contains confusion matrices for the cases where drone group, sub-group, and model classification were carried out across entire tracks. Only included are the figures that are not present elsewhere in Chapter 6.



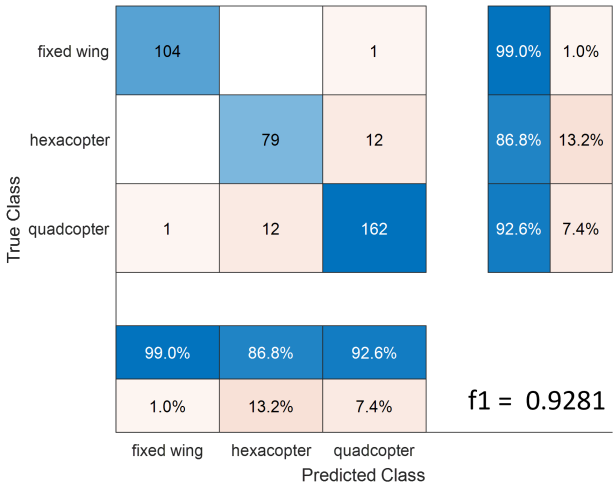
(a) (b)

Figure A.1: Drone group classification performance where classification has been performed across whole files a) Low signal to noise ratio case, b) All data case.



(a)

(b)



(c)

Figure A.2: Confusion matrices showing drone subgroup classification performance where classification has been performed across whole files a) Baseline case, b) Low signal to noise ratio case, c) All data case.

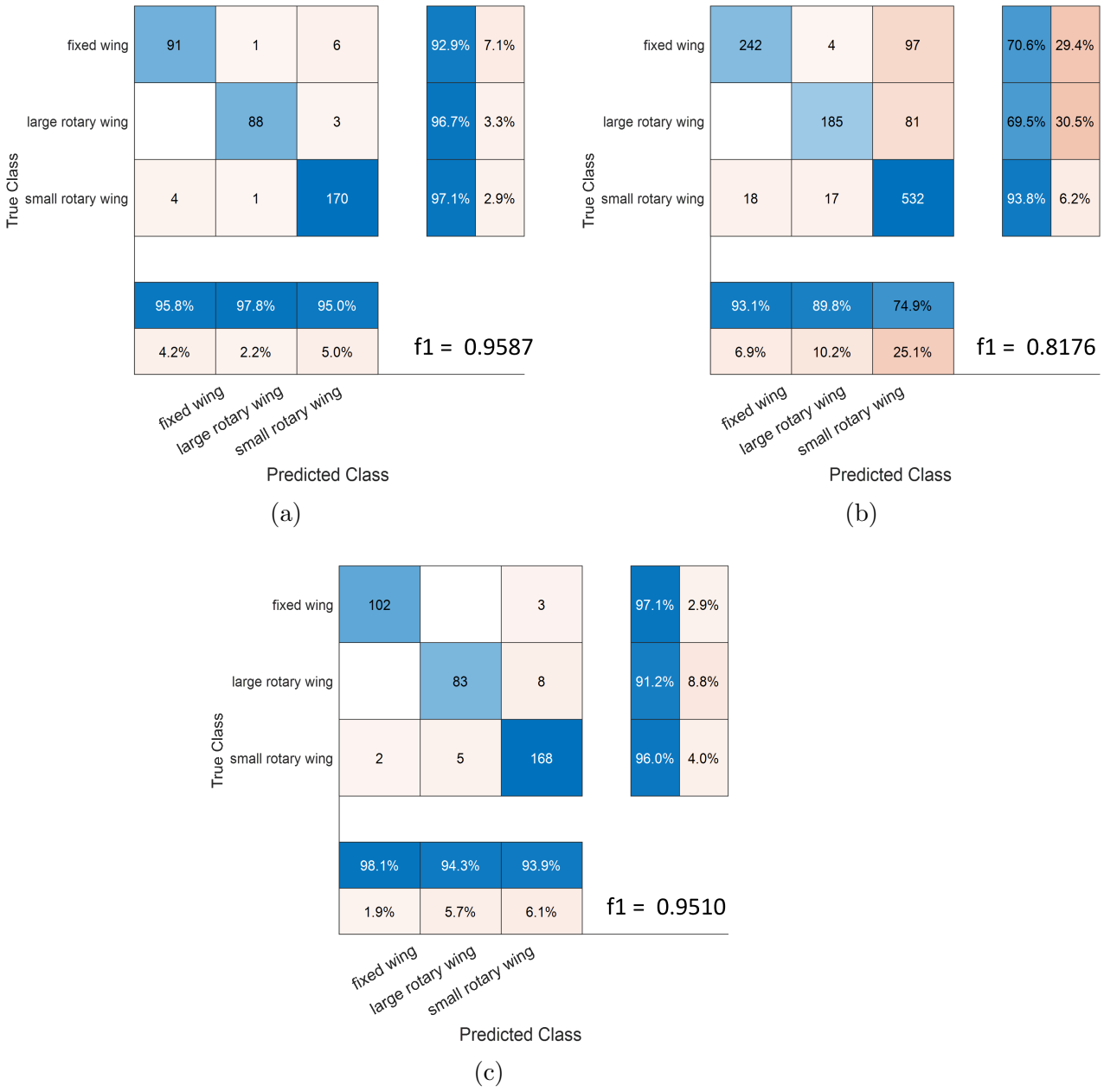
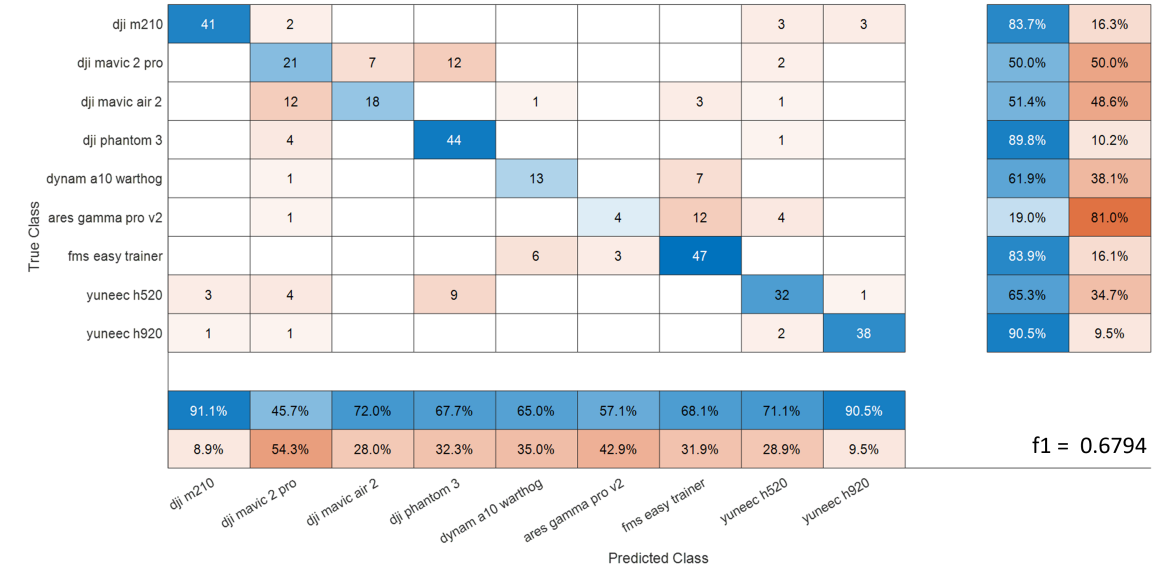
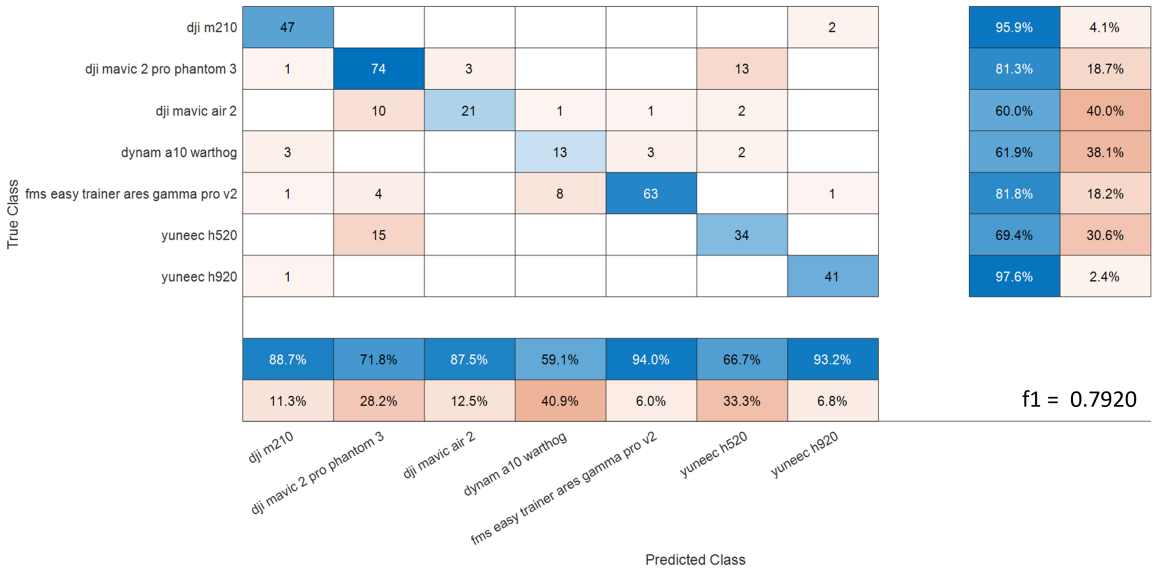


