INSTITUTO
UNIVERSITÁRIO
DE LISBOA

RF-Based Drone Detection using Deep Learning approaches

Miguel Alexandre Moreira Romana

Master's Degree in
Telecommunications and Computer Engineering

Supervisor:
PhD Nuno Manuel Branco Souto, Associate Professor (with Habilitation),
Iscte – University Institute of Lisbon

Co-Supervisor:
PhD Renato Branco Ferreira, Invited Assistant Researcher,
Iscte – University Institute of Lisbon

December, 2025

TECHNOLOGY
AND ARCHITECTURE

Department of Information Science and Technology

RF-Based Drone Detection using Deep Learning approaches

Miguel Alexandre Moreira Romana

Master's Degree in
Telecommunications and Computer Engineering

Supervisor:
PhD Nuno Manuel Branco Souto, Associate Professor (with Habilitation),
Iscte – University Institute of Lisbon

Co-Supervisor:
PhD Renato Branco Ferreira, Invited Assistant Researcher,
Iscte – University Institute of Lisbon

December, 2025

‘‘We can only see a short distance ahead, but
we can see plenty there that needs to be done.’’

Alan Turing

Acknowledgements

First of all, I would like to express my sincerest gratitude for my **supervisor**, Professor Nuno Souto, for his continuous guidance, dedication, patience, respect, and all the knowledge shared throughout this dissertation and within my time at university. His assistance was paramount in helping me realise this academic achievement, and it was an honour to collaborate with and learn from such an outstanding person and educator. I would also like to formally acknowledge Professor Renato Ferreira for the support given in his role as my **co-supervisor**. I also wish to extend my gratitude towards my **university** (ISCTE – University Institute of Lisbon) for allowing me to grow both academically and personally through many opportunities and experiences; and also for IT (Instituto de Telecomunicações), for providing the necessary resources essential to the development of this work.

I am delighted to express my deepest appreciation to **all my dear friends** and astounding colleagues for their friendship, motivation, and unwavering support; their encouragement has driven me to pursue my academic career with commitment and perseverance. I am eternally grateful to have grown personally alongside them over these last few years and throughout my life.

In particular, I am profoundly fortunate for the friends I met at university, who have been instrumental to my success, resilience, and well-being, and whom I hold in high regard. I would especially like to thank Tiago Almeida, Duarte Casaleiro, Rodrigo Guerreiro, Oleksandr Kobelyuk, Filipe Gonçalves, Henrique de Sousa, Pedro Arsénio, and Margarida Neves, as well as Gonçalo Benido, Mara Alves, António Crossas, and Roberto Miguel, among several others.

I am also equally privileged for the friends I have had the honour of accompanying along my academic journey and life, who have also been fundamental in contributing to the person I am today. I am greatly thankful for Pedro Santana, André Cardoso, Afonso Gaspar, Bernardo Ramalhete, Filipe Lopes, Pedro Souto, and Ricardo Pinto, along with Sofia Morais and numerous others.

I wish to offer a special thanks to my **family**, particularly my brother and my parents, for their enduring love, support, and understanding through these challenging times and across my life.

Finally, I would like to honour the memory of those who are sadly no longer with us, as a way to immortalise their impact, and forever cherish the memories made by their presence.

Resumo

Drones tornaram-se amplamente utilizados em múltiplos domínios, e incidentes associados têm aumentado e intensificado as preocupações quanto a segurança e privacidade, motivando consequentemente o desenvolvimento de soluções precisas, eficientes e fiáveis capazes de detetar e identificar Veículos Aéreos Não Tripulados (UAV).

Este trabalho propõe duas arquiteturas de Redes Neuronais (NN): uma Rede Convolucional Neuronal (CNN) com camadas convolucionais unidimensionais (1-D); e uma Rede Neuronal Recorrente (RNN) com unidades Memória Longa de Curto-Prazo (LSTM). O sistema recorre a uma base de dados existente composta por sinais de drones de Rádio Frequência (RF), para classificação de UAV.

Várias experiências foram investigadas, focando no impacto de fatores de dilatação, convoluções separáveis em profundidade e camadas recorrentes para deteção; e na incorporação da taxa de dilatação para identificação. O desempenho do sistema é avaliado mediante matrizes de confusão e métricas tradicionais, incluindo exatidão, precisão, *recall* e *F1-score*.

Os resultados demonstram que as experiências que usam fatores de dilatação e convoluções separáveis em profundidade alcançam deteção quase-perfeita de drones; enquanto abordagens que integram camadas recorrentes, e taxa de dilatação para identificação, apresentam capacidade limitada, possivelmente devido à menor complexidade de arquitetura e também à semelhança de sinais entre drones do mesmo fabricante.

Em síntese, as observações indicam assim que a combinação de sinais RF com métodos de Aprendizagem Profunda (DL) constitui uma abordagem eficaz para o desenvolvimento de sistemas fiáveis de deteção e identificação de drones.

Palavras-chave: Deteção de drone, Identificação de UAV, Aprendizagem Profunda, Redes Neuronais, Sinais RF.

Abstract

Drones have become widely adopted across numerous domains, and associated incidents have increased security and privacy concerns, consequently motivating the development of accurate, efficient, and reliable solutions capable of detecting and identifying Unmanned Aerial Vehicles (UAV).

This work proposes two Neural Network (NN) architectures: a Convolutional Neural Network (CNN) with one-dimensional (1-D) convolutional layers; and a Recurrent Neural Network (RNN) using Long Short-Term Memory (LSTM) units. The system leverages an existing database composed of recorded drone Radio Frequency (RF) signals for the classification of UAV.

Several experiments were investigated, focusing on the impact of dilation factors, depthwise separable convolutions, and recurrent layers for detection; and the incorporation of dilation rate for identification. The system performance is evaluated through confusion matrices and standard metrics, including accuracy, precision, recall, and F1-score.

Results demonstrate that the experiments using dilation factors and depthwise separable convolutions configurations achieve near-perfect detection of drones; whereas approaches integrating recurrent layers, and dilation rate for identification, exhibit limited capability, potentially due to lower architecture complexity and signal similarity between drones of the same manufacturer.

Ultimately, observations indicate that combining RF signals with Deep Learning (DL) methods constitutes an effective approach for the development of reliable systems for drone detection and identification.

Keywords: Drone detection, UAV identification, Deep Learning, Neural Networks, RF signals.

Contents

List of Figures

List of Tables

List of Acronyms

1-D	One-Dimensional
ADC	Analogue-to-Digital Converter
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CRNN	Convolutional Recurrent Neural Network
DDI	Drone Detection and Identification
DFT	Discrete Fourier Transform
DL	Deep Learning
DNN	Deep Neural Network
DSC	Depthwise Separable Convolution
DSR	Design Science Research
FDR	False Discovery Rate
FNR	False Negative Rate
FN	False Negative
FP	False Positive
FFT	Fast Fourier Transform
FHSS	Frequency-Hopping Spread Spectrum
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRC	GNU Radio Companion
ISM	Industrial, Scientific, and Medical
LSTM	Long Short-Term Memory

ML	Machine Learning
NN	Neural Network
OFDM	Orthogonal Frequency-Division Multiplexing
ReLU	Rectified Linear Unit
RF	Radio Frequency
RNN	Recurrent Neural Network
SDR	Software-Defined Radio
SNR	Signal-to-Noise Ratio
Tanh	Hyperbolic Tangent
TCN	Temporal Convolutional Network
TN	True Negative
TP	True Positive
UAV	Unmanned Aerial Vehicle

CHAPTER 1

Introduction

This chapter introduces this thesis' motivation, objectives, research methodology, and structure.

1.1. Context and Motivation

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are aircraft that operate without a human pilot on board. They can be remotely controlled by an operator, or function autonomously following pre-programmed flight paths, utilising onboard sensors along with advanced technology for navigation, control and perception [1], [2].

Although initially implemented for military applications, drones have become integral to modern technology, and are now widely used in various civilian and commercial sectors, ranging in applications from aerial cinematography and parcel delivery to environmental monitoring and emergency response operations, among a plethora of others [1], [3], [4], [5], [6], [7].

However, the growing accessibility and proliferation of drones has raised concerns for security and privacy. Unauthorised drones can be used for malicious activities, especially in restricted areas, for instance to interfere with air traffic, or violate personal privacy via surveillance [8], [9], [10].

Tragic episodes have highlighted the dangers posed by unauthorised drones, particularly airport disruptions caused by rogue UAVs, which have endangered lives and resulted in significant financial losses [8]. A notable example is the Gatwick Airport drone incident of 2018, in which the airport runway had to be closed in order to prevent the risk of collision with aircraft [11], [12]; moreover, drone usage has extended from civilian disruptions to active warfare, as evidenced in the Russia–Ukraine conflict as of 2025, where UAVs have become pivotal offensive and surveillance tools which have redefined strategic operations on the battlefield [13], [14].

To address these challenges, there is a pressing need to develop Drone Detection and Identification (DDI) systems that can trigger and apply appropriate countermeasures to unauthorised drones in real-time when necessary, ensuring safe and responsible integration of drones into the airspace.

As Khan *et al.* emphasise in [1, p. 11451], ‘‘to protect people, assets and critical infrastructure from…unauthorized…drones, an efficient and reliable drone detection method is urgently needed’’.

Various solutions have been developed to counter drones, including physical strategies such as missiles, nets, or trained birds, as well as electromagnetic methods, especially jamming and spoofing. The latter are favoured for their lower cost and risk [15], [16]. Notably, Ferreira and Gaspar *et al.* [17] developed a mobile Software-Defined Radio (SDR)-based platform with both jamming and spoofing capabilities. Similar strategies have been examined in other studies, demonstrating the sustained significance of these methods [15], [16], [18], [19], [20], [21].

Furthermore, accurate and reliable detection of drones is a fundamental prerequisite for the selective and effective deferred application of countermeasures. In this regard, Gonçalves [22] and Soeira [23] proposed more traditional approaches concerning the detection of the presence of UAVs, by analysing the Radio Frequency (RF) spectrum using SDRs and considering the signal power to estimate the direction of arrival, and the operating frequency band, respectively.

This dissertation seeks to advance the development and discovery of technologies and techniques supporting anti-drone systems and contribute to the ongoing collaborative efforts of [17], [22], [23].

Moreover, drone detection approaches employing Deep Learning (DL) strategies have shown encouraging results recognising RF signal patterns [24], [25], [26], [27], [28], [29], providing a strong motivation for the direction pursued in this research.

This thesis will address the study of reliable and robust real-time solutions for detecting UAVs; these solutions combine SDR-based spectrum monitoring and advanced DL techniques to identify the presence of drones and potentially their manufacturers and models. The resulting system aims to facilitate the activation of appropriate anti-drone procedures at a subsequent stage.

1.2. Objectives

The main objective of the proposed research is to design and implement a reliable system capable of detecting the presence of unauthorised UAVs by identifying their communication signals within the RF spectrum. The system will also aim to classify the make and model of detected UAVs, through the analysis of their spectral signatures.

This system will combine SDR-based spectrum monitoring with advanced DL techniques, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to ensure robust detection and classification of drone communication signals. The training process for the CNNs and RNNs will leverage existing drone RF-based datasets, which will be further refined and validated using real-world data from commercial UAVs.

Hence, this dissertation will address the following research questions:

RQ1 How can SDR and DL techniques be combined to develop a reliable system for detecting and identifying UAVs?

RQ2 What can be an effective strategy to optimise DL techniques for classifying drone communication signals?

1.3. Research Methodology

The Design Science Research (DSR) methodology is a valuable approach for conducting research by defining a problem that needs to be addressed, developing and evaluating new solutions, and communicating the results.

Fig. 1.1 demonstrates the process model presented by Peffers *et al.* in [30].

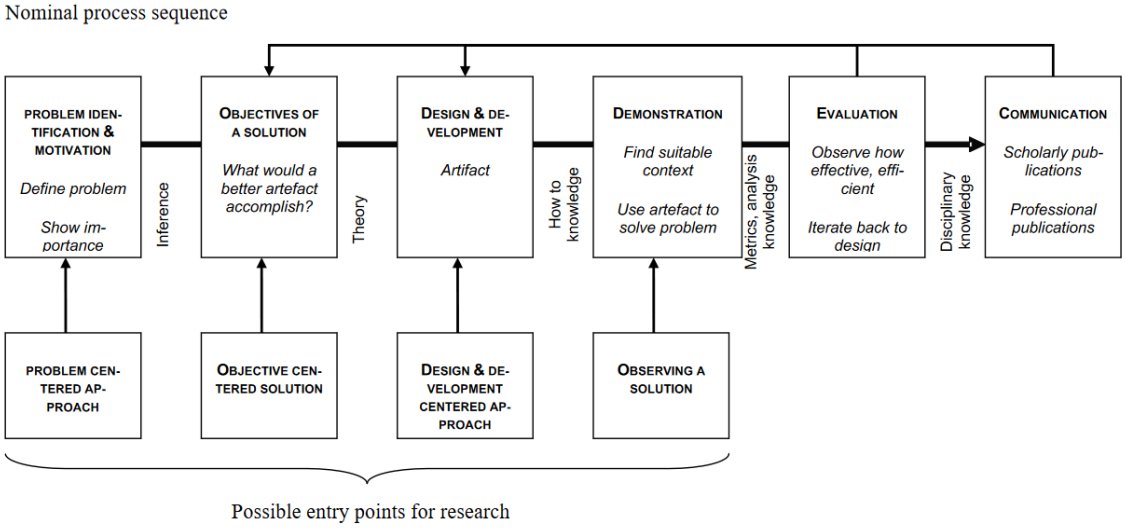


Fig. 1.1. Design Science Research process model. From [30, Fig. 1].

In this first chapter of the dissertation, the first and second activities are presented by defining the problem to be solved or the opportunity to improve upon, and stating the objectives and research questions to be addressed, respectively. For the following three activities, namely design and development, demonstration, and evaluation, the artefact used to solve the problem is created, indicating its architecture and functionality, and experiments and analysis are performed to prove and assess its effectiveness. The final activity is communication, which is accomplished through work publications such as theses and journal articles, for instance.

As highlighted in the figure (Fig. 1.1), the DSR represents an iterative process, such that it is possible to iterate on previous activities and develop new and different solutions. Furthermore, since there are four possible beginnings, researchers are neither required nor expected to begin with the first activity of problem identification and motivation. Nevertheless, as this dissertation follows a problem-centred approach, it will begin with the problem definition and continue in the sequential order in which it is presented.

1.4. Dissertation Structure

This section provides an overview of this thesis structure.

Chapter 1 provides a brief introduction of the context and motivation of this thesis, along with the adopted research methodology, the considered objectives and research questions, as well as the overall structure of this dissertation.

Chapter 2 presents a review of the state of the art regarding UAV detection and countermeasure systems, SDR platforms and software, and also DL techniques used to accomplish the objectives of this work, as well as related literature.

Chapter 3 describes the methodology considered to conduct the experiments and discusses the details of the equipment utilised, the dataset used, the proposed models and design, and the evaluation metrics employed.

Chapter 4 discloses the results achieved by analysis of the simulations, alongside the approaches and configurations adopted for the various experiments.

Chapter 5 presents the conclusions of this work and suggests further improvements and alternatives to be implemented.

CHAPTER 2

State of the Art

This chapter provides a review to conduct the research of the state of the art concerning drone detection and countermeasure strategies, a comparison of several SDR platforms and supported software, and advanced DL techniques used to develop and accomplish the objectives of this thesis, as well as related work.

2.1. Drone Communications

Drones, also known as UAVs, are aircraft that operate without a human pilot on board, which can function either autonomously or under remote control. These devices have been deployed across a wide range of applications in various sectors, including: aerial photography, delivery services, environmental monitoring, emergency response, and many more [1], [2], [3], [4], [5], [6], [7].

Drones can be mainly categorised according to their aerodynamic configuration (e.g., size, shape, weight, rotors) and flight potential (e.g., speed, manoeuvrability, range, endurance), with varying capabilities and technology. Furthermore, these aircraft integrate equipment such as: a power source, an onboard flight controller, embedded processors, multiple sensors (e.g., navigation modules including gyroscopes or Global Navigation Satellite System (GNSS), such as Global Positioning System (GPS)), and a wireless transceiver. The transceiver enables communication through data exchange with a remote operator or ground control station, namely concerning command-and-control instructions (for navigation), telemetry information (e.g., position, orientation and velocity measurements), and multimedia content (i.e., images or videos) [1], [2], [15].