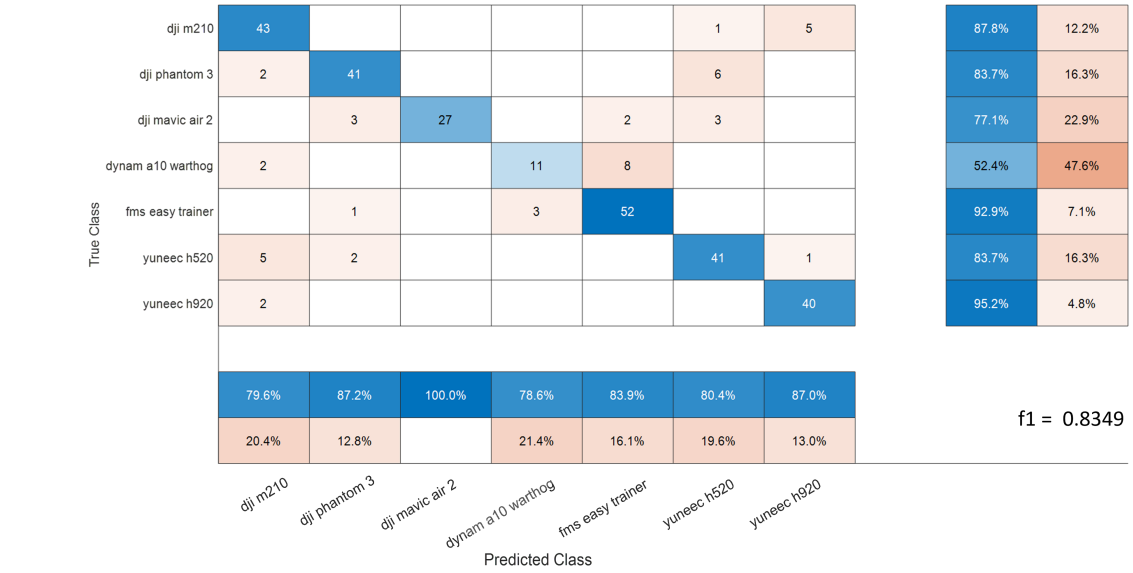
Figure A.3: Confusion matrices showing drone size classification performance where classification has been performed across whole files a) Baseline case, b) Low signal to noise ratio case, c) All data case.


dji m210
dji mavic 2 pro
dji mavic air 2
dji phantom 3
dynam a10 warthog
ares gamma pro v2
fms easy trainer
yuneec h520
yuneec h920
True Class
41 2 3 3
21 7 12 2
12 18 1 3 1
4 44 1
1 13 7
1 4 12 4
6 3 47
3 4 9 32 1
1 1 2 38
83.7% 16.3%
50.0% 50.0%
51.4% 48.6%
89.8% 10.2%
61.9% 38.1%
19.0% 81.0%
83.9% 16.1%
65.3% 34.7%
90.5% 9.5%
91.1% 45.7% 72.0% 67.7% 65.0% 57.1% 68.1% 71.1% 90.5%
8.9% 54.3% 28.0% 32.3% 35.0% 42.9% 31.9% 28.9% 9.5%
f1 = 0.6794
dji m210
dji mavic 2 pro
dji mavic air 2
dji phantom 3
dynam a10 warthog
ares gamma pro v2
fms easy trainer
yuneec h520
yuneec h920
Predicted Class

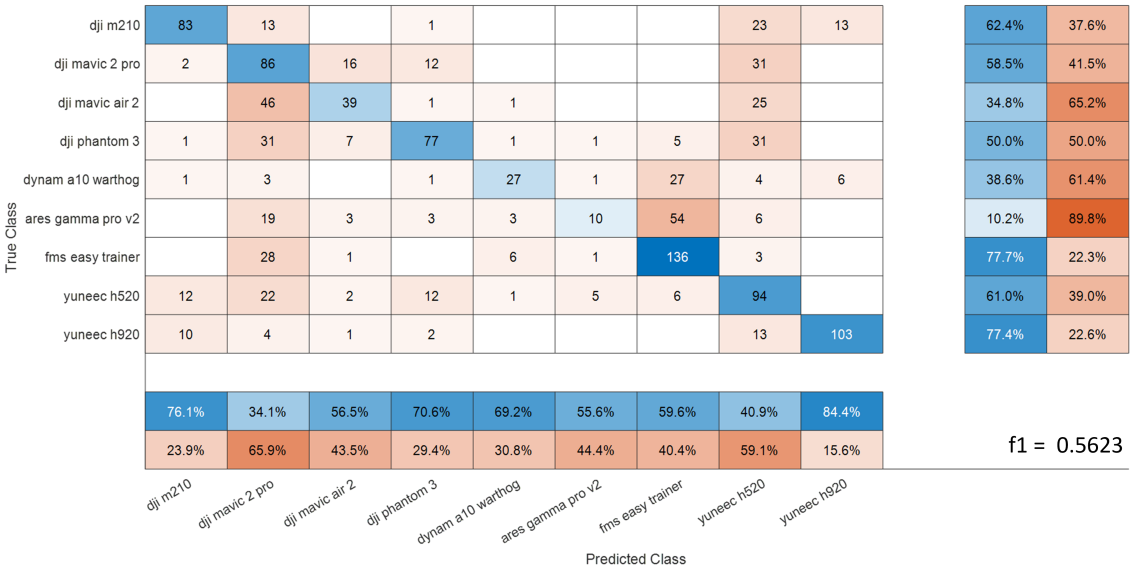
(a)


dji m210
dji mavic 2 pro phantom 3
dji mavic air 2
dynam a10 warthog
fms easy trainer ares gamma pro v2
yuneec h520
yuneec h920
True Class
47 2
1 74 3 13
10 21 1 1 2
3 13 3 2
1 4 8 63 1
15 34
1 41
95.9% 4.1%
81.3% 18.7%
60.0% 40.0%
61.9% 38.1%
81.8% 18.2%
69.4% 30.6%
97.6% 2.4%
88.7% 71.8% 87.5% 59.1% 94.0% 66.7% 93.2%
11.3% 28.2% 12.5% 40.9% 6.0% 33.3% 6.8%
f1 = 0.7920
dji m210
dji mavic 2 pro phantom 3
dji mavic air 2
dynam a10 warthog
fms easy trainer ares gamma pro v2
yuneec h520
yuneec h920
Predicted Class

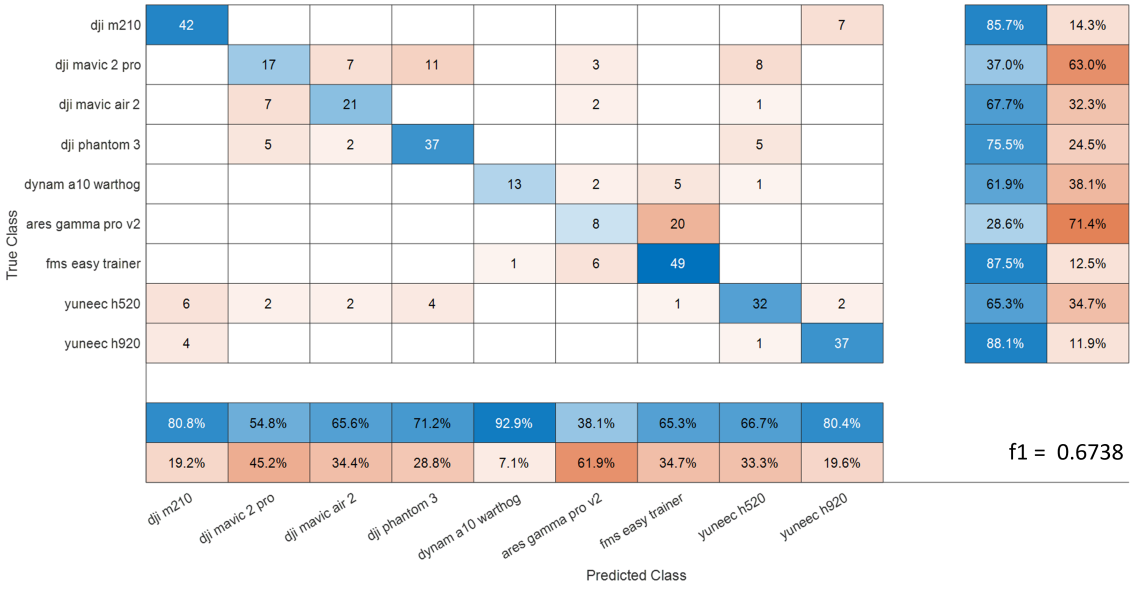
(b)


dji m210
dji phantom 3
dji mavic air 2
dynam a10 warthog
fms easy trainer
yuneec h520
yuneec h920
True Class
43 1 5
2 41 6
3 27 2 3
2 11 8
1 3 52
5 2 41 1
2 40
87.8% 12.2%
83.7% 16.3%
77.1% 22.9%
52.4% 47.6%
92.9% 7.1%
83.7% 16.3%
95.2% 4.8%
79.6% 87.2% 100.0% 78.6% 83.9% 80.4% 87.0%
20.4% 12.8% 21.4% 16.1% 19.6% 13.0%
f1 = 0.8349
dji m210
dji phantom 3
dji mavic air 2
dynam a10 warthog
fms easy trainer
yuneec h520
yuneec h920
Predicted Class

(c)

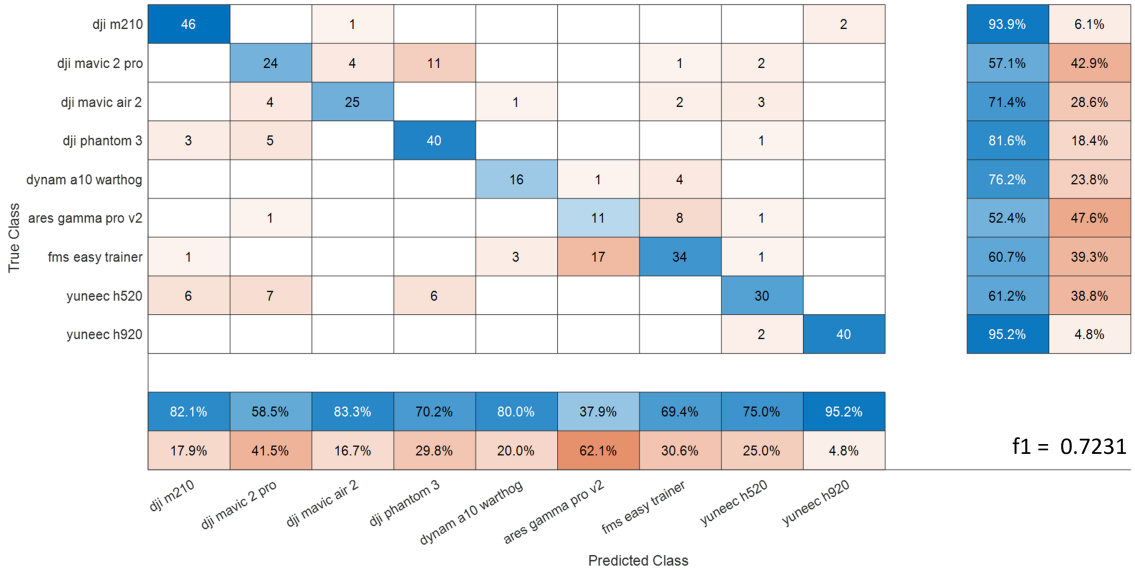

dji m210
dji mavic 2 pro
dji mavic air 2
dji phantom 3
dynam a10 warthog
ares gamma pro v2
fms easy trainer
yuneec h520
yuneec h920
True Class
83 13 1 23 13
2 86 16 12 31
46 39 1 1 25
1 31 7 77 1 1 5 31
1 3 1 27 1 27 4 6
19 3 3 3 10 54 6
28 1 6 1 136 3
12 22 2 12 1 5 6 94
10 4 1 2 13 103
62.4% 37.6%
58.5% 41.5%
34.8% 65.2%
50.0% 50.0%
38.6% 61.4%
10.2% 89.8%
77.7% 22.3%
61.0% 39.0%
77.4% 22.6%
76.1% 34.1% 56.5% 70.6% 69.2% 55.6% 59.6% 40.9% 84.4%
23.9% 65.9% 43.5% 29.4% 30.8% 44.4% 40.4% 59.1% 15.6%
f1 = 0.5623
dji m210
dji mavic 2 pro
dji mavic air 2
dji phantom 3
dynam a10 warthog
ares gamma pro v2
fms easy trainer
yuneec h520
yuneec h920
Predicted Class

(d)

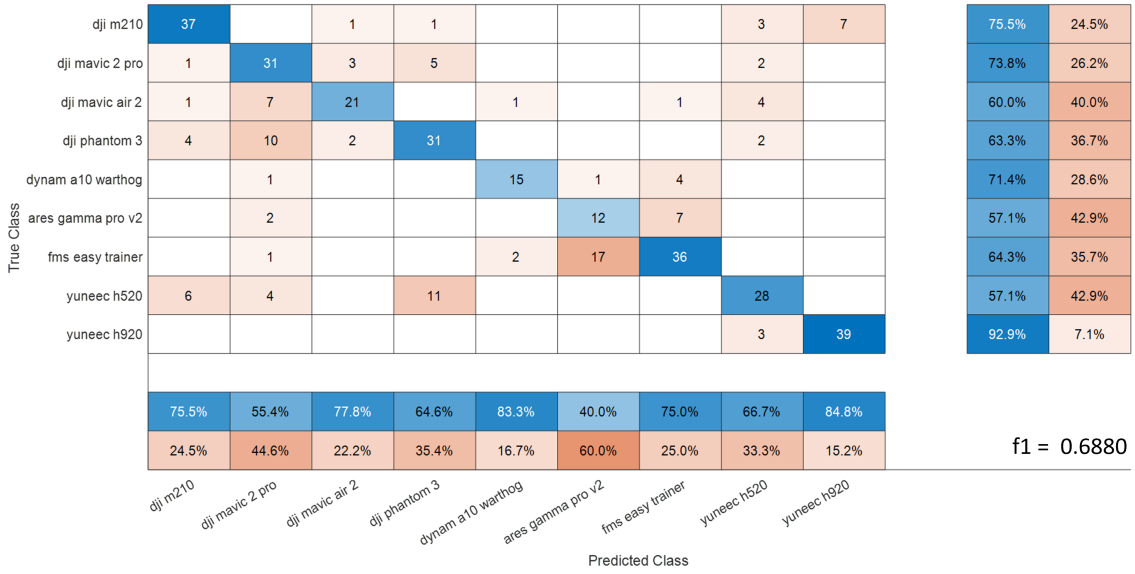


(e)

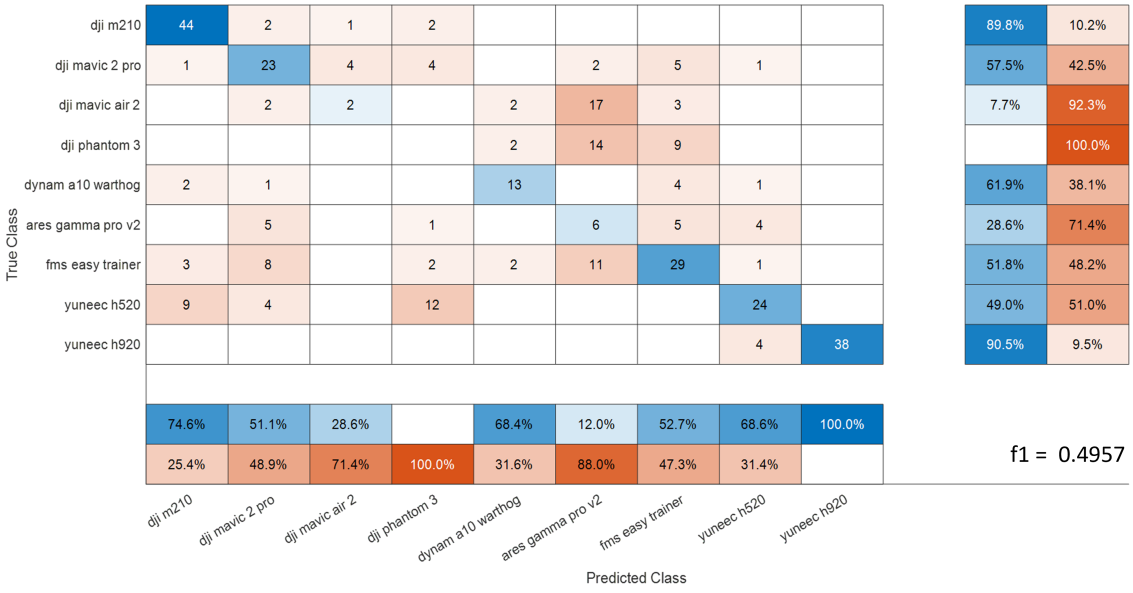
Figure A.4: Confusion matrices showing single-stage drone model classification performance where classification has been performed across whole files a) b) Baseline case, 9 classes b) Baseline case, 7 classes (weaker classes combined) c) Baseline case, 7 classes (weaker classes removed) d) Low signal to noise ratio (SNR) case, 9 classes, e) All SNR case, 9 classes.


dji m210 46 1 2 93.9% 6.1%
dji mavic 2 pro 24 4 11 1 2 57.1% 42.9%
dji mavic air 2 4 25 1 2 3 71.4% 28.6%
dji phantom 3 3 5 40 1 81.6% 18.4%
dynam a10 warthog 16 1 4 76.2% 23.8%
ares gamma pro v2 1 11 8 1 52.4% 47.6%
fms easy trainer 1 3 17 34 1 60.7% 39.3%
yuneec h520 6 7 6 30 61.2% 38.8%
yuneec h920 2 40 95.2% 4.8%
True Class
82.1% 58.5% 83.3% 70.2% 80.0% 37.9% 69.4% 75.0% 95.2%
17.9% 41.5% 16.7% 29.8% 20.0% 62.1% 30.6% 25.0% 4.8%
f1 = 0.7231
dji m210
dji mavic 2 pro
dji mavic air 2
dji phantom 3
dynam a10 warthog
ares gamma pro v2
fms easy trainer
yuneec h520
yuneec h920
Predicted Class

(a)


dji m210 37 1 1 3 7 75.5% 24.5%
dji mavic 2 pro 1 31 3 5 2 73.8% 26.2%
dji mavic air 2 1 7 21 1 1 4 60.0% 40.0%
dji phantom 3 4 10 2 31 2 63.3% 36.7%
dynam a10 warthog 1 15 1 4 71.4% 28.6%
ares gamma pro v2 2 12 7 57.1% 42.9%
fms easy trainer 1 2 17 36 64.3% 35.7%
yuneec h520 6 4 11 28 57.1% 42.9%
yuneec h920 3 39 92.9% 7.1%
True Class
75.5% 55.4% 77.8% 64.6% 83.3% 40.0% 75.0% 66.7% 84.8%
24.5% 44.6% 22.2% 35.4% 16.7% 60.0% 25.0% 33.3% 15.2%
f1 = 0.6880
dji m210
dji mavic 2 pro
dji mavic air 2
dji phantom 3
dynam a10 warthog
ares gamma pro v2
fms easy trainer
yuneec h520
yuneec h920
Predicted Class

(b)



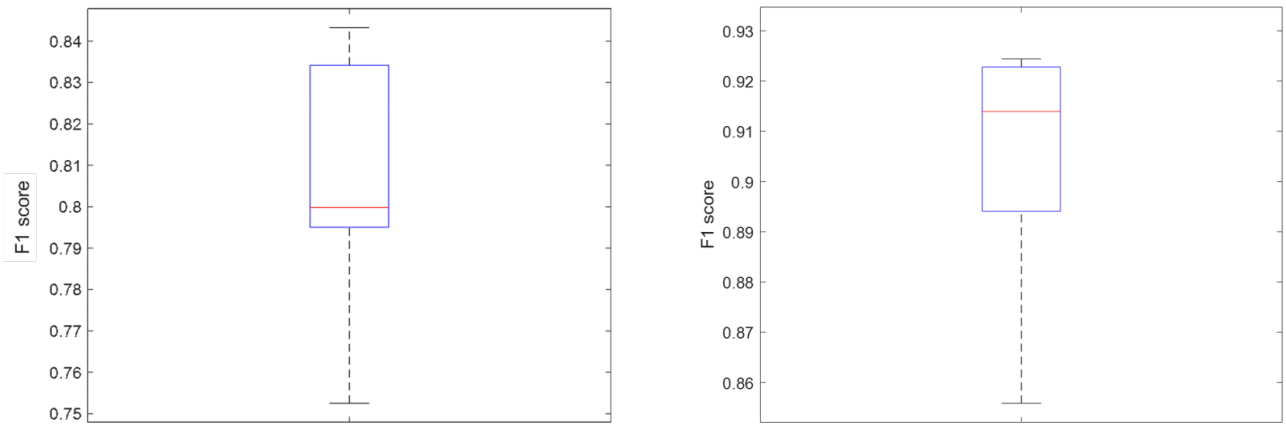
(c)

Figure A.5: Confusion matrices showing multi-stage drone model classification performance where classification has been performed across whole files a) 2-stages: Group classification followed by model classification, b) 2-stages: Subgroup classification followed by model classification, c) 3-stage classifier.