The communication between the drone and its controller may be established through satellite links, cellular connectivity, or RF signals. Particularly, UAVs typically utilise RF signals within the 2.4 to 5 GHz range, leveraging wireless technologies such as Wi-Fi, Bluetooth, or via cellular networks (e.g. 3G, 4G, or 5G). Furthermore, most commercial drones operate in the license-exempt Industrial, Scientific, and Medical (ISM) frequency bands of 433 MHz (433.05–434.79 MHz), 2.4 GHz (2400–2500 MHz) and 5.8 GHz (5725–5875 MHz), though some long-range drones may instead prefer frequencies in the 1.2–1.3 GHz range [1], [2], [15].

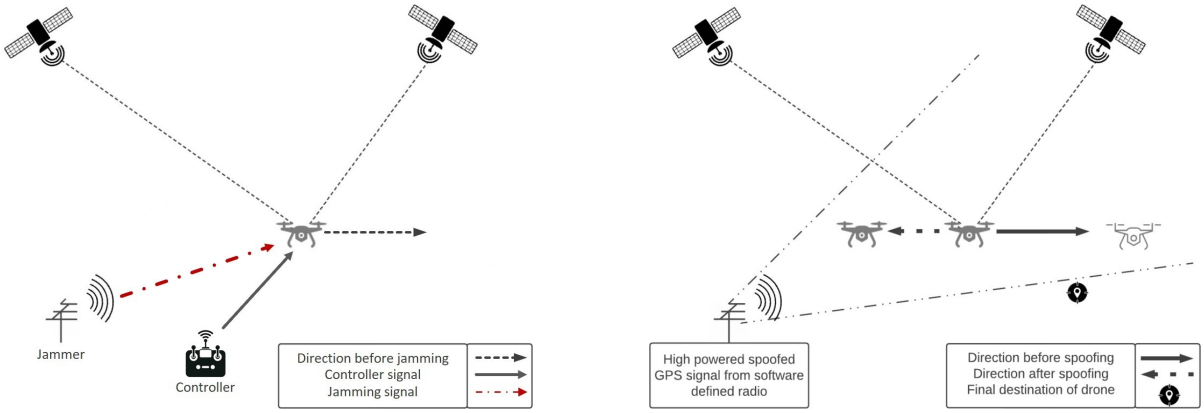
Moreover, to improve link reliability and mitigate interference, drones utilise several modulation and transmission techniques, including: Orthogonal Frequency-Division Multiplexing (OFDM), which distributes data across multiple adjacent subcarriers, to maintain signal integrity in reflective environments; and Frequency-Hopping Spread Spectrum (FHSS), which dynamically switches the carrier frequency of the signal, to reduce susceptibility to interception and jamming [2], [15], [19].

Notably, the RF signals transmitted during communication can be monitored and analysed for drone detection and classification [1], [2], [15].

2.2. Drone Countermeasures

The increasing accessibility of drones has intensified security and privacy concerns, as these devices are frequently misused in malicious activities, evidenced by numerous reported incidents, thereby posing a significant threat to public safety, and reinforcing the need for reliable and effective UAV countermeasure systems [8], [9], [13].

A variety of anti-drone strategies have been proposed and implemented, ranging from physical approaches (e.g., projectiles, trained birds, or nets) to electromagnetic methods, particularly jamming and spoofing, as illustrated in Fig. 2.1. The latter have garnered increasing attention due to their lower cost and reduced operational risks [15], [16].



(a) Jamming. Adapted from [20]. **(b)** Spoofing. From [20, Fig. 4].

Fig. 2.1. Drone Countermeasures illustration.

Jamming Floods the frequency spectrum with interference signals – such as noise – disrupting the communications between the controller and the drone, causing it to lose control or enter a fail-safe mode; depending on its programmed behaviour, it is forced to hover, land, or return to its base [19].

Spoofing Transmits counterfeit GNSS navigation signals (e.g., GPS) or false control commands, causing the controller of the drone to accept these signals as authentic; this induces erroneous position estimates, misleading it to deviate from its intended flight path, land prematurely, or navigate to an attacker-controlled location, ultimately resulting in the hijacking of the drone [20].

2.2.1. Counter-drone systems

Existing anti-drone defence systems generally integrate detection, identification, and neutralisation stages to ensure effective and selective countermeasures [16].

One example of these systems is a mobile anti-drone platform developed by Ferreira and Gaspar *et al.*, employing technology capable of both jamming and spoofing functionalities [17]. Other research has explored similar approaches, highlighting the continued importance of these techniques in anti-drone applications [15], [16], [18], [19], [20], [21].

Typically, RF-based anti-drone solutions first detect and identify the presence and the signals of the drone, subsequently implementing jamming to disrupt its communications, or spoofing to mislead its navigation. This structured approach allows for a safe and controlled neutralisation with minimal collateral risk [15].

It should be noted that the purpose of this work exclusively regards the study of reliable and robust solutions concerning the detection and identification of UAVs; therefore, the objectives of this thesis do not concern the implementation of any countermeasure nor focus on the neutralisation phase.

2.3. Drone Detection

The detection of drones constitutes a fundamental component of the neutralisation of UAVs, as the identification of the presence of drones represents the first step in this process, and it is essential towards enabling the effective application of appropriate countermeasures [15].

2.3.1. Detection methods

Various detection techniques have been developed to achieve UAV detection, with anti-drone systems employing either a single strategy or a combination of approaches [1]; these techniques include acoustic, visual, radar, and radio detection methods [1], [3], [15], [31], as shown in Fig. 2.2:

- Acoustic detection: Utilises microphone arrays to capture the sound signatures produced by drone propellers and rotors;
- Visual detection: Employs optical and infrared cameras, combined with computer vision algorithms, to detect and track drones based on their shape, size, and movement patterns;
- Radar detection: Emits radio waves and analyses their reflections to determine the location, trajectory, and velocity of a drone;
- RF-based detection: Scans the frequency spectrum to identify communication signals between the drone and its remote controller – such as command and control signals or video transmission. The system monitors the spectrum by analysing the frequency usage, signal strength, and modulation patterns.

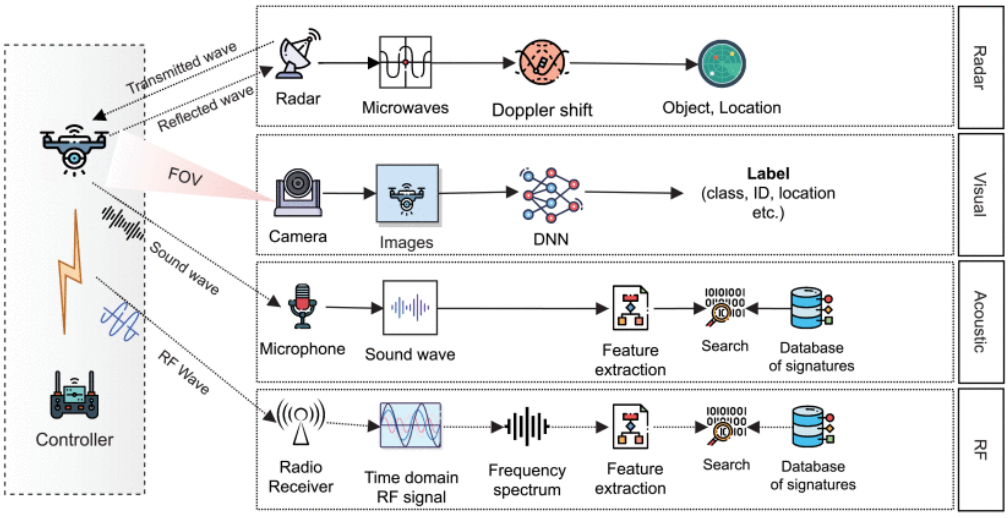


Fig. 2.2. Drone Detection approaches. From [1, Fig. 4].

Each method has advantages and limitations, influenced by its intended application, operating environment, and specific objectives and requirements, as well as by the balance between characteristics and the challenges of real-world deployment, including reliability, adaptability, scalability, performance, complexity, and cost, among a wide range of considerations [1].

Acoustic methods are sensitive to noise and range and therefore cannot accurately detect drones from a long distance or those which are equipped with nearly silent rotors or emit masking sounds – such as bird chirps or ambient noise – to conceal their acoustic signature [1]. Furthermore, these systems often rely on commercially available microphones, which provide limited coverage and high sensitivity to environmental noise [32].

Visual methods are hindered by scenarios that hide or disguise the UAVs, such as: blinding illumination or dark environments; terrain or structural obstructions; design modifications or stealth adaptations for camouflage; or adverse weather conditions, particularly storms, rain and fog [33]. Additionally, these setups also require expensive high-end optical or thermal imaging equipment [34].

Radar approaches – similarly to visual techniques – are compromised by cluttered environments, the physical characteristics of the drone, and their flight behaviour. These methods usually struggle differentiating from birds with small or composite-built drones or when they are hovering; also in complex environments, including rain, fog, snow, and dust, since the reflections from these UAVs are often attenuated, absorbed or scattered [16], [35].

Even though RF-based detection is not without vulnerabilities – such as signal interference, for example – it has proven more resistant and robust to misleading and deceptive tactics [1].

2.3.2. RF-based detection

The RF-based method shares some capabilities with the visual, acoustic, and radar detection approaches. However, this technique offers several reliable advantages, including [1], [15]:

- Versatility: Operates effectively in challenging conditions, including rain, fog, and snow, while also being capable of detecting hidden or small drones, even in environments with low visibility;
- Perception: Accurately classifies the type and purpose of the UAV by analysing its communication with the controller and effectively tracks the drone over long distances, particularly when transmitting strong signals;
- Affordability: More cost-effective to deploy than radar-based systems, as it does not require expensive infrastructure or specialised equipment;
- Scalability: Enables simultaneous detection of multiple drones and allows for easy deployment of sensors over large areas.

Furthermore, RF-based detection systems inherently operate in a passive mode, by monitoring the radio-frequency spectrum non-intrusively without emitting any signals, thereby representing a more discreet approach of surveillance. This method contrasts with radar-based detection systems, which employ an active and invasive approach [2], [36].

It is important to note that drones operating autonomously – without active communication links – or employing encrypted or frequency-hopping transmissions may evade detection by passive RF-based systems. Nevertheless, any approach must comply with privacy regulations, as unauthorised transmission or signal interference may incur legal penalties [1], [10], [16].

2.4. SDR: Software-Defined Radio

SDRs are radio communication systems that use software instead of hardware to perform signal processing operations such as tuning, modulation, and decoding. This design allows easy reconfiguration for various frequencies and protocols, reducing the need for physical adjustments [37].

SDRs play a crucial role in detecting UAVs by capturing and analysing their RF communication signals, which include command transmissions from the controller and telemetry data from the drone. Some of the advantages of these systems include [15]:

- Flexibility: Software updates allow support for several frequencies and protocols;
- Cost-effectiveness: Reduces the need for costly specialised hardware replacements;
- Real-Time Processing: Enables immediate signal analysis, demodulation, and decoding;
- Wide Range: Covers multiple frequency bands simultaneously.

2.4.1. SDR platforms

There are numerous SDR hardware options available on the market from multiple manufacturers, each designed to offer different capabilities, performance levels, and resolutions. Evaluating these platforms requires understanding several critical features, including [38]:

- Frequency range: Defines the interval of frequencies over which the board can effectively transmit and receive signals;
- RF Bandwidth: Represents the maximum range of frequencies the platform can process instantaneously, limited by its sampling rate;
- Sample rate: Indicates how frequently the Analogue-to-Digital Converter (ADC) measures or samples an incoming signal. According to the Nyquist-Shannon theorem, with I/Q sampling, the observable bandwidth is approximately equivalent to the sample rate;

- Resolution: Specifies the number of bits used in the ADC, otherwise also known as sample depth, defining how precisely the analogue signal is digitised;
- Number of Tx/Rx channels: Defines the amount of independent transmitting (Tx) and receiving (Rx) paths supported by the SDR;
- Duplex Mode: Indicates whether the board can transmit and receive simultaneously (i.e., full-duplex) or perform only one function at a time (i.e., half-duplex);
- Transmit Power: Represents the maximum output power the board can generate without external amplification.

A comparative overview of several boards and their specifications is presented in Table 2.1. In particular, the evaluated platforms are HackRF One [39], [40]; National Instruments Ettus Research USRP B210, also regarded as USRP-2901 [41], [42], [43]; ADALM-Pluto, also recognised as PlutoSDR [44], [45], [46]; bladeRF 1.0 x40 [47], [48]; and LimeSDR Mini 2.0 [49].

TABLE 2.1
SDR Platforms specifications.

Model	Frequency range (MHz)	Bandwidth (MHz)	Sample rate (MS/s)	Resolution (bits)	Tx/Rx Channels	Duplex Mode	Tx Power (dBm)	Software Support	
								GNU Radio	Simulink
HackRF One	1–6000	20	20	8	1/1	Half	15	Yes	No
USRP B210	70–6000	56	61.44	12	2/2	Full	20	Yes	Yes
ADALM-Pluto	325–3800	20	61.44	12	1/1	Full	8	Yes	Yes
BladeRF x40	300–3800	28	40	12	1/1	Full	6	Yes	No
LimeSDR Mini 2.0	10–3500	40	30.72	12	1/1	Full	10	Yes	No

2.4.2. Software tools

SDRs rely fundamentally on flexible and efficient software environments to implement signal processing operations, which traditionally required dedicated hardware [37]. For this purpose, some of the most prevalent software tools are GNU Radio and Simulink, widely used across various domains – including academia and research, for teaching and technological development, as well as industry, government, and hobbyist environments, for real-world deployment [38].

GNU Radio is a free, open-source software toolkit used to develop signal processing applications, which can be used in a pure simulation-based environment, or with external hardware to create software-defined radios [50], [51].

It provides a graphical interface called GNU Radio Companion (GRC), which enables users to build complete radio systems by combining functional blocks, each with its specific task: for sourcing or providing input data from documents, devices, or generators; processing, such as filtering, modulation, and amplification; and sink blocks for displaying, saving, or transmitting the signals, to name a few capabilities [52]. Furthermore, this tool allows developers to define and integrate custom blocks, written in C++ or Python code, enabling new functionality and enhancing its flexibility for specialised solutions [53].

As for supported SDR hardware for GNU Radio, the main platforms include USRPs, bladeRFs, LimeSDRs, ADALM-Pluto and HackRF One, along with several others [54].

Simulink – which is part of MATLAB by MathWorks – is a graphical block-diagram environment devised for multi-domain simulation and model-based design [55].

This tool provides a visual interface with a built-in editor, customisable block libraries, code generation, and solvers to assist with system design, modelling, and simulation. It also supports continuous rigorous experimentation and validation to ensure a reliable and consistent workflow with reproducible results [56]. Additionally, since Simulink is integrated with MATLAB, it allows users to implement MATLAB algorithms directly within models through the MATLAB function block, and export simulation results for further detailed data analysis and visualization [57].

It is noteworthy that there is a very limited selection of supported SDR hardware for MATLAB, which includes USRPs and ADALM-Pluto, among a few others [58].

2.5. Neural Networks

A Neural Network (NN) is a computational model inspired by the human brain, designed to recognise patterns and make predictions. It consists of an input layer and an output layer and may also have hidden layers in between [59]. As data flows through the layers, the nodes – also called neurons or perceptrons – apply mathematical functions to produce an output; during training, the neural network learns by iteratively adjusting its parameters through backpropagation, such as the weights and bias between nodes, to minimise the error and converge towards better approximations between its predictions and the correct description, effectively learning to infer more accurate assessments [60].

These networks are particularly instrumental when the task is too complex to be explicitly programmed. Instead of following a pre-defined algorithm, these networks learn autonomously, which enables them to identify non-linear relationships and hidden structures that would otherwise be too difficult or even impossible to capture through conventional methods [59], [60].

A Perceptron Artificial Neural Network (ANN) is a single-layer NN, and it is the simplest form of a neural network [59], [61], as evident in Fig. 2.3. On the other hand, a Deep Neural Network (DNN) is a type of NN with multiple hidden layers between the input and output, as illustrated in Fig. 2.4. This increased complexity allows these NNs to learn a broader range of features and capture higher levels of abstraction compared to simpler networks. This makes them particularly effective for applications regarding images and speech recognition, requiring large, multi-dimensional datasets [61].