Appendix B

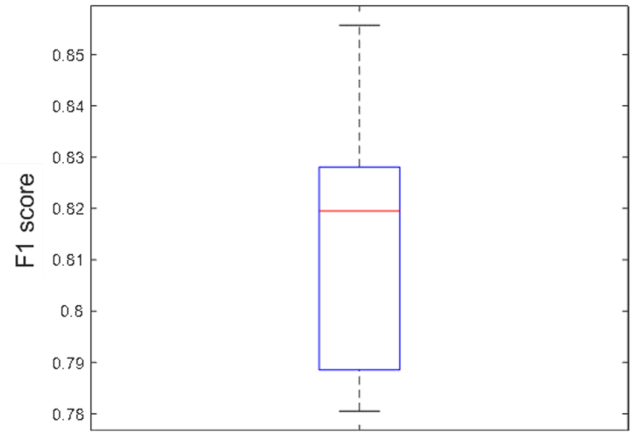
Drone Classification F1 Box Plots

This appendix contains the F1 score box plots for the cases where drone group, subgroup, and model classification were carried out on a per-frame basis. Only included are the figures that are not present elsewhere in Chapter 6.

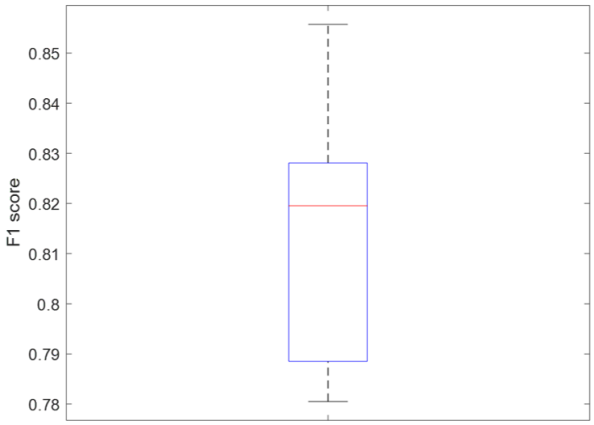


(a) (b)

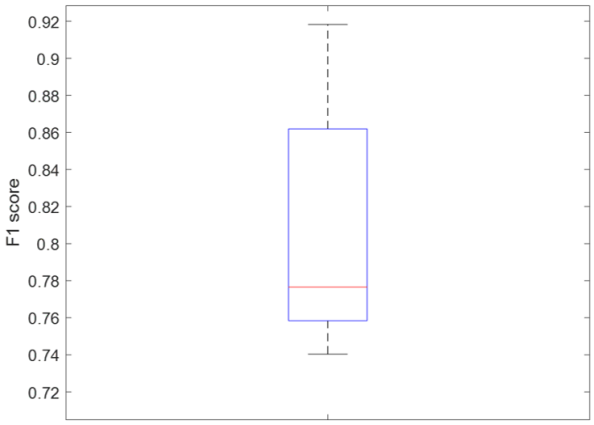
Figure B.1: Drone group classification F1 score median and interquartile range a) Low signal to noise ratio case, b) All data case.



(a)



(b)



(c)

Figure B.2: Drone subgroup classification F1 score median and interquartile range a) Baseline case b) Low signal to noise ratio case, c) All data case.

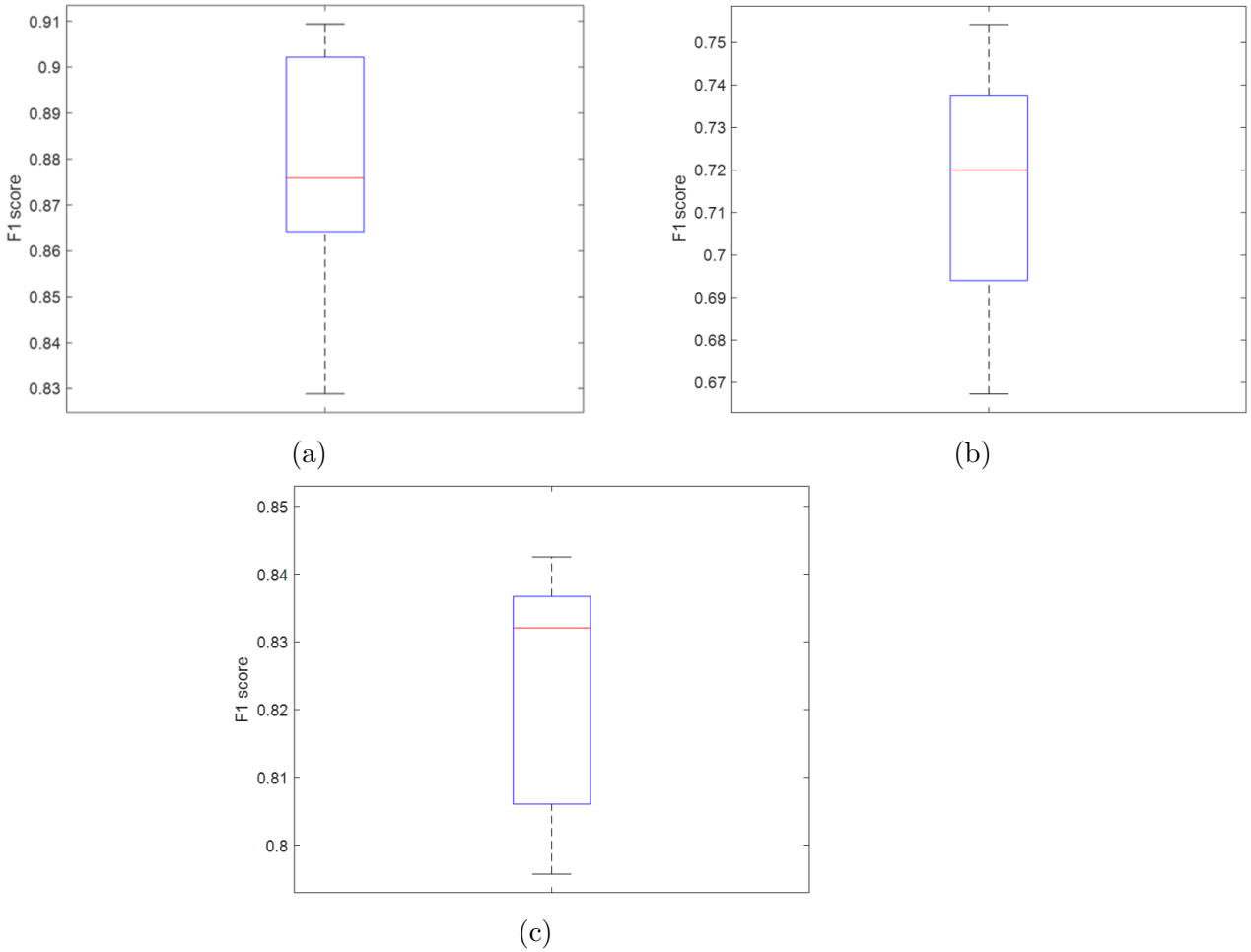


Figure B.3: Drone size classification F1 score median and interquartile range a) Baseline case b) Low signal to noise ratio case, c) All data case.

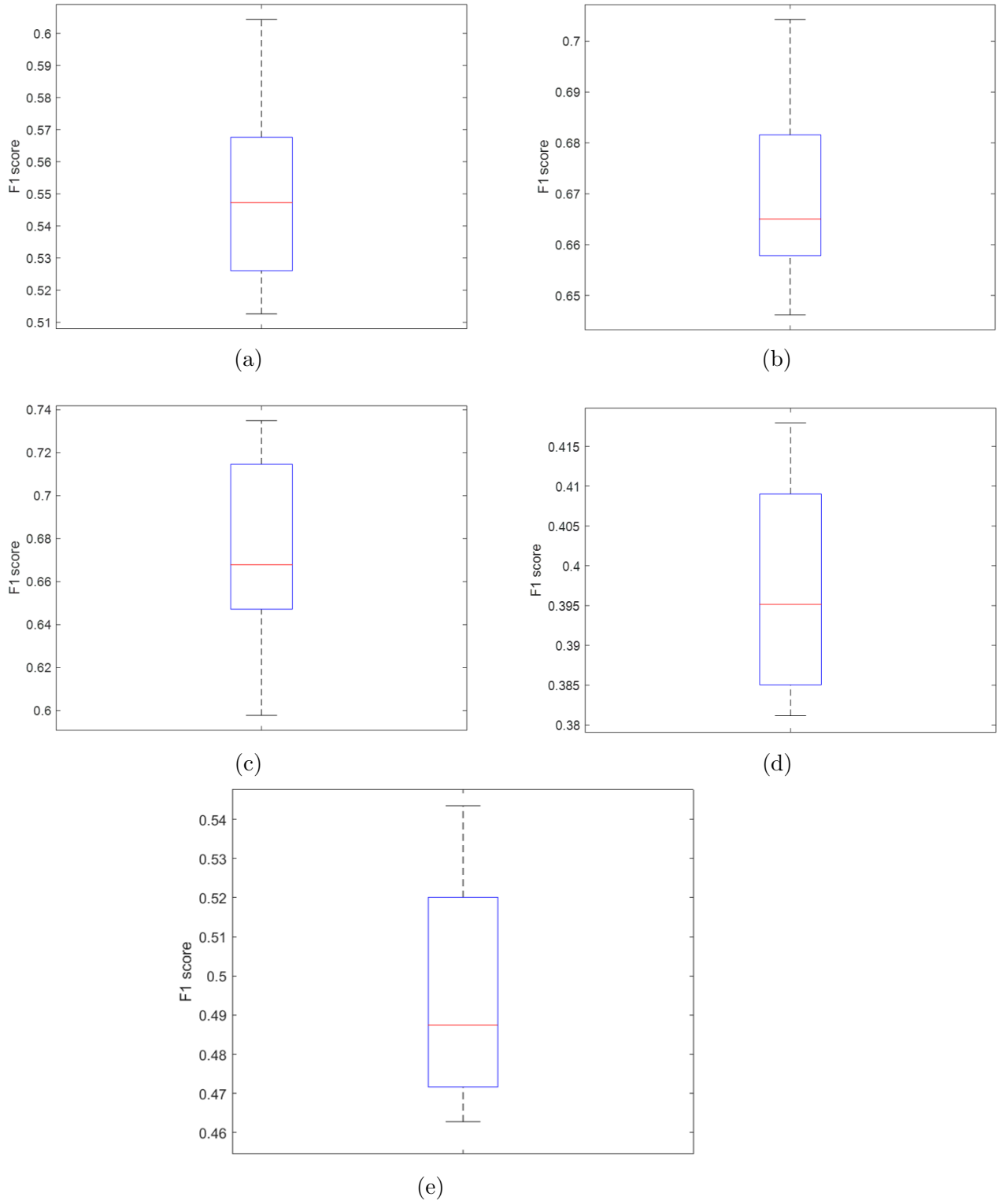


Figure B.4: Drone model single-stage F1 score median and interquartile range a) Baseline case, 9 classes, b) Baseline case, 7 classes (weaker classes combined) c) Baseline case, 7 classes (weaker classes removed) d) Low signal to noise ratio case, 9 classes e) All data case, 9 classes.