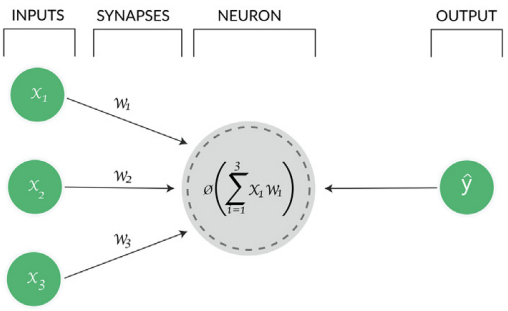


Fig. 2.3. Basic ANN architecture. From [61, Fig. 5].

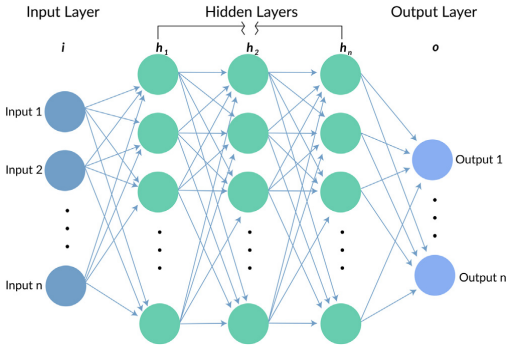


Fig. 2.4. Simple DNN architecture. From [61, Fig. 4].

2.5.1. Supervised and Unsupervised Learning

Supervised and Unsupervised Learning are two essential and widely used approaches in the fields of Machine Learning (ML) and DL [60]. These paradigms are fundamental as they shape how algorithms learn, generalise, and solve various problems in the field of Artificial Intelligence (AI) [61].

Unsupervised Learning trains a model with unlabelled data, aiming to discover underlying patterns or structures within the data without explicit guidance. Therefore, it is particularly well-suited for tasks where the correct output is unknown, such as exploratory data analysis [60].

Supervised Learning, in contrast, involves training a model on labelled data, where the input data is paired with its correct output, with the goal to predict the corresponding outcome based on the provided input features [61]. Thus, given the nature of drone detection where drones have labelled identifiers available, supervised learning proves the most adequate ML approach [60].

2.5.2. CNNs: Convolutional Neural Networks

CNNs are a type of neural network optimised for processing grid-structured data, such as images – by employing convolutional layers to automatically and efficiently detect spatial patterns and extract hierarchical features, along with other key elements [59], [60], as portrayed in Fig. 2.5. As information progresses through the many layers of the network, it initially recognises simple features, such as edges and basic shapes, gradually capturing increasingly complex patterns, such as textures, and ultimately recognising entire objects [60].

This approach is especially well-suited for tasks where the spatial configuration of elements is crucial for precise identification, such as image classification, object recognition, and computer vision applications in general [59], [60].

Notably, CNNs typically consist of multiple layers, mainly including convolutional layers, pooling layers, and fully-connected layers – as illustrated in Fig. 2.5 – and may include various other components [59], [60], [61].

- Convolutional Layer: Applies a set of filters to the input, and performs a convolution operation, resulting in feature maps that capture different aspects of the data. Furthermore, this layer is adjusted according to numerous hyperparameters, including:
  - Filters: Details the depth or amount of features extracted, and the spatial range of each filter; determined by the number of filters or kernels, and their size, respectively;
  - Stride: Represents the leap distance with which the filter shifts across the data, influencing the resolution of the output;
  - Padding: Indicates the amount of added values (often zeros) applied around the input data, ensuring specific output dimensions;
  - Dilation Rate: Defines the spacing between filter elements, enabling the network to capture a larger receptive field, without increasing kernel size.

    Fig. 3.2 illustrates the operation of the hyperparameter known as dilation factor, demonstrating its behaviour for different values.
- Activation Function: Applies a non-linear function to the feature map, enabling the network to learn complex patterns. Notably, the most common activation function employed in the field of ML is known as Rectified Linear Unit (ReLU), and it is often used due to its simplicity and efficiency [59], [60], [62].

– ReLU: Sets all negative values to zero, while preserving the positive values. This activation function is particularly useful for drone detection, since it mitigates a prevalent issue in deep neural networks known as the vanishing gradient problem – characterised by the progressive attenuation of gradient values during the backpropagation through layers – and therefore allows the network to learn efficiently from large datasets.
Mathematically, this function is defined as shown in (2.1):

$$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}, \tag{2.1}$$

or simply as $f(x) = \max(0, x)$;

- Pooling Layer: Reduces the dimensionality and number of parameters the network has to learn, whilst retaining the most important features; this results in diminished computational load and avoids overfitting;
- Fully-Connected Layer: Connects all the nodes from the previous layer to every node in the following layer, combining the learned features and producing a prediction. In particular, the output layer is responsible for producing the final classification result. Usually, the activation functions used for the output layer (σ) are sigmoid and softmax [62], as expressed in (2.2) and (2.3); they are respectively employed for binary and multi-class classification tasks, such as drone detection and identification.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2.3}$$

Namely for the softmax function, z_i represents the score for class i, and the function ensures that the predicted probabilities sum to 1 across all classes j.

2.5.3. RNNs: Recurrent Neural Networks

RNNs are a distinct class of neural networks specifically designed to handle sequential data by incorporating feedback connections and retaining information from previous inputs to influence current predictions [59], [60], as depicted in Fig. 2.6.

This architecture is particularly effective for tasks where the order of the input data is essential for accurate predictions, such as speech recognition, natural language processing, and time-series forecasting [60], [61].

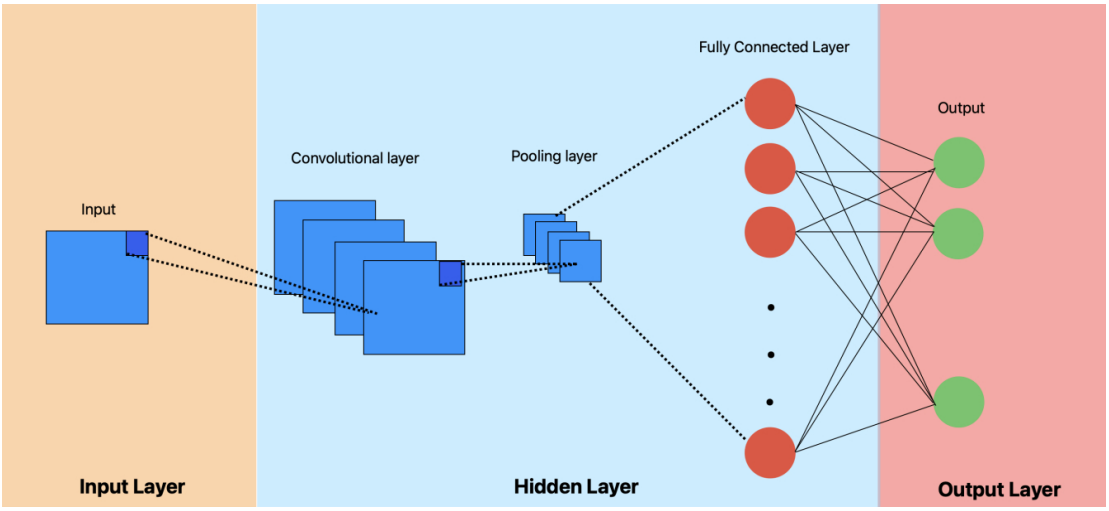


Fig. 2.5. Simple CNN architecture. From [60, Fig. 2].

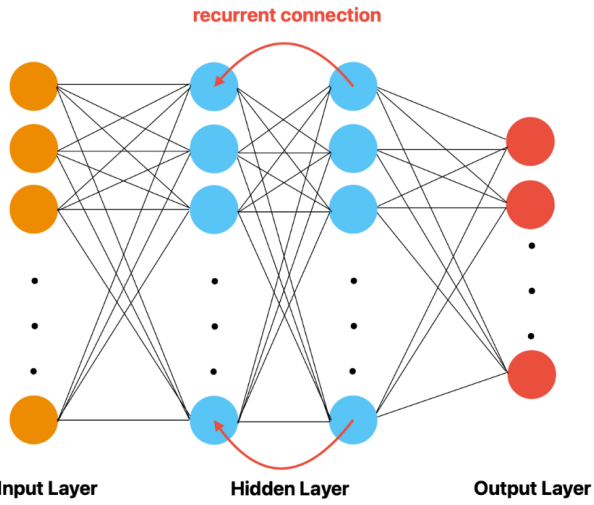


Fig. 2.6. Simple RNN architecture. From [60, Fig. 3].

2.5.4. LSTM: Long Short-Term Memory

Long Short-Term Memory (LSTM) represents a variant of RNNs which was first introduced by Hochreiter and Schmidhuber in 1997 [63]. They are particularly effective at capturing long-term dependencies in sequential data, especially for time-series applications [60], [64], [65].

LSTM networks were especially designed to prevent the vanishing gradient problem, which is a standard problem often encountered with traditional RNNs – where gradient magnitudes diminish as they are backpropagated through time – and thereby enables the long-term learning capability of the network [60].

These units address this limitation by integrating memory cells that selectively retain or discard information across time steps [65]. The fundamental components of this cell are gate mechanisms, which regulate the flow of information through the network; these consist of the input gate (i_t), output gate (o_t), and forget gate (f_t), and additional elements include cell state (c_t) update, and hidden state (h_t) update [60], [64].

Each gate has its own function, and at each time step t, they operate as follows, as shown in Fig. 2.7. The input gate (i_t) defines the amount of new information added to the cell state (c_t), whereas the output gate (o_t) controls the information added to the hidden state (h_t). Furthermore, the forget gate (f_t) determines what portion of information to retain or discard from the previous cell state (c_{t-1}). Moreover, the cell state (c_t) is updated by a combination of the values from the previous state as well as the new candidate. Finally, the hidden state (h_t) is updated according to the new cell state and the output gate modulation [60], [64], [65].

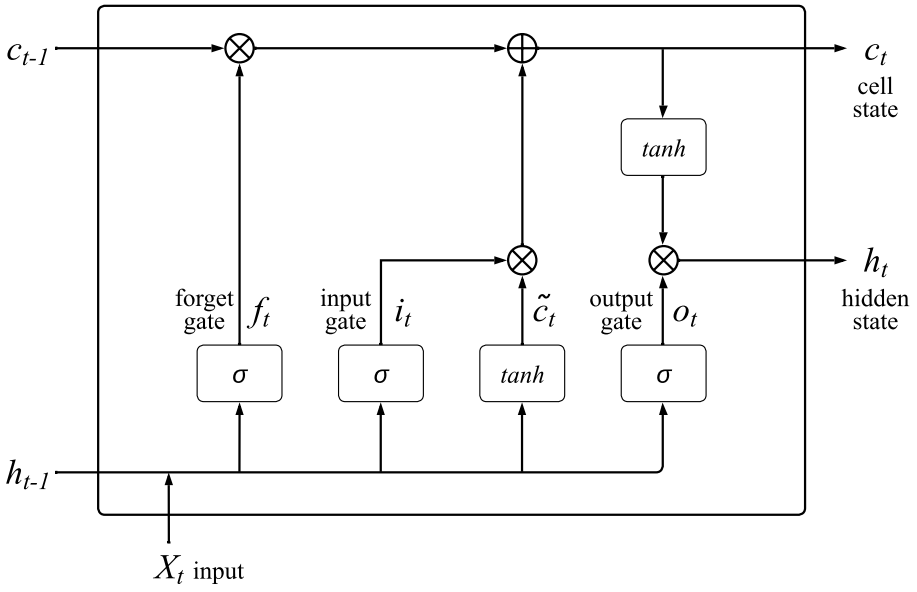


Fig. 2.7. LSTM unit architecture. Adapted from [65, Fig. 2.11].

LSTM networks can capture both short-term and long-term dependencies, by maintaining and updating the cell state across time steps. Hence, they demonstrate to be highly effective for drone detection, especially for fluctuating acoustic signals, while also showing potential for RF-based detection [60], [66].

2.5.5. Deep Learning regarding DDI

There has been encouraging evidence showing positive results concerning the impact of ML algorithms for drone detection and identification [67], [68].

Notably, several studies have shown promising results demonstrating that DL can significantly enhance UAV detection and identification by improving the robustness of signal analysis [60], [61]. In particular, some prove most relevant to this work [24], [25], [26], [27], [28], [29]; while others also show significant contributions [69], [70], [71], [72], [73], [74], mainly in the pursuit for lightweight and resilient solutions; moreover, other techniques have also been implemented, such as LSTM [66]. Remarkably, the integration of One-Dimensional (1-D) CNNs has demonstrated most popular in this field, as evident in [25], [26], for instance.

In [24], the authors aimed at building an open-source database of several drone RF signals, and produced a system for detection and identification tasks. Furthermore, to validate its feasibility, three identical DNNs were namely devised to: (1) detect the presence of a drone, achieving an average 99.7% accuracy and 99.5% F1-score; (2) identify its presence and type, with 84.5% accuracy and 78.8% F1-score; (3) and finally, detect, identify and infer its flight mode, obtaining 46.8% accuracy and 43.0% F1-score. Notably, the authors acknowledged that the system struggled identifying two drones which were produced by the same manufacturer; and observed a general performance decline with class proliferation. Moreover, ensuing research [25], [26] has adopted this dataset towards the design of increasingly effective UAV detection and identification systems, and outperformed the aforementioned DNN models.

In particular, the authors of [25] developed two distinct CNN models, for drone detection, and drone identification and mode recognition tasks, respectively. The detection model reached an average 99.8% accuracy and 99.7% F1-score; furthermore, the identification model obtained 85.8% accuracy and 84.6% F1-score for drone identification, and 59.2% accuracy and 55.1% F1-score for mode identification.

Similarly, the authors in [26] further improved upon the previous study, by proposing an alternative CNN architecture; particularly focused on identification, the model achieved higher performance with lower complexity, obtaining an average 92.5% accuracy and 94.5% F1-score.

Additionally, the authors from [74] developed a dataset regarding spectrogram data of RF fingerprints under low-Signal-to-Noise Ratio (SNR) conditions, enabling increased classification robustness against different types of noise. Moreover, several studies [27], [28], [29] have also utilised this dataset, focusing on UAV detection and classification tasks.

In a related study [27], the authors provided a comparison concerning the MobileNet V2, SqueezeNet, and ResNet algorithms for UAV identification, reaching an average accuracy of 95.24%, 91.72%, and 89.83%, and F1-score of 94.16%, 91.54%, and 89.74%, respectively.

Furthermore, the authors of [28] and [29] extended the research through the addition of white Gaussian noise and leveraging DL models including CNNs, for comparative analysis. Namely, [28] reached accuracy values ranging from 92% to 100%, at SNR $\in [-10, 30]$ dB; and [29] reached accuracies above 85%, for SNRs over -12 dB.

CHAPTER 3

Methodology

This chapter describes the methodology devised to conduct the experiments, including the dataset employed, the setup implemented for their execution, the proposed models and architectures, and evaluation metrics used to assess their performance.

Notably, the various implementations were designed and considered according to the objectives of the experiments performed, for the purpose of developing a reliable system for drone detection and identification, using DL techniques and leveraging recorded RF signals.

3.1. Dataset

RF-based drone detection and identification applications require large descriptive databases of UAV signals for training and testing.

For this purpose, and in the efforts of building and continually expanding and evolving a large labelled open-source drone database, the authors of [24] have made the initiative to create and share a comprehensive dataset of drone signals named "DroneRF dataset", which is available for download from [75]. They have encouraged the adoption and further development of this database by a community of researchers and students alike, by performing the same or similar experiments on other types of drone models and manufacturers, allowing the collection to have more diverse signals, and therefore enable more accurate and efficient detection and identification of UAVs.

The developed open-source database is composed of a collection of drone signals, structured as shown in Table 3.1; with a total of 227 segments, where each segment includes both high and low frequency band signals, consisting of one million samples each. This corresponds to a total of 4.54×10^9 raw samples captured representing the amplitude of the signal; and results in $46\,489\,600$ processed samples, for use in the DDI system.

The system devised for the development of the distributed dataset comprises a set of drones under analysis along with several modules to capture, analyse and record RF signals, namely for flight control and RF sensing.

The drones selected for analysis to create the dataset are namely Parrot Bebop [76], Parrot AR.Drone 2.0 Elite Edition [77] and DJI Phantom 3 Standard [78] drones, which will henceforth

TABLE 3.1
Dataset composition.

Class	Segments	Samples	Ratio (%)
Bebop	84	1680×10^6	37.00
AR	81	1620×10^6	35.68
Phantom	21	420×10^6	9.25
No Drone	41	820×10^6	18.06

be referred to as Bebop, AR and Phantom drones, respectively. These drones are commonly used in both research and civilian applications, as they vary in characteristics such as size, capabilities and technology. Additionally, it is noteworthy that the shared dataset contains recordings of RF background activities portraying the absence of drones, as well as RF drone activities demonstrating their presence and concerning their operation under different flight modes, such as on and connected, hovering, flying and video recording.

The flight control module is dedicated for the communication between the controller and the drones by sending and receiving RF commands, and has the ability to assign their flight modes – which can be done using either a flight controller or a mobile phone.

The RF sensing module is designed for the interception of the communication signals between the flight control module and the drone, by using RF receivers, which are connected to laptops that execute LabVIEW programs devised for subsequently fetching, processing and storing the RF signals in a database.

Furthermore, it is important to note that the RF samples were acquired using two USRP-2943 SDRs boards, and since each receiver has a maximum instantaneous bandwidth of 40 MHz, it was necessary for both receivers to operate simultaneously in order to capture a sufficiently wide spectrum of approximately 80 MHz; accordingly, the first receiver recorded the lower half of the frequency band, whilst the second captured the upper half. This arrangement was based on the premise that the authors of the dataset [24] considered that all drones employed Wi-Fi operating within the ISM band of 2.4 GHz – which covers 100 MHz (supporting 94 MHz across 14 channels, along with guard bands of 3 MHz each at the extremities) – disregarding channel 14 and 1 MHz at each edge, as the associated information was deemed negligible.

Notably, the files containing the RF signals are first stored in a `.csv` format; each entry in the database is labelled according to the drone activity, type, and flight mode, as well as the frequency band in which the signal was captured, and the segment number. Namely, their designation is performed using a binary identifier, and is structured as expressed in (3.1), where the characters `b`, `L` and `N` represent bits, letters, and numbers, respectively.

$$\text{Filename: } \underbrace{\texttt{b}}_{\substack{\text{drone}\\\text{activity}}}\ \underbrace{\texttt{bb}}_{\substack{\text{drone}\\\text{type}}}\ \underbrace{\texttt{bb}}_{\substack{\text{drone}\\\text{flight mode}}}\ \underbrace{\texttt{L}}_{\substack{\text{frequency}\\\text{band}}}\ \texttt{-}\ \underbrace{\texttt{N}}_{\substack{\text{segment}\\\text{number}}}\ \texttt{.csv} \tag{3.1}$$

In particular, the first character is a bit that represents the activity status of the drones (i.e., `0` for background activities and `1` for drone activity). Furthermore, the succeeding two bits, namely the second and third characters, indicate the type of the drone (i.e., `00` for Bebop, `01` for AR, and `10` for Phantom). Moreover, the subsequent two bits, regarding the fourth and fifth characters, specify the flight mode, following a linear binary progression (i.e., `00`, `01`, `10`, and `11`) for modes 1 through 4. Additionally, the sixth character is a letter that denotes the frequency band, namely `L` and `H` for low and high frequency bands, respectively. Finally, the last character is a digit which represents the segment number associated with the recorded signal.

This results in a filename such as `11011L_5.csv`, for instance, which corresponds to the fifth segment of a signal recorded from a Phantom drone operating in flight mode 4, captured within the low frequency band, for example.

Following the aforementioned system used to create the published dataset, this work also requires performing the subsequent procedures instrumental for the classification of UAVs signals. This is possible through the execution of MATLAB scripts provided in [79], which include labelling the database entries according to the conducted experiment task, drone type, and flight mode; and also transforming the complex signals through MATLAB Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) operation functions, to reveal underlying information useful for efficient detection and identification.

Notably, certain adjustments were applied to the original dataset, and further modifications were incorporated within the original scripts, to enhance functionality and suit the requirements of this research. Initially, the dataset downloaded from [75] contained the segments grouped by drone type. Therefore, the original compressed files were unzipped, fully extracted, and reorganised into a single folder for easier access.

Furthermore, the MATLAB scripts (particularly `Main_1_Data_aggregation.m` along with `Main_2_Data_labeling.m`) were altered according to several aspects, consisting of: refining file organisation through their location paths and document designations, to improve clarity, discovery and overall interoperability with the database; incorporating timers, to measure the execution time of every processing stage individually; adding diagnostic prints throughout the code, to monitor progress for debugging, and ensure all procedures functioned as expected.

Moreover, along with equivalent changes as applied to the aforementioned scripts, the Python script was additionally modified according to each experiment individually, including adapting the signal processing techniques pertinent to the implemented DL models.

3.2. Setup equipment

The development, training, and testing of the proposed models was conducted on a workstation running a Windows operating system.

Table 3.2 summarises further details concerning the specifications and configurations of the equipment employed, focusing on the hardware components and software configurations used. Furthermore, the proposed models were developed and integrated within a Python script, utilising a set of open-source libraries namely including NumPy, TensorFlow, and scikit-learn [80], [81], [82].

TABLE 3.2
Setup equipment specifications.

Component	Specification
Operating System	Windows 10 Education (64-bit)
Processor [CPU]	AMD Ryzen 7 1700 (8-core, 16-thread @ 3.00 GHz)
Graphics Card [GPU]	NVIDIA GeForce GT 710
Memory [RAM]	48 GB
Python version	3.11.9
MATLAB version	R2024b

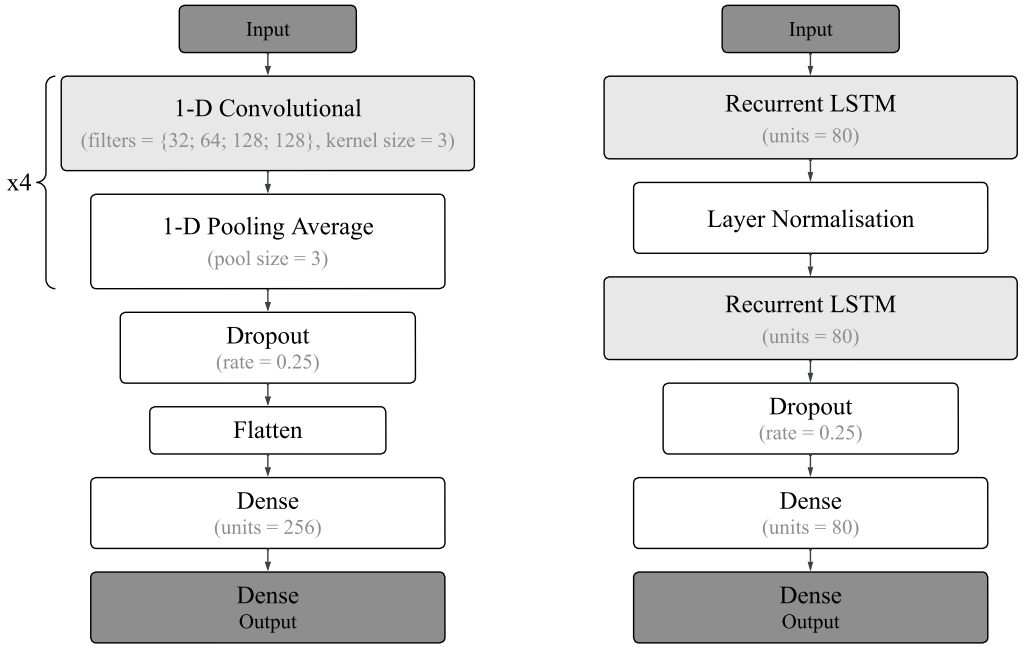
3.3. Proposed models

In order to implement an efficient and reliable system for drone detection and identification, this work proposes two neural network architectures, as demonstrated in Fig. 3.1.

The first architecture concerns a CNN, as presented in Fig. 3.1(a), which employs 1-D convolutional layers, to improve the classification accuracy while maintaining low computational complexity. The proposed architecture consists of four 1-D convolutional layers, each followed by a 1-D average pooling layer, to reduce the dimensionality of the feature maps; More precisely, the convolutional layers are defined with: number of filters 32, 64, 128, and 128, respectively; a kernel size of 3; and a stride of 1 alongside a padding of `'same'`, to preserve the resolution of the input signal; the pooling layers also have a pool size of 3. Moreover, the ReLU activation function, expressed in (2.1), is utilised following prevalent selections that have proven effective [62], as it mitigates the vanishing gradient problem, and enhances the learning efficiency of the proposed network. Additionally, a dropout layer with a 25% dropout rate was included to prevent overfitting; furthermore, a flatten layer was integrated to collapse the multi-dimensional feature maps into a unified one-dimensional feature vector; finally, the architecture concludes with two dense or fully-connected layers, the latter serving as the output layer, where the learned features are combined to produce the final classification; respectively, these layers are defined with 256 units and with a number of units corresponding to the classes associated with the conducted experiment. Notably, this network architecture is replicated from [25], as designed for drone detection.

The second architecture regards an RNN, as shown in Fig. 3.1(b), incorporating LSTM layers, which are particularly suited for temporal sequence modelling tasks and signal analysis. The proposed architecture has a relatively simple design, mainly composed of two LSTM layers separated by a layer normalisation operation to stabilise and accelerate the learning process and gradient flow, succeeded by a dropout layer with a 25% dropout rate to avoid overfitting; lastly, the architecture terminates with two dense layers, the last of which functions as the output layer, where the extracted temporal dependencies are aggregated to perform the final prediction; respectively, these fully-connected layers are configured with 256 units and with a number of units determined by the classes pertaining to the conducted experiment. Moreover, the Hyperbolic Tangent (Tanh) activation function is implemented in the recurrent layers, as originally established for LSTM units [63], since it provides a more stable and balanced learning process for the network.

Fig. 3.1 provides an overview of the proposed NN architectures, detailing the structural design of the layers, including the definition of hyperparameters such as kernel size and number of filters or units. In particular, (a) illustrates the CNN architecture, while (b) depicts the RNN architecture.



(a) CNN. **(b)** RNN.

Fig. 3.1. Architectures proposed for NN models.

3.3.1. Dilation Factor

Dilated CNNs represent a variation of traditional CNNs, which utilise values greater than 1 for the dilation factor – also recognised as dilation rate – within dilated convolutional layers [83], [84]. The increase of the dilation factor allows the network to expand the receptive field and therefore capture deeper and broader multi-scale contextual information from sequential inputs, without increasing the number of parameters, the amount of computations, or the size of the filter, whilst also preserving spatial resolution and enhancing coverage [83], [84], [85], [86]. This approach enables these networks to retain information from previous inputs, which is important to identify long-term patterns in drone signals [83], [85].

Particularly, the deployment of 1-D convolutional layers with defined dilation factors in CNNs relates to Temporal Convolutional Networks (TCNs), as these architectures are specifically designed to process time-dependent data, applying convolutions along the temporal dimension [86].

The dilation rate is a hyperparameter which defines the spacing between the kernel elements, or the dimension expansion for the input data sampling, effectively extending the receptive field, without needing to increase the kernel size or the number of parameters [65], [83], [85]. In particular, a dilation factor of 1 corresponds to a standard convolution, whereas a dilation factor of d_F accesses input values separated by $d_F - 1$ units, therefore skipping intermediate positions and introducing gaps between each element in the kernel [83], [86].

Fig. 3.2 illustrates the concept of dilation factor or dilation rate, as employed in 1-D CNN layers with filters of kernel size 3, in particular; notably, the dilation factor is designated by d_F, namely with values 1, 2, and 4, although other configurations may be used.

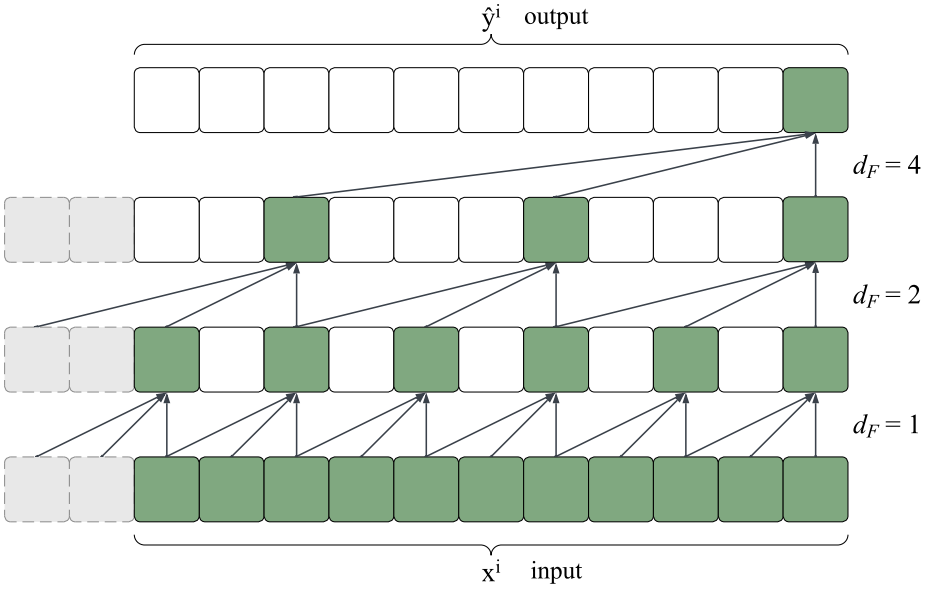


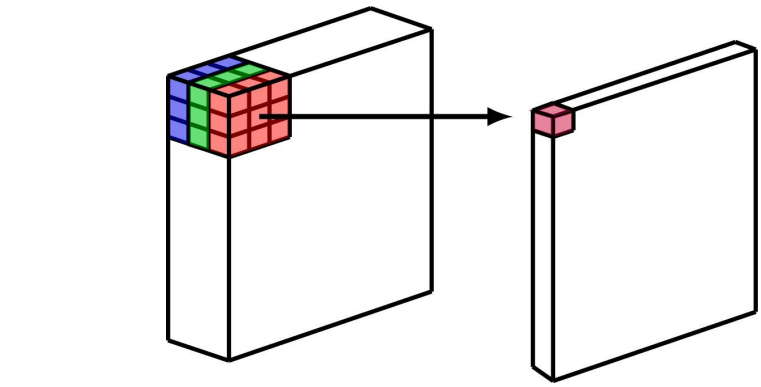
Fig. 3.2. Dilation Factor illustration. Adapted from [85, Fig. 5].

3.3.2. Depthwise Separable Convolutions

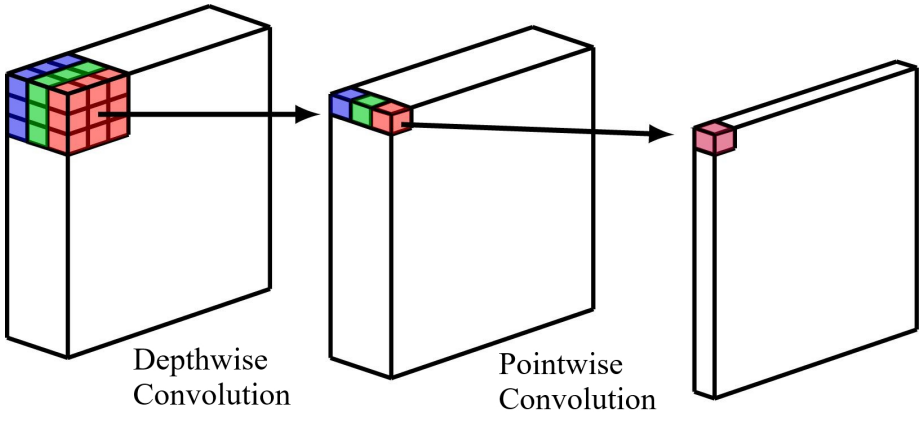
Depthwise Separable Convolutions (DSCs) represent an alternative technique used to replace standard convolutions of conventional CNNs architectures, with the aim of decreasing the number of parameters. This approach reduces the computational complexity and cost, thereby improving efficiency, whilst maintaining accuracy and overall performance [87]. This strategy is particularly beneficial for applications regarding mobile and embedded systems, where computational resources are limited, and lower latency is essential [59], [88].

A DSC divides a standard convolution into two distinct sequential operations, consisting of a depthwise convolution followed by a pointwise convolution. Initially, the depthwise convolution applies kernels of a single filter on each channel individually, performing spatial convolution filtering per channel independently. Subsequently, a pointwise convolution employs a conventional 1×1 convolution kernel to combine the resulting information linearly, across the whole depth of channels [87], [88].

Fig. 3.3 demonstrates the concept of depthwise separable convolutions [illustrated in (b)], as compared to standard convolutions [portrayed in (a)], in a two-dimensional convolutional network with 3 channels and a 3×3 filter size.



(a) Conventional CNN.



(b) Depthwise Separable CNN.

Fig. 3.3. Depthwise Separable Convolutions illustration. From [88, Fig. 1].