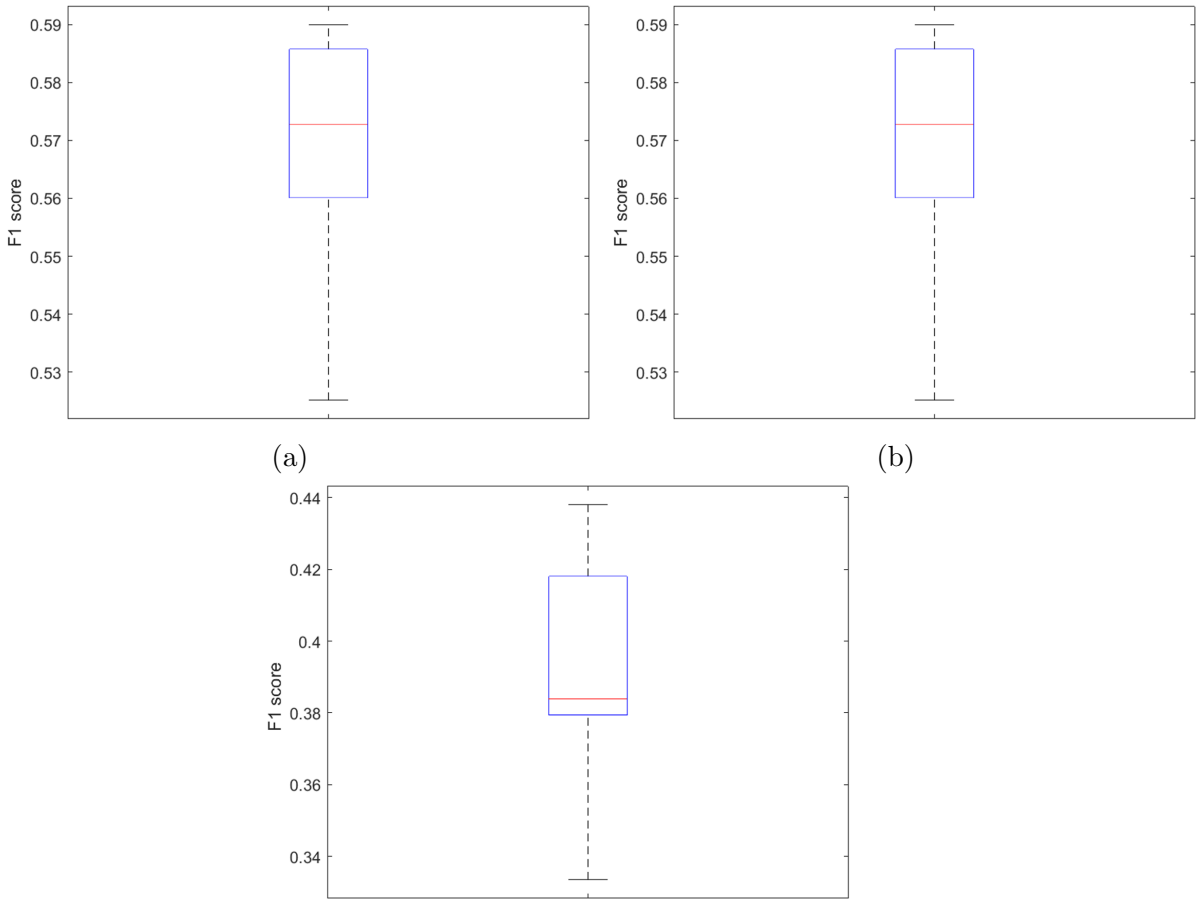


Figure B.5: Drone model multi-stage classification F1 score median and range a) 2-stages: Group classification followed by model classification, b) 2-stages: Subgroup classification followed by model classification, c) 3-stage classifier.

Appendix C

Drone Classification Performance Summary

This appendix contains the summary tables for drone group, subgroup and model classification. F1 scores are presented for the three SNR regimes considered, where classification was performed on single frames and single files.

Table C.1: Summary of the F1 scores achieved by the drone group classifiers.

Classes	Training SNR	Testing SNR	Classification	F1 score
Fixed wing vs rotary wing drone	>30 dB	>30 dB	Single frame	0.9361
Fixed wing vs rotary wing drone	>30 dB	>30 dB	Single file	0.9825
Fixed wing vs rotary wing drone	>30 dB	<30 dB	Single frame	0.7939
Fixed wing vs rotary wing drone	>30 dB	<30 dB	Single file	0.8891
Fixed wing vs rotary wing drone	All	All	Single frame	0.9022
Fixed wing vs rotary wing drone	All	All	Single file	0.9734

Table C.2: Summary of the F1 scores achieved by the drone subgroup classifiers, where the rotary wing drones are grouped by number of rotors.

Classes	Training SNR	Testing SNR	Classification	F1 score
Fixed vs quad- vs hexa-copter	>30 dB	>30 dB	Single frame	0.8132
Fixed vs quad- vs hexa-copter	>30 dB	>30 dB	Single file	0.8650
Fixed vs quad- vs hexa-copter	>30 dB	<30 dB	Single frame	0.6218
Fixed vs quad- vs hexa-copter	>30 dB	<30 dB	Single file	0.7632
Fixed vs quad- vs hexa-copter	All	All	Single frame	0.8036
Fixed vs quad- vs hexa-copter	All	All	Single file	0.9281

Table C.3: Summary of the F1 scores achieved by the drone size subgroup classifiers.

Classes	Training SNR	Testing SNR	Classification	F1 score
Fixed vs large rotary- vs small rotary	>30 dB	>30 dB	Single frame	0.8784
Fixed vs large rotary- vs small rotary	>30 dB	>30 dB	Single file	0.9587
Fixed vs large rotary- vs small rotary	>30 dB	<30 dB	Single frame	0.7100
Fixed vs large rotary- vs small rotary	>30 dB	<30 dB	Single file	0.8176
Fixed vs large rotary- vs small rotary	All	All	Single frame	0.8176
Fixed vs large rotary- vs small rotary	All	All	Single file	0.9510

Table C.4: Summary of the F1 scores achieved by the drone model classifiers.

Classes	Stages	Training SNR	Testing SNR	Classification	F1 score
9 classes	Single stage	>30 dB	>30 dB	Single frame	0.5542
9 classes	Single stage	>30 dB	>30 dB	Single file	0.6794
9 classes	Single stage	>30 dB	<30 dB	Single frame	0.3901
9 classes	Single stage	>30 dB	<30 dB	Single file	0.5623
9 classes	Single stage	All	All	Single frame	0.4861
9 classes	Single stage	All	All	Single file	0.6738
9 classes	2 stages (A)	>30 dB	>30 dB	Single frame	0.5619
9 classes	2 stages (A)	>30 dB	>30 dB	Single file	0.7231
9 classes	2 stages (B)	>30 dB	>30 dB	Single frame	0.5410
9 classes	2 stages (B)	>30 dB	>30 dB	Single file	0.6880
9 classes	3 stages	>30 dB	>30 dB	Single frame	0.4027
9 classes	3 stages	>30 dB	>30 dB	Single file	0.4957

Bibliography

[1] Google. Google Trends. `https://trends.google.com/trends/?geo=GB`. Accessed: 2022-07-11.

[2] M. A Richards, Jim Scheer, and William A Holm. *Principles of modern radar*, chapter 9.2. Scitech Pub., 2010.

[3] J. J. M de Wit, R. I. A. Harmanny, and G. Prémel-Cabic. Micro-Doppler analysis of small UAVs. In *2012 9th European Radar Conference*, pages 210–213, 2012.

[4] Daniel B Herr and Dave Tahmoush. Data-driven STFT for UAV Micro-Doppler signature analysis. In *2020 IEEE International Radar Conference (RADAR)*, pages 1023–1028, 2020.

[5] M. A Richards, Jim Scheer, and William A Holm. *Principles of modern radar*, chapter 14.4. Scitech Pub., 2010.

[6] Mohammed Jahangir, Chris J Baker, and Gordon A Oswald. Doppler characteristics of micro-drones with L-Band multibeam staring radar. In *2017 IEEE Radar Conference (RadarConf)*, pages 1052–1057, 2017.

[7] 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 & Navigation*, pages 435–444, 2021.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, vol 60:84–90, May 2017.

[9] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and 0.5MB model size, 2016.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv*, 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv*, 2015.

[12] Google. Google Earth. `https://www.google.com/intl/en_uk/earth/`. Accessed: 2021-11-02.

[13] Hunter Research and Technology LLC. THEODOLITE: The original, indispensable, pioneering AR viewfinder. `http://hrtapps.com/theodolite/`. Accessed: 2021-11-02.

[14] MathWorks. Pretrained deep neural networks. `https://uk.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html`. Accessed: 2022-06-03.

[15] Ismail Guvenc, Farshad Koohifar, Simran Singh, Mihail L. Sichitiu, and David Matolak. Detection, tracking, and interdiction for amateur drones. *IEEE Communications Magazine*, vol 56:p. 75–81, 2018.

[16] Georgia Lykou, Dimitrios Moustakas, and Dimitris Gritzalis. Defending Airports from UAS: A Survey on Cyber-Attacks and Counter-Drone Sensing Technologies. *Sensors*, vol 20:p. 3537, 2020.

[17] PwC UK. The impact of drones on the UK economy. https://www.pwc.co.uk/dronesreport. Accessed: 2022-07-11.

[18] GOV.UK. Drones: how to fly them safely and legally. https://www.gov.uk/government/news/drones-are-you-flying-yours-safely-and-legally, September 2017. Accessed: 2022-07-11.

[19] M. Hassanalian and A. Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, vol 91:p. 99–131, 2017.

[20] About Amazon. Amazon Prime Air prepares for drone deliveries. https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries, June 2022. Accessed: 2022-07-11.

[21] Lydia Harriss and Zara Mir. *Misuse of Civilian Drones*. UK Parliament POST 610. UK Parliament, 2020.