3.4. Evaluation metrics

The evaluation metrics used to assess the performance of the proposed model are namely accuracy, precision, recall, F1-score, error, False Discovery Rate (FDR), and False Negative Rate (FNR); these metrics are obtained and presented via confusion matrices, as delineated in Fig. 3.4. They represent statistical measures widely adopted in the field of ML to evaluate the performance of binary and multi-class classification models [89].

The mathematical formulas defining the performance metrics are presented in (3.2) to (3.8). These equations incorporate designations such as True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). In this regard, the label true represents correct predictions, whereas false denotes incorrect classifications; similarly, the labels positive and negative indicate whether predictions were classified as belonging to the target class or to other classes, respectively [90].

Accuracy is defined as the ratio between correctly predicted segments and the total number of segments; and measures the overall effectiveness of the classifier [89], [90], as demonstrated in (3.2).

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.2}$$

Precision is specified as the ratio between correctly predicted positive instances, and the number of instances assigned as positive [89], as formulated in (3.3). Hence, this metric denotes the proportion of predicted positive instances that are identified correctly.

Recall is ascertained as the ratio between correctly predicted positive instances, and the number of instances actually positive [89], as presented in (3.4). Therefore, this metric represents the proportion of actual positive instances that are identified correctly; thus, it is also known as true positive rate.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.3}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.4}$$

F1-score is considered the harmonic mean of precision and recall, as it represents a combination of both measures within a single metric, providing balance to the evaluation of the classifier [89], as expressed in (3.5).

$$\text{F1-score} = 2 \times \left(\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \right) \tag{3.5}$$

Error, FDR, and FNR are derived as the complements of the previously mentioned metrics accuracy, precision, and recall [89], as shown in (3.6) to (3.8), respectively.

$$\text{error} = 1 - \text{accuracy} \tag{3.6}$$

$$\text{FDR} = 1 - \text{precision} \tag{3.7}$$

$$\text{FNR} = 1 - \text{recall} \tag{3.8}$$

3.4.1. Confusion matrix

A confusion matrix represents a grid table that compares predicted labels with actual labels in a classification task [89]; it can also be portrayed alongside other metrics, as depicted in Fig. 3.4.

Fig. 3.4 illustrates the corresponding layout of the evaluation metrics surrounding a 2×2 confusion matrix, representing a 2-class binary classification problem, as typically used for detection of presence of drones. Likewise, Fig. 3.5 shows the corresponding layout of the evaluation metrics surrounding a 4×4 confusion matrix, regarding a 4-class classification task, as utilised for the identification of the drones in the aforementioned database.

Furthermore, the rows and columns of the inner matrix designate the predicted and actual classes, respectively; the diagonal cells – highlighted in green – indicate correct classifications, whereas the remaining cells – marked in red – depict incorrect predictions. Moreover, the aforementioned designations (i.e., TP, TN, FP and FN) are also represented; in particular, the instances labelled true are delineated in green text, and those labelled false are accentuated in red text, namely focusing on class 1 for Fig. 3.4 and class 2 for Fig. 3.5, as target references for the evaluations.

Notably, usually each cell of the inner matrix displays the number of segments, along with their proportion relative to the total number of segments.

For each of the cells in the outer component of the confusion matrix, the highlighted metric is displayed on its upper section, and the complementary measure is also provided directly underneath. The blue cell in the bottom-right corner of the matrix demonstrates the overall accuracy and error. Furthermore, the gray column on the right details the precision and FDR. Moreover, the gray row at the bottom presents the recall and FNR. Additionally, the yellow row and column, namely at the top and left of the matrix, display the F1-scores and their complementary below. Finally, the orange cell in the top-left corner reports the F1-score averaged for all classes, with its complementary presented beneath.

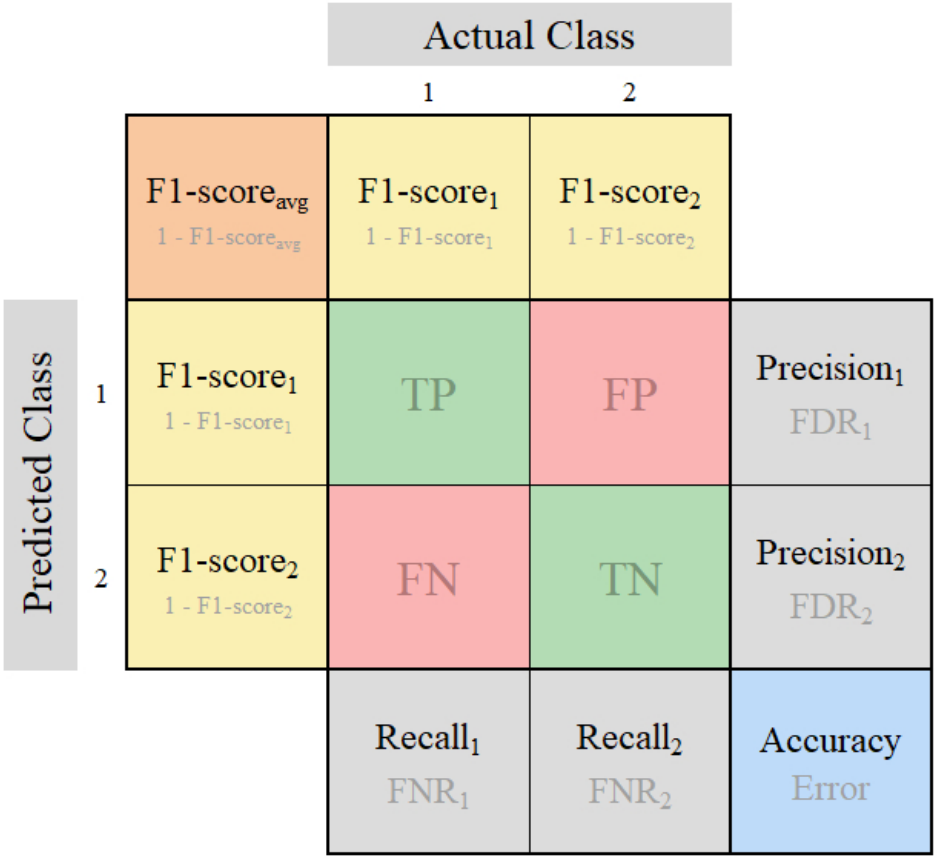


Fig. 3.4. Confusion Matrix layout of Evaluation Metrics (2-class).

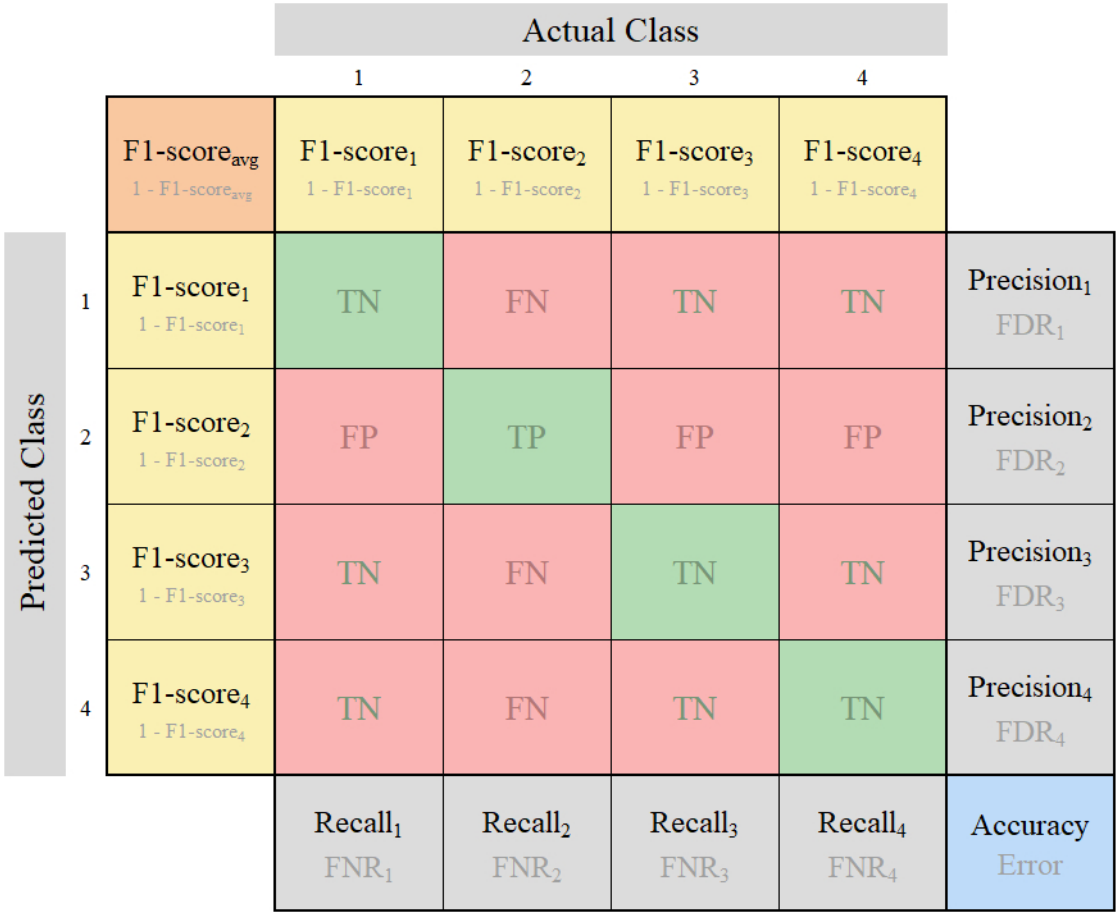


Fig. 3.5. Confusion Matrix layout of Evaluation Metrics (4-class).

CHAPTER 4

Results and Discussion

This chapter reports the descriptions of the implemented setup and configurations concerning the various conducted experiments and corresponding tests, along with an analysis of the simulation results.

4.1. Experimental Setup

In this work, several experiments were conducted, and numerous simulations were performed for both detection and identification tasks. The detection task is defined as a binary classification problem, which involves determining the absence or presence of a drone, represented by classes 1 and 2, respectively. Additionally, the identification problem characterises a multi-class classification problem, which focuses on recognising the specific model of the detected drone. In this regard, particularly background activity, Bebop, AR and Phantom drones are denoted as classes 1, 2, 3, and 4, respectively. For further clarification, section 3.4.1 expounds on the interpretation of the confusion matrices, analysing the various relevant aspects delineated throughout their representations.

Furthermore, these experiments aimed to evaluate the performance of various model configurations, utilising the proposed architectures presented in Fig. 3.1. In particular, the following aspects were investigated: the application of dilation in convolutional layers; the inclusion of depthwise convolutions; and the implementation of recurrent layers.

Moreover, to demonstrate the effectiveness of the proposed models, a comparison is performed concerning existing state-of-the-art research [25], focusing on the performance of the respective models designed for detection and identification. In particular, this comparison specifically regards the effect of the dilation factor, noting that increasing the dilation value represents a mechanism to expand the receptive field of the NN, without introducing additional parameters nor enlarging the kernel size, while also keeping spatial resolution.

The proposed NNs were designed and developed within a Python environment, leveraging a set of open-source libraries including NumPy, TensorFlow, and scikit-learn [80], [81], [82]. Furthermore, the simulations were conducted using specific parameters, which were optimised according to the objectives of each experiment.

Table 4.1 presents further details regarding the hyperparameters selected for the simulations of each experiment performed in this work.

TABLE 4.1
Hyperparameters considered for experiment simulations.

Task	Experiment Focus	Epochs	Batch Size	Optimiser Loss Function	Output Activation Function
Detection	Dilation Factor	200	50	Binary cross-entropy	Sigmoid
	Depthwise Convolutions	200	50	Binary cross-entropy	Sigmoid
	Recurrent Layers	200	50	Binary cross-entropy	Sigmoid
Identification	Dilation Factor	400	50	Categorical cross-entropy	Softmax

The selection of the hyperparameters was guided by the need to ensure stable convergence and consistent performance across all experiments. Moreover, the training for all the simulations was conducted using the following hyperparameters: a total of 200 epochs for detection, and 400 epochs for identification, to provide sufficient iterations for convergence without overfitting; and a batch size of 50 samples, to balance computational efficiency and gradient stability during training. Additionally, the optimiser algorithm and loss function and the activation function in the output layer were selected following established best practices [62]. In particular, the Adam optimiser algorithm was chosen for its adaptive learning rate capabilities and proven robustness in training DNNs. Furthermore, the binary cross-entropy loss function was used in combination with the sigmoid function for detection; whereas, the categorical cross-entropy loss function was employed along with the softmax function for identification.

Moreover, the classification performance of each proposed NN model was validated through a stratified K-fold cross-validation, particularly a 10-fold cross-validation process. This decision was motivated by considering that the dataset is imbalanced, with each class containing a significant difference in the number of samples, which could compromise the ability of the model to generalise.

Firstly, the complete dataset is randomly segmented into K sections or subsets each called a fold, ensuring every fold has a balanced number of instances from all classes. Afterwards, at each iteration, one fold is designated as the testing set, while the remaining $K-1$ subsets are used for training, such that each fold is employed only once as test data. This procedure is then repeated K times, allowing the model to be evaluated on the entire database. Finally, the estimated performance is obtained by the average value across all iterations [91].

4.2. Detection using Dilation Factor

In this section, the succeeding experiments regard the detection of drones, focusing on the use of dilation factor, described in section 3.3.1. The adoption of dilation factor values greater than 1 enables the network to analyse a broader receptive field, therefore aggregating more detailed contextual information, and thereby facilitating the pattern recognition.

4.2.1. Experiment with dilation rate 2

In this experiment, the dilation rate was applied to the 1-D convolutional layers of the proposed CNN architecture, portrayed in Fig. 3.1(a). Furthermore, numerous tests were conducted to evaluate the impact of different dilation factor values on the performance, until the best results were achieved.

The testing procedure consisted of defining the dilation rate as a specific value and assigning it to selected convolutional layers, described as follows. In particular, the dilation factor was set to the value 2 (i.e., $d_F = 2$), for this experiment. Initially, the dilation rate was applied exclusively to the first layer, while the remaining layers maintained a standard dilation rate of 1. Furthermore, in subsequent tests, the dilation rate was applied individually to the second layer, then only to the third layer, and afterwards to the fourth and last layer. Finally, the dilation rate was defined for all convolutional layers simultaneously.

Fig. 4.2 demonstrates the results of the optimal configuration for this experiment, demonstrating the highest performance is attained when the dilation factor of 2 is applied across all convolutional layers. Notably, the implementation regarding the dilation factor of 2 in layer 4 also yields similar results. Additionally, the results of the remaining tests are also included for comparison, in Appendix A (Fig. A.1). Moreover, the results of existing literature [25] are presented in Fig. 4.1, for comparison.

The proposed configuration (Fig. 4.2) achieves an average performance approaching 100% across all evaluated metrics, namely accuracy, precision, recall, and F1-score; these results represent a high level of performance, indicating near-perfect detection of drones. Furthermore, a comparison with the existing state-of-the-art [25] (Fig. 4.1) reveals this approach outperforms their proposed model, with improvements of 0.2% in accuracy, 0.1% in precision, 0.35% in recall, and 0.2% in F1-score.

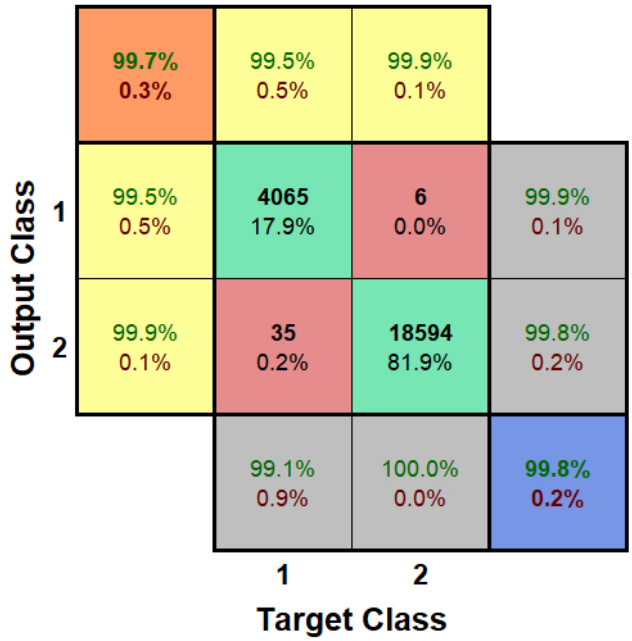


Fig. 4.1. Literature Results regarding Detection. Adapted from [25].