[22] Jian Wang, Yongxin Liu, and Houbing Song. Counter-unmanned aircraft system(s) (C-UAS): State of the art, challenges and future trends. *IEEE Aerospace and Electronic Systems Magazine*, 2020.

[23] UK Department for Transport and Military Aviation Authority. Small Remotely Piloted Aircraft Systems (Drones) Mid-Air Collision Study. https://www.gov.uk/government/publications/drones-and-manned-aircraft-collisions-test-results, 2017.

[24] UK Airprox Board. Analysis of Airprox in UK Airspace: January to December 2020. Technical report, UK Airprox Board, 2021. https://www.airproxboard.org.uk/media/oahp00s3/bluebook36.pdf.

[25] Matthew Weaver, Damien Gayle, Patrick Greenfield, and Frances Perraudin. Military called in to help with Gatwick drone crisis. https://www.theguard

ian.com/uk-news/2018/dec/19/gatwick-flights-halted-after-drone-sighting, December 2018. Accessed: 2022-07-11.

[26] Calder Simon. Gatwick drone disruption over Christmas cost £50m. www.independent.co.uk/travel/news-and-advice/gatwick-drone-airport-cost-easyjet-runway-security-passenger-cancellation-a8739841.html, 2019. Accessed: 2022-07-11.

[27] Dan Shea, Amanda Essex, and Ben Husch. Drones and Critical Infrastructure. https://www.ncsl.org/research/energy/drones-and-critical-infrastructure.aspx, 2016. Accessed: 2022-07-11.

[28] Katie Jones. How Organized Crime Networks Are Using Drones to Their Advantage. https://insightcrime.org/news/brief/drones-narcotrafficking-surveillance, 2020. Accessed: 2022-07-11.

[29] Sinclair Hudson and Bhashyam Balaji. Application of machine learning for drone classification using radars. In *Signal Processing, Sensor/Information Fusion, and Target Recognition XXX*, pages 72 – 84. SPIE, 2021.

[30] UK Department for Transport and Ministry of Defence. Public dialogue on drone use in the UK. https://www.gov.uk/government/publications/drone-use-in-the-uk-public-dialogue, 2016.

[31] Sam Gyimah. New squad formed to tackle drone threat to prisons. https://www.gov.uk/government/news/new-squad-formed-to-tackle-drone-threat-to-prisons, April 2017. Accessed: 2022-07-11.

[32] Rob Moore. Guernsey prison testing pioneering anti-drone tech. https://www.itv.com/news/channel/2017-05-12/guernsey-prison-testing-pioneering-anti-drone-tech, May 2017. Accessed: 2022-07-11.

[33] David Dunn. Small drones and the use of chemical weapons as a terrorist threat. https://www.birmingham.ac.uk/research/perspective/small-drones-chemical-weapons-terrorist-threat.aspx. Accessed: 2022-07-11.

[34] BBC News. Ukraine conflict: How are drones being used? https://www.bbc.co.uk/news/world-62225830, 2022. Accessed: 2022-08-02.

[35] Arthur Michel. Counter-Drone Systems. Technical report, The Center for the Study of the Drone, Bard College, 2018. https://dronecenter.bard.edu/files/2018/02/CSD-Counter-Drone-Systems-Report.pdf.

[36] Anna Holligan. Eagles trained to take down drones. *BBC News*, March 2016. https://www.bbc.co.uk/news/av/world-europe-35750816, Accessed: 2022-07-11.

[37] Mohammed Jahangir and Chris Baker. Persistence surveillance of difficult to detect micro-drones with L-band 3-D holographic radar™. In *2016 CIE International Conference on Radar (RADAR)*, pages 1–5, 2016.

[38] Matthew Ritchie, Francesco Fioranelli, Hugh Griffiths, and Borge Torvik. Micro-drone RCS analysis. In *2015 IEEE Radar Conference*, pages 452–456, 2015.

[39] C.R. Vaughn. Birds and insects as radar targets: A review. *Proceedings of the IEEE*, vol 73:p. 205–227, 1985.

[40] Samiur Rahman and Duncan A. Robertson. Radar micro-Doppler signatures of drones and birds at K-band and W-band. *Scientific Reports*, vol 8, 2018.

[41] 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)*, 2022.

[42] H. Dale, C.J. Baker, M. Antoniou, and M. Jahangir. An Initial Investigation into Using Convolutional Neural Networks for Classification of Drones. In *2020 IEEE International Radar Conference (RADAR)*, pages 618–623, 2020.

[43] 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. In *2021 IEEE Radar Conference (RadarConf21)*, pages 1–5, 2021.

[44] H. Dale, M. Antoniou, C.J. Baker, M. Jahangir, and A. Catherall. Convolutional neural networks for drone model classification. In *18th European Radar Conference (EuRAD 2021)*, 2022.

[45] Christian Hülsmeyer. Patent DE165546: Verfahren, um metallische gegenstände mittels elektrischer wellen einem beobachter zu melden. *Deutschland Patent*, 1904.

[46] Christian Hülsmeyer. Patent DE169154: Verfahren zur bestimmung der entfernung von metallischen gegenständen (schiffen o. dgl.), deren gegenwart durch das verfahren nach Patent 165546 festgestellt wird. *Deutschland Patent*, 1904.

[47] M. A. Richards. *Fundamentals of radar signal processing.* McGraw Hill, 2022.

[48] B. T. Neale. CH — the first operational radar. *GEC journal of research*, vol3:p. 73–83, 1985.

[49] J. Lombacher and C. Wohler. Potential of radar for static object classification using deep learning methods. *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 2016.

[50] K. Patel, K. Rambach, T. Visentin, D. Rusev, M. Pfeiffer, and B. Yang. Deep learning-based object classification on automotive radar spectra. *IEEE Radar Conference 2019*, 2019.

[51] Moeness Amin, Yimin Zhang, Fauzia Ahmad, and K.C. Ho. Radar signal processing for elderly fall detection: The future for in-home monitoring. *IEEE Signal Processing Magazine*, Vol 33:p. 71–80, 2016.

[52] Branka Jokanovic and Moeness Amin. Fall detection using deep learning in range-doppler radars. *IEEE Transactions on Aerospace and Electronic Systems*, 2017.

[53] M. A Richards, Jim Scheer, and William A Holm. *Principles of modern radar*, chapter 2.5. Scitech Pub., 2010.

[54] R.C Johnson. *Antenna engineering handbook*. McGraw-Hill, New York, 3. ed edition, 1993.

[55] M. A Richards, Jim Scheer, and William A Holm. *Principles of modern radar*, chapter 2.7. Scitech Pub., 2010.

[56] M. Jahangir and C.J. Baker. Characterisation of low observable targets with a multi-beam staring radar. In *International Conference on Radar Systems (Radar 2017)*, 2017.

[57] M. A Richards, Jim Scheer, and William A Holm. *Principles of modern radar*, chapter 6.4. Scitech Pub., 2010.

[58] Jiangkun Gong, Jun Yan, Deren Li, and Huiping Hu. Interference of radar detection of drones by birds. In *Progress In Electromagnetics Research M*, volume 81, pages 1–11, 2019.

[59] T Young. I. The Bakerian Lecture. experiments and calculations relative to physical optics. *Philosophical Transactions of the Royal Society of London*, vol 94:p. 1–16, 1804.

[60] Samuel M Sherman and David K Barton. *Monopulse principles and techniques*. Artech House, 1984.

[61] M. A Richards, Jim Scheer, and William A Holm. *Principles of modern radar*. Scitech Pub., 2010.

[62] Stephen Harman. Analysis of the radar return of micro-UAVs in flight. In *2017 IEEE Radar Conference (RadarConf)*, pages 1159–1164, 2017.

[63] Paul A. Lynn and Wolfgang Fuerst. *Introductory digital signal processing with computer applications*. John Wiley, 2nd edition, 1998.

[64] M. Jahangir, G. Atkinson, M. Antoniou, C. J. Baker, J. P. Sadler, and S. J. Reynolds. Measurements of Birds and Drones with L-Band Staring Radar. In *2021 21st International Radar Symposium (IRS)*. IEEE, 2021.

[65] V.C. Chen, Fayin Li, Shen Shyang ho, and Harry Wechsler. Micro-Doppler Effect in Radar: Phenomenon, Model, and Simulation Study. *Aerospace and Electronic Systems, IEEE Transactions on*, Vol 42:p. 2 – 21, 2006.

[66] D. Blacknell and H. Griffiths. *Radar Automatic Target Recognition (ATR) and Non-Cooperative Target Recognition (NCTR)*. Radar, Sonar & Navigation Series. Institution of Engineering and Technology, 2013.

[67] Peter Tait. *Introduction to Radar Target Recognition*, chapter 5.2. Radar, Sonar & Navigation. Institution of Engineering and Technology, 2005.

[68] J. Martin and B. Mulgrew. Analysis of the theoretical radar return signal from aircraft propeller blades. In *IEEE International Conference on Radar*, pages 569–572, 1990.

[69] Vikas Agnihotri, Munish Sabharwal, and Vinay Goyal. Effect of frequency on micro-doppler signatures of a helicopter. *International Conference on Advances in Big Data, Computing and Data Communication Systems*, pages 1–5, 2019.

[70] Peter Tait. *Introduction to Radar Target Recognition*, chapter 5.3. Radar, Sonar & Navigation. Institution of Engineering and Technology, 2005.

[71] M.R. Bell and R.A. Grubbs. JEM modeling and measurement for radar target identification. *IEEE Transactions on Aerospace and Electronic Systems*, Vol 29:p.73–87, 1993.

[72] Xiaolin Ma, Running Zhao, Xinhua Liu, Hailan Kuang, and Mohammed A. A. Al-Qaness. Classification of human motions using micro-Doppler radar in the environments with micro-motion interference. *Sensors*, vol 19, 2019.

[73] Sherif Abdulatif, Qian Wei, Fady Aziz, Bernhard Kleiner, and Urs Schneider. Micro-doppler based human-robot classification using ensemble and deep learning approaches. In *2018 IEEE Radar Conference (RadarConf18)*. IEEE, 2018.

[74] Youngwook Kim and Hao Ling. Human activity classification based on micro-doppler signatures using a support vector machine. *IEEE T. Geoscience and Remote Sensing*, vol 47:p. 1328–1337, 2009.

[75] Matthew Ritchie, Francesco Fioranelli, Hugh Griffiths, and Brge Torvik. Monostatic and bistatic radar measurements of birds and micro-drone. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–5, 2016.

[76] S. Rahman and D.A. Robertson. Millimeter-wave micro-Doppler measurements of small UAVs. In *Radar Sensor Technology XXI*, pages 307 – 315, 2017.

[77] R. I. A. Harmanny, J. J. M. de Wit, and G. Prémel Cabic. Radar micro-doppler feature extraction using the spectrogram and the cepstrogram. In *2014 11th European Radar Conference*, pages 165–168, 2014.

[78] Ino Choi, Ki-Bong Kang, Joo-Ho Jung, and Sang-Hong Park. Efficient estimation of the helicopter blade parameter by independent component analysis. *IEEE Access*, Vol 8, 2020.

[79] Ashish Kumar Singh and Yong-Hoon Kim. Automatic measurement of blade length and rotation rate of drone using W-Band micro-Doppler radar. *IEEE Sensors Journal*, vol 18:p. 1895–1902, 2018.

[80] J. J. M. de Wit, R. I. A. Harmanny, and P. Molchanov. Radar micro-Doppler feature extraction using the singular value decomposition. In *2014 International Radar Conference*, pages 1–6, 2014.

[81] J. Markow and A. Balleri. Examination of Drone Micro-Doppler and JEM/HERM Signatures. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–6, 2020.

[82] A. Huang, P. Sevigny, B. Balaji, and S. Rajan. Fundamental Frequency Estimation of HERM Lines of Drones. In *2020 IEEE International Radar Conference (RADAR)*, pages 1013–1018. IEEE, 2020.

[83] Peter Klaer, Andi Huang, Pascale Sévigny, Sreeraman Rajan, Shashank Pant, Prakash Patnaik, and Bhashyam Balaji. An investigation of rotary drone HERM line spectrum under manoeuvering conditions. *Sensors*, vol 20, 2020.

[84] D. White, M. Jahangir, M. Antoniou, C.J. Baker, J. Thiyagalingam, and S. Harman. Multi-rotor drone micro-Doppler simulation validation with genuine motor speeds and L-band staring radar. In *2022 IEEE Radar Conference (RadarConf22)*, 2022.

[85] F.J. Harris. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, vol 66:p. 51–83, 1978.

[86] MathWorks. Fast Fourier transform - MATLAB fft. `https://www.mathworks.com/help/matlab/ref/fft.html`. Accessed: 2022-08-02.

[87] FFTW. FFTW homepage. `https://fftw.org/`. Accessed: 2022-08-02.

[88] Piers Beasley, Matthew Ritchie, Hugh Griffiths, William Miceli, Michael Inggs, Simon Lewis, and Bradley Kahn. Multistatic radar measurements of UAVs at X-band and L-band. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–6, 2020.

[89] Plextek. Plextek. `https://www.plextek.com`. Accessed: 2022-08-01.

[90] Peter Wellig, Peter Speirs, Christof Schuepbach, Roland Oechslin, Matthias Renker, Urs Boeniger, and Hans Pratisto. Radar systems and challenges for C-UAV. In *2018 19th International Radar Symposium (IRS)*, pages 1–8, 2018.

[91] Pierre Poitevin, Michel Pelletier, and Patrick Lamontagne. Challenges in detecting UAS with radar. In *2017 International Carnahan Conference on Security Technology (ICCST)*, 2017.

[92] DJI. DJI inspire 1 product information. `https://www.dji.com/inspire-1/info`. Accessed: 2022-08-03.

[93] M. Skolnik. Improvements for air surveillance radar. *1999 IEEE Radar Conference*, 1999.

[94] IEEE. IEEE standard for radar definitions. *IEEE Std 686-2017 (Revision of IEEE Std 686-2008)*, pages 1–54, 2017.

[95] M. Skolnik. Attributes of the ubiquitous phased array radar. *2003 IEEE International Conference on Phased Array Systems and Technology*, 2003.

[96] Mohammed Jahangir and Chris Baker. Robust detection of micro-UAS drones with L-Band 3-D holographic radar. In *2016 Sensor Signal Processing for Defence (SSPD)*, 2016.

[97] Mohammed Jahangir and Chris J Baker. Extended dwell doppler characteristics of birds and micro-UAS at L-band. In *2017 18th International Radar Symposium (IRS)*, 2017.

[98] 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*, 2015.

[99] S. Harman and A. Hume. Applications of staring surveillance radars. *IEEE National Radar Conference - Proceedings*, 2015.

[100] Aveillant. Aveillant ltd. `https://www.aveillant.com`. Accessed: 2022-08-01.

[101] MathWorks. predictorimportance. `https://www.mathworks.com/help/stats/compactclassificationtree.predictorimportance.html`. Accessed: 2023-05-16.

[102] Ana Stroescu, Mikhail Cherniakov, and Marina Gashinova. Classification of high resolution automotive radar imagery for autonomous driving based on deep neural networks. In *2019 20th International Radar Symposium (IRS)*, pages 1–10, 2019.

[103] Mehmet Saygın Seyfioğlu and Sevgi Zübeyde Gürbüz. Deep neural network initialization methods for micro-Doppler classification with low training sample support. *IEEE Geoscience and Remote Sensing Letters*, vol 14:p. 2462–2466, 2017.

[104] Youngwook Kim and Taesup Moon. Human detection and activity classification based on micro-doppler signatures using deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, vol 13:p. 8–12, 2016.

[105] Yue Lang, Qing Wang, Yang Yang, Chunping Hou, Yuan He, and Xu Jinchen.

Person identification with limited training data using radar micro-Doppler signatures. *Microwave and Optical Technology Letters*, 62, 2020.

[106] S. Franceschini, M. Ambrosanio, S. Vitale, F. Baselice, A. Gifuni, G. Grassini, and V. Pascazio. Hand gesture recognition via radar sensors and convolutional neural networks. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–5, 2020.

[107] Warren Mcculloch and Walter Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, vol 5:p. 127–147, 1943.

[108] M.A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[109] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, vol 86:p. 2278–2324, 1998.

[110] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[111] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

[112] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, vol 15:p. 1929–1958, 2014.

[113] Samiur Rahman and Duncan A. Robertson. Classification of drones and birds using convolutional neural networks applied to radar micro-Doppler spectrogram images. *IET Radar, Sonar & Navigation*, vol 14:p. 653–661, 2020.

[114] J. Park, R. Javier, T. Moon, and Y. Kim. Micro-Doppler Based Classification of Human Aquatic Activities via Transfer Learning of Convolutional Neural Networks. *Sensors*, vol 16:p. 1990, 2016.

[115] Li Fei-Fei, Jia Deng, Olga Russakovsky, Alex Berg, and Kai Li. Imagenet. `https://www.image-net.org/index.php`. Accessed: 2022-09-01.

[116] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge (ILSVRC). `https://www.image-net.org/challenges/LSVRC/`. Accessed: 2022-06-03.

[117] Md. Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C. Van Essen, Abdul A. S. Awwal, and Vijayan K. Asari. The history began from AlexNet: A comprehensive survey on deep learning approaches. *CoRR*, abs/1803.01164, 2018.

[118] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv*, 2015.

[119] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview, 2020.

[120] Zeus Gannon and David Tahmoush. Measuring UAV propeller length using micro-doppler signatures. In *2020 IEEE International Radar Conference (RADAR)*, pages 1019–1022, 2020.

[121] P. Molchanov, R.I.A. Harmanny, J.M. 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:p. 435–444, 2014.

[122] L. Fuhrmann, O. Biallawons, J. Klare, R. Panhuber, R. Klenke, and J. Ender. Micro-Doppler analysis and classification of UAVs at Ka band. In *2017 18th International Radar Symposium (IRS)*, pages 1–9, 2017.

[123] Xin Guo, Chea Siang Ng, Erwin de Jong, and Adriaan B. Smits. Micro-doppler based mini-UAV detection with low-cost distributed radar in dense urban environment. In *2019 16th European Radar Conference (EuRAD)*, pages 189–192, 2019.

[124] Marco Messina and G. Pinelli. Classification of drones with a surveillance radar signal. In *International Conference on Computer Vision Systems (ICVS 2019)*, 2019.

[125] Samiur Rahman and Duncan Alexander Robertson. Millimeter-wave radar micro-Doppler feature extraction of consumer drones and birds for target discrimination. In *Radar Sensor Technology XXIII*, 2019.

[126] Beom-Seok Oh, Xin Guo, and Zhiping Lin. A UAV classification system based on FMCW radar micro-Doppler signature analysis. *Expert Systems with Applications*, vol 132:p. 239–255, 2019.

[127] Borge Torvik, Karl Olsen, and Hugh Griffiths. Classification of birds and UAVs based on radar polarimetry. *IEEE Geoscience and Remote Sensing Letters*, vol 13:p. 1–5, 2016.

[128] Nima Mohajerin, Jonathan Histon, Reza Dizaji, and Steven Waslander. Feature extraction and radar track classification for detecting UAVs in civilian airspace. In *IEEE National Radar Conference - Proceedings*, pages 0674–0679, 2014.

[129] Mohammad Jahangir, Bashar Ahmad, and Chris Baker. Robust drone classification using two-stage decision trees and results from SESAR SAFIR trials. In *2020 IEEE International Radar Conference (RADAR)*, pages 636–641, 2020.

[130] Cameron Bennett, Mohammad Jahangir, Francesco Fioranelli, Bashar I Ahmad, and Julien Le Kernec. Use of symmetrical peak extraction in drone micro-doppler classification for staring radar. In *2020 IEEE Radar Conference (RadarConf20)*, 2020.

[131] P. Zhang, L. Yang, G. Chen, and G. Li. Classification of drones based on micro-Doppler signatures with dual-band radar sensors. In *2017 Progress in Electromagnetics Research Symposium - Fall (PIERS - FALL)*, pages 638–643, November 2017.

[132] Wenyu Zhang and Gang Li. Detection of multiple micro-drones via cadence velocity diagram analysis. *Electronics Letters*, vol 54, 2018.

[133] Yefeng Cai, Oleg Krasnov, and Alexander Yarovoy. Radar recognition of multi-propeller drones using micro-doppler linear spectra. In *2019 16th European Radar Conference (EuRAD)*, pages 185–188, 2019.

[134] Lasse Lehmann and Jørgen Dall. Simulation-based approach to classification of airborne drones. In *2020 IEEE Radar Conference (RadarConf20)*, 2020.

[135] Nir Regev, Ilia Yoffe, and Dov Wulich. Classification of single and multi propelled miniature drones using multilayer perceptron artificial neural network. In *International Conference on Radar Systems (Radar 2017)*, pages 1–5, 2017.