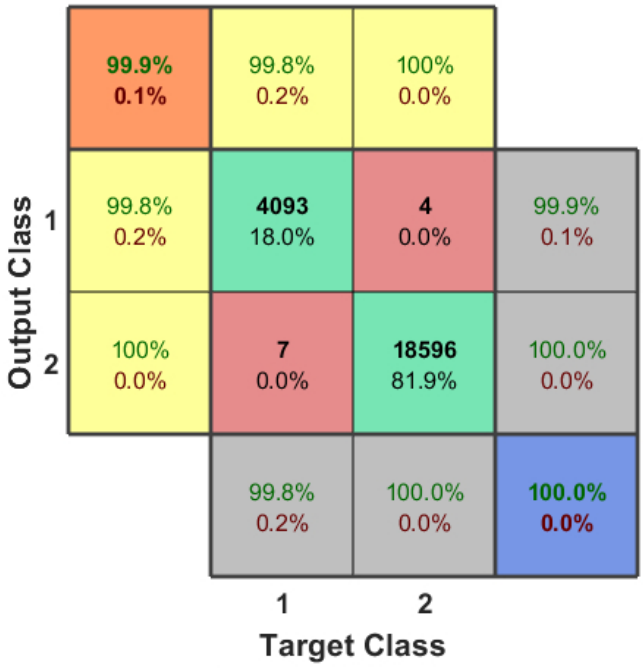


Fig. 4.2. Confusion Matrix regarding Detection using Dilation Rate 2.

4.2.2. Experiment with dilation rate 3

Building upon the previous experiment, additional tests were conducted following a similar methodology. In particular, the dilation factor was set to the value 3 (i.e., $d_F = 3$), for this experiment. Furthermore, the same testing procedure was repeated, applying the defined dilation rate to different convolutional layers, as previously described.

Fig. 4.3 depicts the results of the ideal approach for this experiment, indicating the best performance is obtained when the dilation factor of 3 is applied exclusively to the third layer. Notably, the configuration concerning the dilation factor of 3 across all layers also shows analogous results. Moreover, the results of the remaining tests are also included for comparison, in Appendix A (Fig. A.2). Additionally, the results of existing literature [25] are presented in Fig. 4.1, for comparison.

The results of this experiment (Fig. 4.3) closely resemble those of the previous experiment, with an average performance approaching 100% across all evaluated metrics, and thus also signifying an almost excellent detection of drones. Furthermore, through a comparison with the model proposed in [25] (Fig. 4.1), this configuration exhibits increases of 0.2% in accuracy, 0.1% in precision, 0.45% in recall, and 0.3% in F1-score.

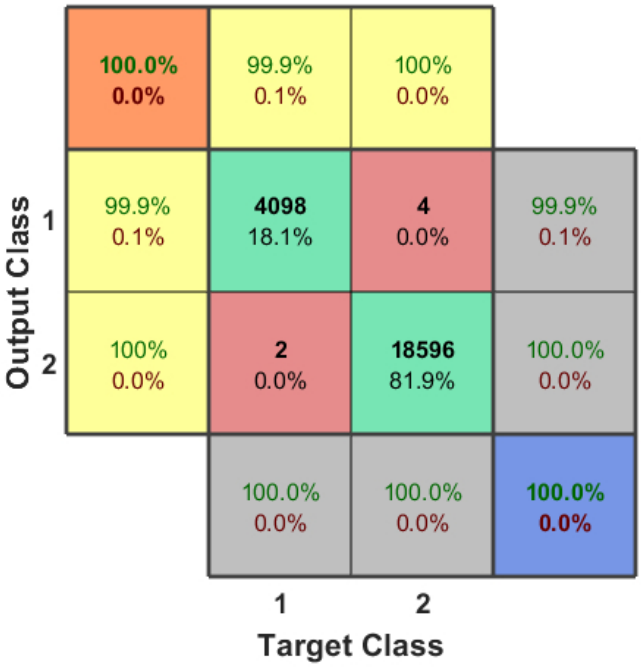


Fig. 4.3. Confusion Matrix regarding Detection using Dilation Rate 3.

Comparing the results of both experiments, it is evident that the configuration with a dilation factor of 2 presents a similar performance to the configuration with a dilation factor of 3; these results differ by only five predictions among a total of 22 700 classifications, all of which notably represent instances where the prediction was made of the presence of the drone, when it was actually absent. Moreover, for both configurations where the dilation factor is applied across all convolutional layers, the results are essentially equivalent, with minimal differences observed. Hence, since the difference in results is negligible, the configurations with a dilation factor of 2 are preferred, as it provides a more balanced receptive field and preserves finer local details, while maintaining overall performance [84].

Notably, in addition to the confusion matrices used to evaluate the performance of the models, the values for accuracy and loss were also obtained throughout the training and validation phases and plotted afterwards, to provide further insights and details for review and ensure proper model learning progress and generalisation capabilities. Fig. 4.4 presents these parameters through temporal evolution graphs based on epochs, particularly regarding the experiment configuration with a dilation factor of 2 across all convolutional layers.

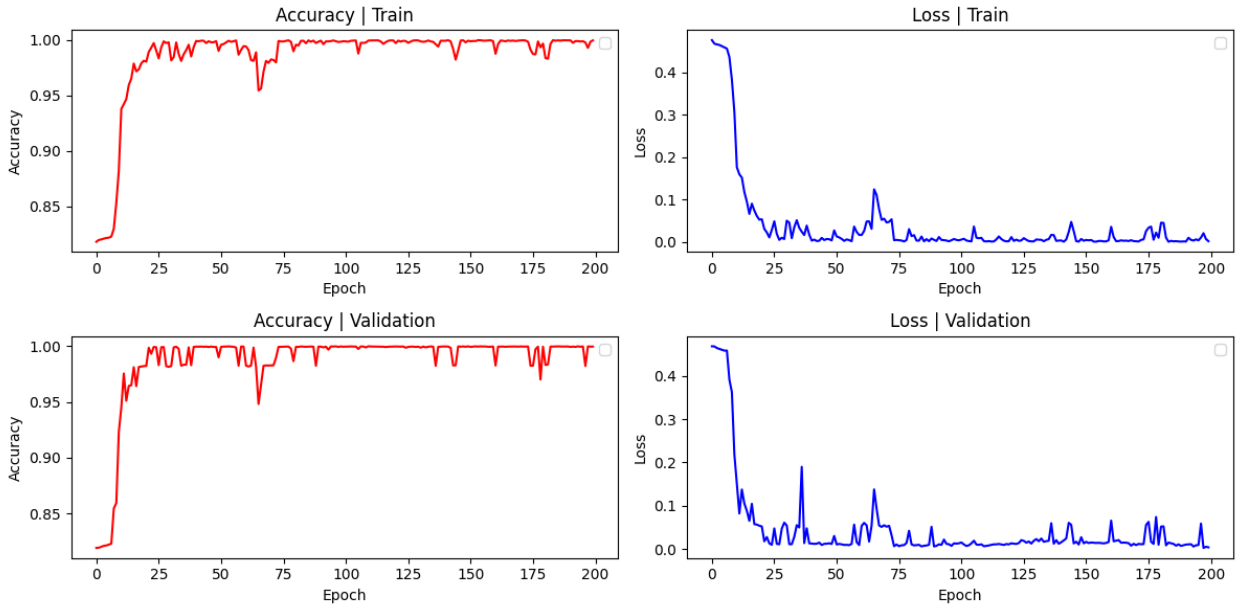


Fig. 4.4. Graphs of Accuracy and Loss regarding Detection using Dilation Rate 2.

In particular, the plots illustrate the evolution of accuracy and loss for both the training and validation sets of the model, represented on the left and right sides, and on the upper and lower sections of the figure, respectively. Furthermore, the values were obtained over 200 epochs, as indicated in Table 4.1.

The accuracy curves exhibit a rapid increase in performance within the first 25 epochs, approximately, reaching values close to 1.0 (100%), indicating that the network quickly learns the relevant and distinct features of the drones. Subsequently, and until epoch 100, both training and validation accuracies stabilise near perfection with minor oscillations, remaining consistently high thereafter up to epoch 200, demonstrating strong convergence and robust generalisation.

The loss curves further support these observations, as both training and validation losses decrease sharply during the initial 25 epochs as well, approaching values towards 0.0, and continue near zero with slight variations, reducing classification error with minimal instability.

Overall, the graphs of accuracy and loss demonstrate that the model learns and generalises effectively. Furthermore, the close alignment between training and validation curves shows consistent performance with no significant overfitting. Moreover, although the model learning appeared to plateau around epoch 100, training was extended to 200 epochs to: observe potential overfitting; guarantee convergence stability; confirm generalisation capability; and also ensure consistency across experiments.

4.3. Detection using Depthwise Convolutions

In this section, the ensuing experiments concern the detection of drones, particularly through the use of depthwise convolutions, described in section 3.3.2. The introduction of DSCs aims to reduce the computational cost and complexity of the model, whilst maintaining performance.

4.3.1. Experiment with depthwise separable convolutions

In this experiment, depthwise separable convolutional layers were used as replacement for the standard convolutional layers of the proposed CNN architecture, illustrated in Fig. 3.1(a). Moreover, various tests were conducted to evaluate the impact of DSCs on performance, until the optimal configuration was identified.

The testing procedure consisted of substituting selected convolutional layers for depthwise separable convolutional layers, as further described. Particularly, the layers were replaced cumulatively, beginning from the fourth and last layer, or layer 4, and progressively supplementing with the preceding layer, concluding with the combination of all layers concurrently substituted. Initially, the replacement was applied exclusively to the fourth layer, or layer 4. Furthermore, in subsequent tests, the substitution was extended: to the fourth and third layers, or layers 4 and 3 simultaneously; moreover, to layers 4, 3, and 2; and finally, to all layers 4, 3, 2, and 1.

Fig. 4.5 displays the results of the ideal approach for this experiment, indicating the best performance is obtained when the replacement is implemented for layers 4 and 3. Moreover, the results of the remaining tests are also included for comparison, in Appendix A (Fig. A.3).

The proposed approach (Fig. 4.5) attains an average performance reaching almost 100% over all evaluated metrics, namely accuracy, precision, recall, and F1-score, thereby also demonstrating a near-perfect detection of drones; as these results closely align with those of the previous experiments (section 4.2).

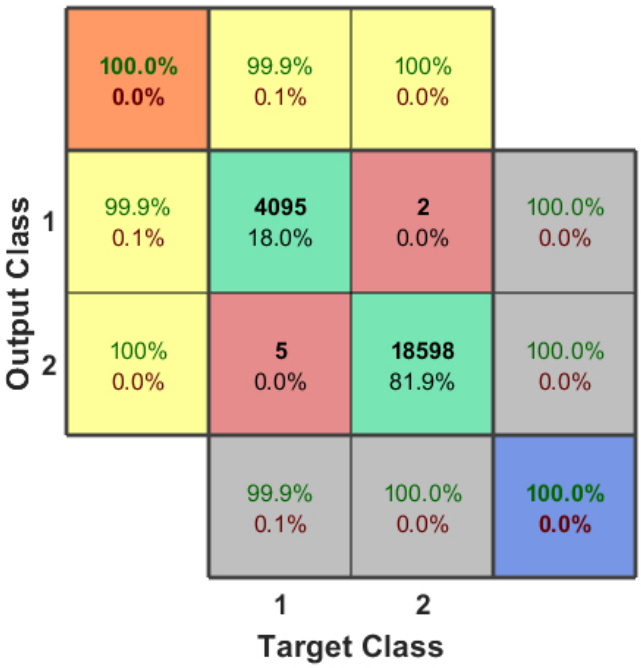


Fig. 4.5. Confusion Matrix regarding Detection using Depthwise Separable Convolutions.

Particularly, comparing the results of this experiment with those of the experiments regarding the use of dilation factor (section 4.2), specifically regarding the preferred dilation factor of 2 configuration, it is noticeable that the approach with depthwise separable convolutions exhibits a similar performance to the preferred dilation factor configuration, differing by only four predictions in a total of $22\,700$ classifications. Furthermore, these results are practically equivalent, with negligible differences observed; thus, the approach with depthwise separable convolutions is preferred, as it achieves lower complexity while maintaining performance [87].

Notably, the DSCs configuration utilises approximately 5% fewer parameters, resulting in $851\,714$ learnable parameters, from the $900\,290$ of the dilation factor of 2 approach.

4.4. Detection using Recurrent layers

In this section, the following experiments regard the detection of drones, specifically concerning the use of recurrent layers, described in section 2.5.4. The integration of LSTM layers seeks to enhance the ability of the network to capture long-term temporal dependencies within sequential signals.

4.4.1. Experiment with LSTM layers

In this experiment, recurrent layers were implemented as per the proposed RNN architecture, portrayed in Fig. 3.1(b). Furthermore, numerous tests were conducted to evaluate the different impacts of various configurations on the performance, until suitable results were achieved.

The testing procedure consisted of modifying selected hyperparameters and layers incrementally, ultimately culminating in the architecture shown in Fig. 3.1(b). It should be noted that RNNs often exhibit greater training sensitivity than other sequential NNs, particularly concerning gradient instabilities – such as the vanishing and exploding gradient problems – especially when processing prolonged sequences. Notably, LSTM networks mainly remain susceptible to the exploding gradient problem, though they may nonetheless encounter vanishing gradients, despite incorporating mechanisms regarding its mitigation [63], [64]. Consequently, these architectures require more careful tuning of hyperparameters. In particular, the dataset imbalance referenced in section 4.1 was identified as a limiting factor incurring pronounced overfitting; moreover, the cross-validation strategy previously adopted to mitigate this issue proved insufficient. Various configurations were explored to optimise the performance of the model, including: the number of hidden units implemented in the LSTM layers (i.e., 64, 80, and 100); the activation function employed (i.e., ReLU and Tanh); the integration of dropout and pooling layers (i.e., global average and max pooling); several learning rate values (i.e., $\alpha = 10^{-k},\ k\ \in\ \{2, 3, 4, 5, 6, 7, 8\}$). Finally, class balancing was achieved by employing the function `compute_class_weight` from scikit-learn library [82]; initially, the one-hot encoded labels from the original data were converted into integer form, after which weights were generated and mapped to each class to control the influence of each class, and consequently counterbalance the dataset distribution inequality.

Fig. 4.6 reports the results of the configuration used for this experiment, indicating the performance attained, after extensive testing involving multiple diverse procedural optimisation attempts.

This strategy (Fig. 4.6) obtains an average performance of 50.0% in accuracy, precision, and recall, and 44.4% in F1-score, therefore showing unreliable performance, and signifying a limited capability to detect the presence of drones.

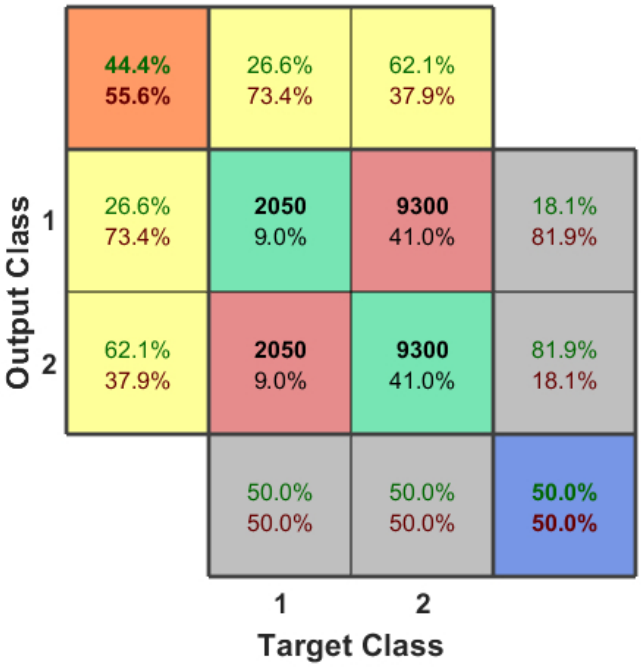


Fig. 4.6. Confusion Matrix regarding Detection using LSTM layers.

Specifically, comparing the results between this experiment and those of the preceding experiments concerning dilation factor and depthwise convolutions (sections 4.2 and 4.3), it can be observed that the approach used reports significantly worse performance than the previous configurations employed. Accordingly, the aforementioned methods are preferred for the detection task.