[136] F. Fioranelli, M. Ritchie, H. Griffiths, and H. Borrion. Classification of loaded/unloaded micro-drones using multistatic radar. *Electronics Letters*, Vol 51:p. 1813–1815, 2015.

[137] 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 & Navigation*, vol 11:p. 116–124, 2017.

[138] L. Pallotta, C. Clemente, A. Raddi, and G. Giunta. A Feature-Based Approach for Loaded/Unloaded Drones Classification Exploiting micro-Doppler Signatures. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–6. IEEE, 2020.

[139] Stamatios Samaras, Vasileios Magoulianitis, Anastasios Dimou, Dimitrios Zarpalas, and Petros Daras. UAV classification with deep learning using surveillance radar data. In *Computer Vision Systems*, pages 744–753, 2019.

[140] Alexander Karlsson, Magnus Jansson, and Mikael Hamalainen. Model-aided drone classification using convolutional neural networks. In *2022 IEEE Radar Conference (RadarConf22)*, pages 1–6, 2022.

[141] G.J. Mendis, T. Randeny, J. Wei, and A. Madanayake. Deep learning based doppler radar for micro UAS detection and classification. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 924–929. IEEE, November 2016.

[142] Daniel Brooks, Frédéric Barbaresco, Yani Ziani, Jean-Yves Schneider, and Claude Adnet. Drone recognition by micro-doppler and kinematic. In *2020 17th European Radar Conference (EuRAD)*, pages 42–45, 2021.

[143] Engin Kaya and Gulay Buyukaksoy Kaplan. Neural network based drone recognition techniques with non-coherent S-Band radar. In *2021 IEEE Radar Conference (RadarConf21)*, pages 1–6, 2021.

[144] Daniel A. Brooks, Olivier Schwander, Frederic Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Temporal deep learning for drone micro-doppler classification. In *2018 19th International Radar Symposium (IRS)*, pages 1–10, 2018.

[145] Junhyeong Park, Jun-Sung Park, and Seong-Ook Park. Small drone classifi-

cation with light CNN and new micro-Doppler signature extraction method based on A-SPC technique. *arXiv*, 2020.

[146] Byung Kwan Kim, Hyun-Seong Kang, and Seong-Ook Park. Drone classification using convolutional neural networks with merged doppler images. *IEEE Geoscience and Remote Sensing Letters*, vol 14:p. 38–42, 2017.

[147] Byung Kwan Kim, Hyun-Seong Kang, Seongwook Lee, and Seong-Ook Park. Improved drone classification using polarimetric merged-doppler images. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020.

[148] Albert Huizing, Matijs Heiligers, Bastiaan Dekker, Jacco de Wit, Lorenzo Cifola, and Ronny Harmanny. Deep learning for classification of mini-UAVs using micro-Doppler spectrograms in cognitive radar. *IEEE Aerospace and Electronic Systems Magazine*, vol 34:p. 46–56, 2019.

[149] Gihan J. Mendis, Jin Wei, and Arjuna Madanayake. Deep learning cognitive radar for micro uas detection and classification. In *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*, pages 1–5, 2017.

[150] Byunggil Choi and Daegun Oh. Classification of drone type using deep convolutional neural networks based on micro- doppler simulation. In *2018 International Symposium on Antennas and Propagation (ISAP)*, pages 1–2, 2018.

[151] Frédéric Barbaresco, Daniel Brooks, and Claude Adnet. Machine and deep learning for drone radar recognition by micro-doppler and kinematic criteria. In *2020 IEEE Radar Conference (RadarConf20)*, pages 1–6, 2020.

[152] Samiur Rahman, Duncan A. Robertson, and Mark A. Govoni. Radar signatures of drones equipped with heavy payloads and dynamic payloads generating inertial forces. *IEEE Access*, vol 8, 2020.

[153] Carmine Clemente, Luca Pallotta, Christos Ilioudis, Francesco Fioranelli, Gaetano Giunta, and Alfonso Farina. Chebychev moments based drone classification, recognition and fingerprinting. In *2021 21st International Radar Symposium (IRS)*, 2021.

[154] Samiur Rahman and Duncan Robertson. FAROS-E: a compact and low-cost millimeter wave surveillance radar for real time drone detection and classification. In *2021 21st International Radar Symposium (IRS)*, 2021.

[155] Divy Raval, Emily Hunter, Ian Lam, Sreeraman Rajan, Anthony Damini, and Bhashyam Balaji. Detecting drones with radars and convolutional networks based on micro-Doppler signatures. In *2022 IEEE Radar Conference (RadarConf22)*, 2022.

[156] H. Visvanathan Sethuraman, A. Yarovoy, and F. Fioranelli. Classification of unmanned aerial vehicles (UAVs) carrying payloads with polarimetric radar. In *2021 18th European Radar Conference (EuRAD)*, pages 365–368, 2022.

[157] J.S. Patel, F. Fioranelli, and D. Anderson. Review of radar classification & RCS characterisation techniques for small UAVs or drones. *IET Radar, Sonar and Navigation*, 2018.

[158] B Taha and A Shoufan. Machine learning-based drone detection and classification: State-of-the-art in research. *IEEE Access*, 2019.

[159] Zahra Nazari, Masoom Nazari, Mir Sayed Shah, and Dongshik Kang. Evaluation of class noise impact on performance of machine learning algorithms. In *IJCSNS - International Journal of Computer Science and Network Security*, 2018.

[160] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, vol 22:p. 177–210, 2004.

[161] Jacob Sim, Mohammad Jahangir, Francesco Fioranelli, Chris J Baker, and Holly Dale. Effective ground-truthing of supervised machine learning for drone classification. In *2019 International Radar Conference (RADAR)*, pages 1–5. IEEE, 2019.

[162] Mathworks. Fitctree. `https://uk.mathworks.com/help/stats/fitctree.html`. Accessed: 2021-09-14.

[163] J. R. Quinlan. Learning from noisy data. In *Proceedings of the Second International Machine Learning Workshop*, pages 81–106, 1986.

[164] J. R. Quinlan. Induction of decision trees. *Machine Learning*, vol 1:p. 81–106, 1986.

[165] Enrico Ajanic, Mir Feroskhan, Stefano Mintchev, Flavio Noca, and Dario Floreano. Bioinspired wing and tail morphing extends drone flight capabilities. *Science Robotics*, Vol 5, 2020.

[166] MathWorks. highpass. `https://www.mathworks.com/help/signal/ref/highpass.html`. Accessed: 2023-05-18.

[167] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, Vol 13:p.281–305, 2012.

[168] J. Močkus. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404. Springer, 1975.

[169] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, vol 13:p. 455–492, 1998.

[170] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

[171] Alex Krizhevsky. CIFAR-10 and CIFAR-100 datasets. `https://www.cs.toronto.edu/~kriz/cifar.html`. Accessed: 2022-06-03.

[172] Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, vol 17:p. 26–40, 2019.

[173] Yann LeCun, Coinna Cortes, and Christopher J.C. Burges. The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`. Accessed: 2022-06-03.

[174] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. *arXiv*, 2016.

[175] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014.

[176] Hyun Min Oh, Hyunki Lee, and Min Young Kim. Comparing convolutional neural network (CNN) models for machine learning-based drone and bird classification of anti-drone system. In *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, pages 87–90, 2019.

[177] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, vol 6:p. 60, 2019.

[178] S. Bjorklund and N. Wadstromer. Target Detection and Classification of Small

Drones by Deep Learning on Radar Micro-Doppler. In *2019 International Radar Conference (RADAR)*, 2019.

[179] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, vol 45:p. 427–437, 2009.

[180] Marek Ciesielski, Krzysztof Stasiak, Mariia Khyzhniak, Konrad Jedrzejewski, Marcin Żywek, and Sebastian Brawata. Simultaneous signal processing with multiple coherent processing intervals in fmcw radar for drone detection. In *2021 21st International Radar Symposium (IRS)*, 2021.

[181] Julien Gérard, Joanna Tomasik, Christèle Morisseau, Arpad Rimmel, and Gilles Vieillard. Micro-Doppler signal representation for drone classification by deep learning. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1561–1565, 2021.

[182] D.B. Leeson and G.F. Johnson. Short-term stability for a doppler radar: Requirements, measurements, and techniques. *Proceedings of the IEEE*, 54(2):244–248, 1966.

[183] Enrico Rubiola. *Phase noise and frequency stability*, page 1–34. The Cambridge RF and Microwave Engineering Series. Cambridge University Press, 2008.

[184] University of Birmingham. Live trials at airport explore bird movement in urban environments. `https://www.birmingham.ac.uk/university/colleges/eps/news/2021/live-trials-at-airport-explore-bird-movement-in-urban-environments.aspx`. Accessed: 2021-08-31.

[185] Z. Qin, F. Yu, C. Liu, and X. Chen. How convolutional neural networks see the world — a survey of convolutional neural network visualization methods. *Mathematical Foundations of Computing*, vol 1:p. 149–180, 2018.

[186] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[187] Mathworks. Visualize Activations of a Convolutional Neural Network - MATLAB & Simulink - MathWorks United Kingdom. `https://uk.mathworks.com/help/deeplearning/ug/visualize-activations-of-a-convolutional-neural-network.html`. Accessed: 2021-09-13.

[188] Dymitr Ruta and Bogdan Gabrys. An overview of classifier fusion methods. *Computing and Information Systems*, vol 7, 2000.

[189] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 20:p. 226–239, 1998.

[190] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *CoRR*, abs/1706.04599, 2017.

[191] Tin Kam Ho, J.J. Hull, and S.N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 16(1):p. 66–75, 1994.

[192] Mathworks. Fitclinear. `https://uk.mathworks.com/help/stats/fitclinear.html`. Accessed: 2021-09-14.

[193] Tim Pearce, Alexandra Brintrup, and Jun Zhu. Understanding softmax confidence and uncertainty. *CoRR*, abs/2106.04972, 2021.

[194] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136, 2016.

[195] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33rd*

International Conference on International Conference on Machine Learning, pages 1050–1059, 2016.

[196] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[197] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv 1412.6572*, 2015.

[198] Simon Wagner, Chandana Panati, and Stefan Brüggenwirth. Fool the COOL - on the robustness of deep learning SAR ATR systems. In *2021 IEEE Radar Conference (RadarConf21)*, 2021.

[199] Stephen Harman and Bashar I. Ahmad. The need for simultaneous tracking and recognition in drone surveillance radar. In *2021 21st International Radar Symposium (IRS)*, 2021.