However, it is noteworthy that the architecture used for this experiment has much lower complexity than that of the previous experiments, which is not be optimally suited for the task, with $84\,562$ learnable parameters, reflecting a reduction of approximately 10 times lesser and 90% fewer parameters than the previous experiment with the lowest complexity (i.e., $851\,714$ training parameters – for the DSCs approach). Moreover, this architecture may still incur overfitting, especially due to the exploding gradient problem, despite the class balancing and other techniques employed. This was expected, as the focus of this experiment remained to explore the potential of RNNs for the detection of drones.

Accordingly, future work could extend the development of an improved architecture specialised in further detail to leverage the potential advantages of RNNs for the drone detection problem.

4.5. Identification using Dilation Factor

In this section, the following experiments concern the identification of drones, particularly regarding the use of dilation factor, described in section 3.3.1.

4.5.1. Experiment with dilation rate 2

In addition to the experiments performed in section 4.2, regarding the dilation factor as applied for the purpose of drone detection, further tests were conducted to evaluate the impact of dilation factor on the performance of the model, for drone identification, in particular. In this regard, as previously noted in section 4.1, the classes 1, 2, 3, and 4 correspond to background activity and the Bebop, AR and Phantom drones, respectively. Moreover, in this experiment, the dilation rate was employed in the same proposed CNN architecture, depicted in Fig. 3.1(a).

Similarly to the previous experiments concerning detection, the testing procedure consisted of defining the dilation rate as a specific value and assigning it across all convolutional layers. This consideration was encouraged by the results obtained in the aforementioned experiments, where the best performance was achieved when this configuration was used. In particular, the dilation factor was set to the value 2 (i.e., $d_F = 2$), for this experiment.

Fig. 4.8 presents the results of the configuration used for this experiment, showing the performance obtained. Moreover, the results of existing literature [25] are presented in Fig. 4.7, for comparison.

This approach (Fig. 4.8) achieves an average performance of 76.70% for accuracy, 89.23% for precision, 71.20% for recall, and 73.50% for F1-score; overall, these results indicate a subpar performance, reflecting a promising yet limited ability to identify drones. Furthermore, a comparison with existing state-of-the-art [25] (Fig. 4.7) reveals this configuration attains lower performance than their model, with approximate reductions of 9.1% in accuracy, 2.1% in precision, 10.5% in recall, and 11.1% in F1-score.

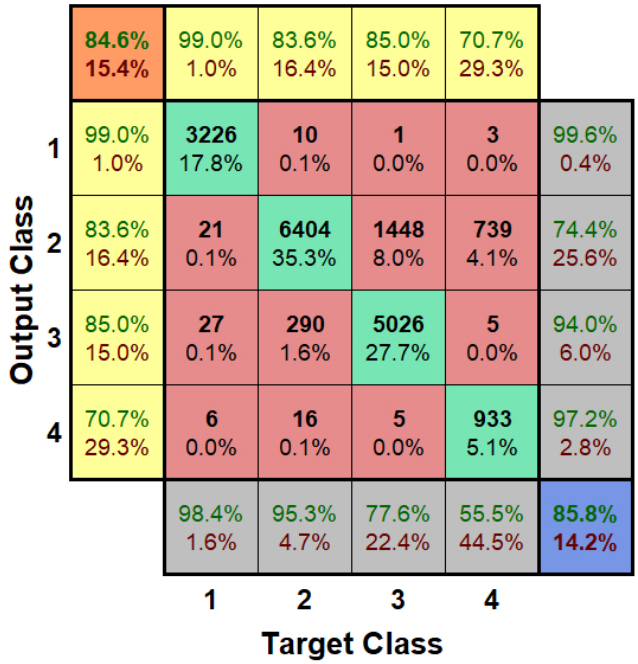


Fig. 4.7. Literature Results regarding Identification. Adapted from [25].

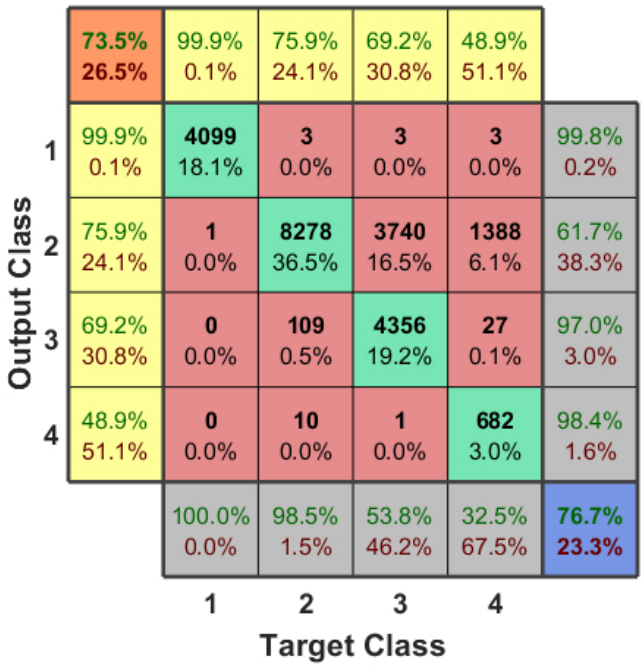


Fig. 4.8. Confusion Matrix regarding Identification using Dilation Rate 2.

Comparing the results of the best experiments for detection and identification, respectively, it is evident that the approach used for identification presents worse performance than the configurations employed for drone detection. However, it is important to note that the identification task is inherently more complex, as it involves distinguishing between multiple classes, concerning different drone models; whereas detection primarily focuses on differentiating between the presence or absence of a drone. Accordingly, identification performance does not reach desired levels, reflecting a constrained model for drone identification; nevertheless, there remains potential for improvement, considering the increased complexity of the task.

Moreover, as also recognised by the authors of the DroneRF article [24] based on their own experimental observations, the main difficulties concern the spectral similarities in RF signals between the Bebop and AR drones, both produced by the same manufacturer, named Parrot.

Furthermore, particularly for the dilation factor experiments, as discussed in sections 4.2 and 4.5, it is noteworthy that the architecture utilised for identification is the same as used for detection, which may not be optimally suited for the identification task. This was expected, as the authors of [25] noted they required numerous attempts to discover the optimal architectures, ultimately resulting in a different architecture from the one used for detection.

Consequently, future work could explore the development of a specialised architecture tailored in more detail for the drone identification problem.

CHAPTER 5

Conclusions and Future Work

This chapter concludes this thesis' work with a brief review and final remarks of its design, development and results; and provides suggestions for further improvements and alternative approaches to be adopted in future work.

5.1. Conclusions

Drones have become widely adopted across various applications, and the growing accessibility and incidents have raised concerns for security and privacy. Thus, there is a pressing need to develop accurate, efficient, and reliable systems for drone detection and identification.

Hence, to address this concern, this work proposes two NN architectures: a CNN employing 1-D convolutional layers; and an RNN using LSTM units. The system leverages an existing dataset with recorded drone RF signals for the classification of UAVs.

To assess model capability, several experiments were conducted, focusing on the implementation of different dilation factors, depthwise separable convolutions, and recurrent layers for detection; and the inclusion of dilation rate for identification. Furthermore, performance is evaluated using confusion matrices and standard metrics, including accuracy, precision, recall, and F1-score.

Moreover, the simulation results prove the potential of the proposed models. In particular, the experiments using dilation factor and depthwise separable convolutions achieve high accuracy values of approximately 100%, demonstrating effective methods for drone detection. Conversely, the results of the experiments incorporating recurrent layers for detection and dilation rate for identification show limited capability, with 50.0% and 76.7% accuracy, respectively; the decreased performance may be attributed to lower architecture complexity, and signal similarity between drones of the same manufacturer.

In summary, SDR-captured drone communication RF signals can be effectively combined with DL techniques to develop reliable systems capable of detecting and identifying UAVs, particularly by using approaches including dilation factor or depthwise separable convolutions.

The scientific investigation conducted in this work contributed to the submission of the research article referenced in [92], as presented in Appendix B.

5.2. Future Work

Drone detection and identification represents a rapidly evolving field of research, whose significance continues to increase with the growing integration of drones across a wide range of applications. Consequently, the improvement of models and techniques remains a constant and necessary challenge towards the design of efficient and reliable systems.

Hence, there remain various considerations for future research to further advance the development of drone detection and identification systems, including:

- Real-world deployment: Implement and evaluate the proposed methodologies in real operational environments, utilising SDRs and commercial UAVs. Such experiments enable the assessment and validation of the system robustness under both controlled and uncontrolled conditions, incorporating realistic impairments such as interference, hardware constraints, and dynamic flight patterns, for instance.
- Design optimisation: Improve the identification model architecture (used in the experiment described in section 4.5), by exploring more specialised network topologies tailored for multi-class classification tasks. Performance enhancements can be achieved by modifying the architecture through the integration of additional or fewer layers, as well as diverse configurations – including pooling, normalisation and dropout layers, among other hyperparameters – thereby enabling accurate and reliable drone identification.
- Alternative strategies: Investigate alternative methods, such as the incorporation of attention mechanisms. These mechanisms have demonstrated encouraging results in improving classification accuracy whilst also increasing computational efficiency, by enhancing relevant feature extraction and mitigating background noise in complex environments [60], [93];
- Hybrid networks: Instantiate a Convolutional Recurrent Neural Network (CRNN), which represents a hybrid approach combining the advantages of CNN and RNN architectures. This architecture combines the spatial feature extraction of CNNs with the temporal sequence modelling of RNNs, and has shown promising results in the literature [94], [95].

References

[1] M. A. Khan, H. Menouar, A. Eldeeb, A. Abu-Dayya, and F. D. Salim, "On the Detection of Unauthorized Drones—Techniques and Future Perspectives: A Review," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 11 439–11 455, 2022. DOI: 10.1109/jsen.2022.3171293

[2] G. Abro, S. Zulkifli, R. Masood, V. Asirvadam, and A. Laouiti, "Comprehensive Review of UAV Detection, Security, and Communication Advancements to Prevent Threats," *Drones*, vol. 6, no. 10, p. 284, 2022. DOI: 10.3390/drones6100284

[3] U. Seidaliyeva, L. Ilipbayeva, K. Taissariyeva, N. Smailov, and E. T. Matson, "Advances and Challenges in Drone Detection and Classification Techniques: A State-of-the-Art Review," *Sensors*, vol. 24, no. 1, p. 125, 2023. DOI: 10.3390/s24010125

[4] I. Mademlis *et al.*, "High-Level Multiple-UAV Cinematography Tools for Covering Outdoor Events," *IEEE Transactions on Broadcasting*, vol. 65, no. 3, pp. 627–635, 2019. DOI: 10.1109/TBC.2019.2892585

[5] K. Kuru, D. Ansell, W. Khan, and H. Yetgin, "Analysis and Optimization of Unmanned Aerial Vehicle Swarms in Logistics: An Intelligent Delivery Platform," *IEEE Access*, vol. 7, pp. 15 804–15 831, 2019. DOI: 10.1109/ACCESS.2019.2892716

[6] S. Sudhakar, V. Vijayakumar, C. S. Kumar, V. S. Priya, L. Ravi, and V. Subramaniyaswamy, "Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires," *Computer Communications*, vol. 149, pp. 1–16, 2020. DOI: 10.1016/j.comcom.2019.10.007

[7] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the Sky: Leveraging UAVs for Disaster Management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017. DOI: 10.1109/MPRV.2017.11

[8] A. Famili, A. Stavrou, H. Wang, J.-M. Park, and R. Gerdes, "Securing Your Airspace: Detection of Drones Trespassing Protected Areas," *Sensors*, vol. 24, no. 7, p. 2028, 2024. DOI: 10.3390/s24072028

[9] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichitiu, and D. Matolak, "Detection, Tracking, and Interdiction for Amateur Drones," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 75–81, 2018. DOI: 10.1109/MCOM.2018.1700455

[10] S.-h. Park and K.-h. Lee, "Developing Criteria for Invasion of Privacy by Personal Drone," in *2017 International Conference on Platform Technology and Service (PlatCon)*, Busan, South Korea, 2017, pp. 1–7. DOI: 10.1109/PlatCon.2017.7883702

[11] S. Shackle, "The mystery of the Gatwick drone," *The Guardian*, Dec. 1, 2020. Accessed: Dec. 17, 2024. [Online]. Available: https://www.theguardian.com/uk-news/2020/dec/01/the-mystery-of-the-gatwick-drone

[12] "Gatwick Airport Drone Incidents: The Need for Airport Drone Security Technology," *Dedrone*, Dec. 20, 2018. Accessed: Dec. 17, 2024. [Online]. Available: https://www.dedrone.com/blog/gatwick-airport-drone-incident-shut-down-from-deliberate-drone-disruption

[13] "Poland Airspace Breach Sparks Security Council Emergency Meeting as Allies Vow Defense," *United Nations*, Sep. 12, 2025. Accessed: Sep. 18, 2025. [Online]. Available: https://press.un.org/en/2025/sc16168.doc.htm

[14] K. L. B. Cook, "The Silent Force Multiplier: The History and Role of UAVs in Warfare," in *2007 IEEE Aerospace Conference*, Big Sky, MT, USA, 2007, pp. 1–7. DOI: 10.1109/AERO.2007.352737

[15] F.-L. Chiper, A. Martian, C. Vladeanu, I. Marghescu, R. Craciunescu, and O. Fratu, "Drone Detection and Defense Systems: Survey and a Software-Defined Radio-Based Solution," *Sensors*, vol. 22, no. 4, p. 1453, 2022. DOI: 10.3390/s22041453

[16] S. Park, H. T. Kim, S. Lee, H. Joo, and H. Kim, "Survey on Anti-Drone Systems: Components, Designs, and Challenges," *IEEE Access*, vol. 9, pp. 42 635–42 659, 2021. DOI: 10.1109/ACCESS.2021.3065926

[17] R. Ferreira, J. Gaspar, P. Sebastião, and N. Souto, "A Software Defined Radio Based Anti-UAV Mobile System with Jamming and Spoofing Capabilities," *Sensors*, vol. 22, no. 4, p. 1487, 2022. DOI: 10.3390/s22041487

[18] F.-L. Chiper, A. Martian, D.-I. Muscalu, C. Vladeanu, and I. Marghescu, "Aerial Drone Defense System based on Software Defined Radio Platforms," in *2022 14th International Conference on Communications (COMM)*, Bucharest, Romania, 2022, pp. 1–4. DOI: 10.1109/COMM54429.2022.9817314

[19] K. Pärlin, M. M. Alam, and Y. L. Moullec, "Jamming of UAV remote control systems using software defined radio," in *2018 International Conference on Military Communications and Information Systems (ICMCIS)*, Warsaw, Poland, 2018, pp. 1–6. DOI: 10.1109/ICMCIS.2018.8398711

[20] S. Liaquat, M. Faizan, J. N. Chattha, F. A. Butt, N. M. Mahyuddin, and I. H. Naqvi, "A framework for preventing unauthorized drone intrusions through radar detection and GPS spoofing," *Ain Shams Engineering Journal*, vol. 15, no. 5, p. 102 707, 2024. DOI: 10.1016/j.asej.2024.102707

[21] S. Bi, K. Li, S. Hu, W. Ni, C. Wang, and X. Wang, "Detection and Mitigation of Position Spoofing Attacks on Cooperative UAV Swarm Formations," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1883–1895, 2024. DOI: 10.1109/TIFS.2023.3341398

[22] F. A. S. Gonçalves, "Signal power directivity analyzer for UAV links using Software Defined Radios," M.S. thesis, Dept. of Inf. Sci. and Tech., Iscte – Univ. Inst. of Lisbon, Lisbon, Portugal, 2024. Accessed: Jan. 16, 2025. [Online]. Available: http://hdl.handle.net/10071/34783

[23] T. F. da C. Soeira, "Analisador de Espectro Inteligente para Deteção de Comunicações de UAVs utilizando Rádios Definidos por Software," M.S. thesis, Dept. of Inf. Sci. and Tech., Iscte – Univ. Inst. of Lisbon, Lisbon, Portugal, 2024. Accessed: Jan. 21, 2025. [Online]. Available: http://hdl.handle.net/10071/34192

[24] M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, "RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database," *Future Generation Computer Systems*, vol. 100, pp. 86–97, 2019. DOI: 10.1016/j.future.2019.05.007

[25] S. Al-Emadi and F. Al-Senaid, "Drone Detection Approach Based on Radio-Frequency Using Convolutional Neural Network," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, Doha, Qatar, 2020, pp. 29–34. DOI: 10.1109/ICIoT48696.2020.9089489

[26] R. Akter, V.-S. Doan, G. B. Tunze, J.-M. Lee, and D.-S. Kim, "RF-Based UAV Surveillance System: A Sequential Convolution Neural Networks Approach," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, South Korea, 2020, pp. 555–558. DOI: 10.1109/ICTC49870.2020.9289281

[27] R. Akter, M. Golam, A. Zainudin, V.-S. Doan, and D.-S. Kim, "RF Signal-Based Multipurpose UAV Surveillance System Using Deep Neural Network," in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, South Korea, 2022, pp. 555–559. DOI: 10.1109/ICTC55196.2022.9952604

[28] E. Ozturk, F. Erden, and I. Guvenc, "RF-Based Low-SNR Classification of UAVs Using Convolutional Neural Networks," *ITU Journal on Future and Evolving Technologies*, vol. 2, no. 5, pp. 39–52, 2021. DOI: 10.52953/QJGH3217

[29] S. Glüge, M. Nyfeler, A. Aghaebrahimian, N. Ramagnano, and C. Schüpbach, "Robust Low-Cost Drone Detection and Classification Using Convolutional Neural Networks in Low SNR Environments," *IEEE Journal of Radio Frequency Identification*, vol. 8, pp. 821–830, 2024. DOI: 10.1109/JRFID.2024.3487303

[30] K. Peffers *et al.*, "Design Science Research Process: A Model for Producing and Presenting Information Systems Research," 2020. DOI: 10.48550/arXiv.2006.02763 arXiv: 2006.02763 [cs.SE].

[31] Z. Shi, X. Chang, C. Yang, Z. Wu, and J. Wu, "An Acoustic-Based Surveillance System for Amateur Drones Detection and Localization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2731–2739, 2020. DOI: 10.1109/TVT.2020.2964110

[32] M. M. Azari, H. Sallouha, A. Chiumento, S. Rajendran, E. Vinogradov, and S. Pollin, "Key Technologies and System Trade-offs for Detection and Localization of Amateur Drones," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 51–57, 2018. DOI: 10.1109/MCOM.2017.1700442

[33] Z. Liu, P. An, Y. Yang, S. Qiu, Q. Liu, and X. Xu, "Vision-Based Drone Detection in Complex Environments: A Survey," *Drones*, vol. 8, no. 11, p. 643, 2024. DOI: 10.3390/drones8110643

[34] S. R. Ganti and Y. Kim, "Implementation of detection and tracking mechanism for small UAS," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, Arlington, VA, USA, 2016, pp. 1254–1260. DOI: 10.1109/ICUAS.2016.7502513

[35] J. S. Patel, F. Fioranelli, and D. Anderson, "Review of radar classification and RCS characterisation techniques for small UAVs or drones," *IET Radar, Sonar & Navigation*, vol. 12, no. 9, pp. 911–919, 2018. DOI: 10.1049/iet-rsn.2018.0020

[36] P. Nguyen, H. Truong, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu, "Cost-Effective and Passive RF-Based Drone Presence Detection and Characterization," *GetMobile: Mobile Computing and Communications*, vol. 21, no. 4, pp. 30–34, 2018. DOI: 10.1145/3191789.3191800

[37] T. Ulversoy, "Software Defined Radio: Challenges and Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 531–550, 2010. DOI: 10.1109/SURV.2010.032910.00019

[38] R. Akeela and B. Dezfouli, "Software-defined Radios: Architecture, state-of-the-art, and challenges," *Computer Communications*, vol. 128, pp. 106–125, 2018. DOI: 10.1016/j.comcom.2018.07.012

[39] HackRF Project, "HackRF One," HackRF's documentation, Accessed: Jan. 16, 2025. [Online]. Available: https://hackrf.readthedocs.io/en/latest/hackrf_one.html

[40] Great Scott Gadgets, "HackRF One," Great Scott Gadgets, Accessed: Jan. 16, 2025. [Online]. Available: https://greatscottgadgets.com/hackrf/one/

[41] National Instruments, "NI Ettus USRP Software Defined Radios," NI, Product Flyer. Accessed: Mar. 17, 2025. [Online]. Available: https://www.ni.com/pdf/product-flyers/usrp-software-defined-radios.pdf

[42] National Instruments, "USRP-2901 Specifications," NI, May 17, 2023. Accessed: Mar. 17, 2025. [Online]. Available: https://www.ni.com/docs/bundle/usrp-2901-specs/page/specs.html

[43] Ettus Research, "USRP B210 SDR Kit," Ettus Research, Accessed: Mar. 17, 2025. [Online]. Available: https://www.ettus.com/all-products/UB210-KIT/

[44] Analog Devices, "ADALM-PLUTO Overview," Analog Devices Wiki, Oct. 22, 2021. Accessed: Feb. 18, 2025. [Online]. Available: https://wiki.analog.com/university/tools/pluto

[45] Analog Devices, “ADALM-PLUTO SDR Active Learning Module,” Analog Devices, Accessed: Feb. 18, 2025. [Online]. Available: https://www.analog.com/en/resources/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html

[46] Analog Devices, “AD9363 RF Agile Transceiver Datasheet,” Analog Devices, Data Sheet AD9363, Nov. 2016, Rev. D. Accessed: Feb. 18, 2025. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD9363.pdf

[47] Nuand, “bladeRF USB 3.0 Software Defined Radio Brief,” Nuand, Product Flyer. Accessed: Mar. 28, 2025. [Online]. Available: https://www.nuand.com/bladeRF-brief.pdf

[48] Nuand, “bladeRF x40,” Nuand, Accessed: Mar. 28, 2025. [Online]. Available: https://www.nuand.com/product/bladerf-x40/

[49] Lime Microsystems, “LimeSDR Mini 2.0,” Lime Microsystems, Accessed: Mar. 19, 2025. [Online]. Available: https://limemicro.com/sdr/limesdr-mini-2-0/

[50] gnuradio, “gnuradio, GNU Radio – the Free and Open Software Radio Ecosystem,” GitHub Repository, version 3.10.12.0, Accessed: Apr. 24, 2025. [Online]. Available: https://github.com/gnuradio/gnuradio

[51] GNU Radio, “GNU Radio – Main Page,” GNU Radio Wiki, Apr. 17, 2024. Accessed: Apr. 24, 2025. [Online]. Available: https://wiki.gnuradio.org/index.php/Main_Page

[52] GNU Radio, “GNU Radio – What Is GNU Radio,” GNU Radio Wiki, Dec. 4, 2022. Accessed: Apr. 24, 2025. [Online]. Available: https://wiki.gnuradio.org/index.php/What_Is_GNU_Radio

[53] GNU Radio, “GNU Radio – OutOfTreeModules,” GNU Radio Wiki, Feb. 3, 2023. Accessed: Apr. 24, 2025. [Online]. Available: https://wiki.gnuradio.org/index.php/OutOfTreeModules

[54] GNU Radio, “GNU Radio – Hardware,” GNU Radio Wiki, Dec. 13, 2022. Accessed: Apr. 24, 2025. [Online]. Available: https://wiki.gnuradio.org/index.php/Hardware

[55] MathWorks, “Simulink – Simulation and Model-Based Design,” MathWorks, Accessed: Apr. 28, 2025. [Online]. Available: https://www.mathworks.com/products/simulink.html

[56] MathWorks, “Model-Based Design with Simulink – MATLAB & Simulink,” MATLAB Help Center, Accessed: Apr. 28, 2025. [Online]. Available: https://www.mathworks.com/help/simulink/gs/model-based-design.html

[57] MathWorks, "Simulink – Documentation," MATLAB Help Center, Accessed: Apr. 28, 2025. [Online]. Available: https://www.mathworks.com/help/simulink/index.html

[58] MathWorks, "Supported Hardware – Software-Defined Radio – MATLAB & Simulink," MATLAB Help Center, Accessed: Apr. 28, 2025. [Online]. Available: https://www.mathworks.com/help/comm/supported-hardware-software-defined-radio.html

[59] A. Ng, K. Katanforoosh, and Y. B. Mourri, "Deep Learning Specialization, DeepLearning.AI," Coursera, 2021. Accessed: Aug. 26, 2024. [Online]. Available: https://www.coursera.org/specializations/deep-learning

[60] F. N. M. Zamri, T. S. Gunawan, M. Kartiwi, A. Pratondo, S. H. Yusoff, and Y. M. Mustafah, "Deep Learning Techniques for Advanced Drone Detection Systems: A Comprehensive Review of Techniques, Challenges and Future Directions," *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, vol. 12, no. 4, pp. 818–857, 2024. DOI: 10.52549/ijeei.v12i4.6028

[61] N. Al-lQubaydhi *et al.*, "Deep learning for unmanned aerial vehicles detection: A review," *Computer Science Review*, vol. 51, p. 100 614, 2024. DOI: 10.1016/j.cosrev.2023.100614

[62] M. Dorofki, A. Elshafie, O. Jaafar, and O. A. Karim, "Comparison of Artificial Neural Network Transfer Functions Abilities to Simulate Extreme Runoff Data," vol. 33, pp. 39–44, 2012. DOI: 10.7763/IPCBEE.2012.V33.8

[63] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735

[64] X. Wu, B. Xiang, H. Lu, C. Li, X. Huang, and W. Huang, "Optimizing Recurrent Neural Networks: A Study on Gradient Normalization of Weights for Enhanced Training Efficiency," *Applied Sciences*, vol. 14, no. 15, p. 6578, 2024. DOI: 10.3390/app14156578

[65] D. M. L. Casaleiro, "Molecular Communication Schemes for Extreme Environments in Future Wireless Networks," M.S. thesis, Dept. of Inf. Sci. and Tech., Iscte – Univ. Inst. of Lisbon, Lisbon, Portugal, 2024. Accessed: Jan. 16, 2025. [Online]. Available: http://hdl.handle.net/10071/34216

[66] L. Yan, "UAV Detection with Radio Frequency Data and Deep Learning Techniques," in *2024 IEEE 6th International Conference on Civil Aviation Safety and Information Technology*

(ICCASIT), Hangzhou, China, 2024, pp. 234–239. DOI: 10.1109/ICCASIT62299.2024.10827957

[67] C. J. Swinney and J. C. Woods, "Low-Cost Raspberry-Pi-Based UAS Detection and Classification System Using Machine Learning," *Aerospace*, vol. 9, no. 12, p. 738, 2022. DOI: 10.3390/aerospace9120738

[68] O. O. Medaiyese, A. Syed, and A. P. Lauf, "Machine Learning Framework for RF-Based Drone Detection and Identification System," in *2021 2nd International Conference On Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*, Tangerang, Indonesia, 2021, pp. 58–64. DOI: 10.1109/ICON-SONICS53103.2021.9617168

[69] Z. Cai, Y. Wang, Q. Jiang, G. Gui, and J. Sha, "Toward Intelligent Lightweight and Efficient UAV Identification With RF Fingerprinting," *IEEE Internet of Things Journal*, vol. 11, no. 15, pp. 26 329–26 339, 2024. DOI: 10.1109/JIOT.2024.3395466

[70] Q. Wang, P. Yang, X. Yan, H.-C. Wu, and L. He, "Radio Frequency-Based UAV Sensing Using Novel Hybrid Lightweight Learning Network," *IEEE Sensors Journal*, vol. 24, no. 4, pp. 4841–4850, 2024. DOI: 10.1109/JSEN.2023.3346209

[71] L. Ma, B. Lian, Y. Liu, H. Li, Q. Wang, and J. Zhang, "UAV Fast Signal Detection Algorithm with Convolutional Neural Network," in *2022 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Xi'an, China, 2022, pp. 1–6. DOI: 10.1109/ICSPCC55723.2022.9984389

[72] C. Zhao, C. Chen, Z. He, and Z. Wu, "Application of Auxiliary Classifier Wasserstein Generative Adversarial Networks in Wireless Signal Classification of Illegal Unmanned Aerial Vehicles," *Applied Sciences*, vol. 8, no. 12, p. 2664, 2018. DOI: 10.3390/app8122664

[73] F. B. Mehouachi *et al.*, "Detection of UAVs Based on Spectrum Monitoring and Deep Learning in Negative SNR Conditions," *URSI Radio Science Letters*, vol. 3, 2021. DOI: 10.46620/21-0043

[74] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Detection and Classification of UAVs Using RF Fingerprints in the Presence of Wi-Fi and Bluetooth Interference," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 60–76, 2020. DOI: 10.1109/ojcoms.2019.2955889

[75] M. Al-Sa'd, M. S. Allahham, A. Mohamed, A. Al-Ali, T. Khattab, and A. Erbad, *DroneRF dataset: A dataset of drones for RF-based detection, classification, and identification*, Mendeley Data, version 1, 2019. DOI: 10.17632/f4c2b4n755.1

[76] Parrot, "Parrot Bebop Drone – User Guide," Parrot, User Guide, Sep. 2021. Accessed: Mar. 18, 2025. [Online]. Available: https://www.parrot.com/assets/s3fs-public/2021-09/bebop-drone_user-guide_uk_v.3.4.pdf

[77] Parrot, "Parrot AR.Drone 2.0 Elite Edition – User Guide," Parrot, User Guide, Sep. 2021. Accessed: Mar. 18, 2025. [Online]. Available: https://www.parrot.com/assets/s3fs-public/2021-09/ar.drone2_user-guide_uk.pdf

[78] DJI, "DJI Phantom 3 Standard – Support page," DJI, 2015. Accessed: Mar. 18, 2025. [Online]. Available: https://www.dji.com/pt/support/product/phantom-3-standard

[79] Al-Sad, "DroneRF," GitHub Repository, Accessed: Sep. 18, 2024. [Online]. Available: https://github.com/Al-Sad/DroneRF

[80] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020. DOI: 10.1038/s41586-020-2649-2

[81] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," 2015. Accessed: Dec. 2, 2024. [Online]. Available: https://www.tensorflow.org/

[82] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[83] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, 2018, pp. 1091–1100. DOI: 10.1109/CVPR.2018.00120

[84] X. Lei, H. Pan, and X. Huang, "A Dilated CNN Model for Image Classification," *IEEE Access*, vol. 7, pp. 124 087–124 095, 2019. DOI: 10.1109/ACCESS.2019.2927169

[85] D. Casaleiro, N. M. B. Souto, and J. C. Silva, "Synchronisation and Detection in Molecular Communication using a Deep-Learning-based Approach," *IEEE Access*, vol. 12, pp. 192 539–192 553, 2024. DOI: 10.1109/ACCESS.2024.3519310

[86] Y. He and J. Zhao, "Temporal Convolutional Networks for Anomaly Detection in Time Series," *Journal of Physics: Conference Series*, vol. 1213, no. 4, p. 042 050, 2019. DOI: 10.1088/1742-6596/1213/4/042050

[87] J. Liu, M. Li, Y. Luo, S. Yang, W. Li, and Y. Bi, "Alzheimer's disease detection using depthwise separable convolutional neural networks," *Computer Methods and Programs in Biomedicine*, vol. 203, p. 106 032, 2021. DOI: 10.1016/j.cmpb.2021.106032

[88] K. Kc, Z. Yin, M. Wu, and Z. Wu, "Depthwise separable convolution architectures for plant disease classification," *Computers and Electronics in Agriculture*, vol. 165, p. 104 948, 2019. DOI: 10.1016/j.compag.2019.104948

[89] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. DOI: 10.1016/j.ipm.2009.03.002

[90] Ž. Đ. Vujovic, "Classification Model Evaluation Metrics," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021. DOI: 10.14569/IJACSA.2021.0120670

[91] X. Zeng and T. R. Martinez, "Distribution-balanced stratified cross-validation for accuracy estimation," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 12, no. 1, pp. 1–12, 2000. DOI: 10.1080/095281300146272

[92] M. A. M. Romana, "RF-Based Drone Detection: A Deep Learning approach," unpublished.

[93] Y. Lin, I. Koprinska, and M. Rana, "Temporal Convolutional Attention Neural Networks for Time Series Forecasting," in *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 2021. DOI: 10.1109/ijcnn52387.2021.9534351

[94] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017, pp. 2392–2396. DOI: 10.1109/ICASSP.2017.7952585

[95] S. Al-Emadi, A. Al-Ali, A. Mohammad, and A. Al-Ali, "Audio Based Drone Detection and Identification using Deep Learning," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Tangier, Morocco, 2019, pp. 459–464. DOI: 10.1109/IWCMC.2019.8766732

APPENDIX A

Additional Results

This section provides further results concerning additional simulation tests.

A.1. Detection using Dilation Factor 2

In particular, the following results pertain to the experiment regarding detection using a dilation factor of 2, namely when applied to the layers as described in section 4.2.1.

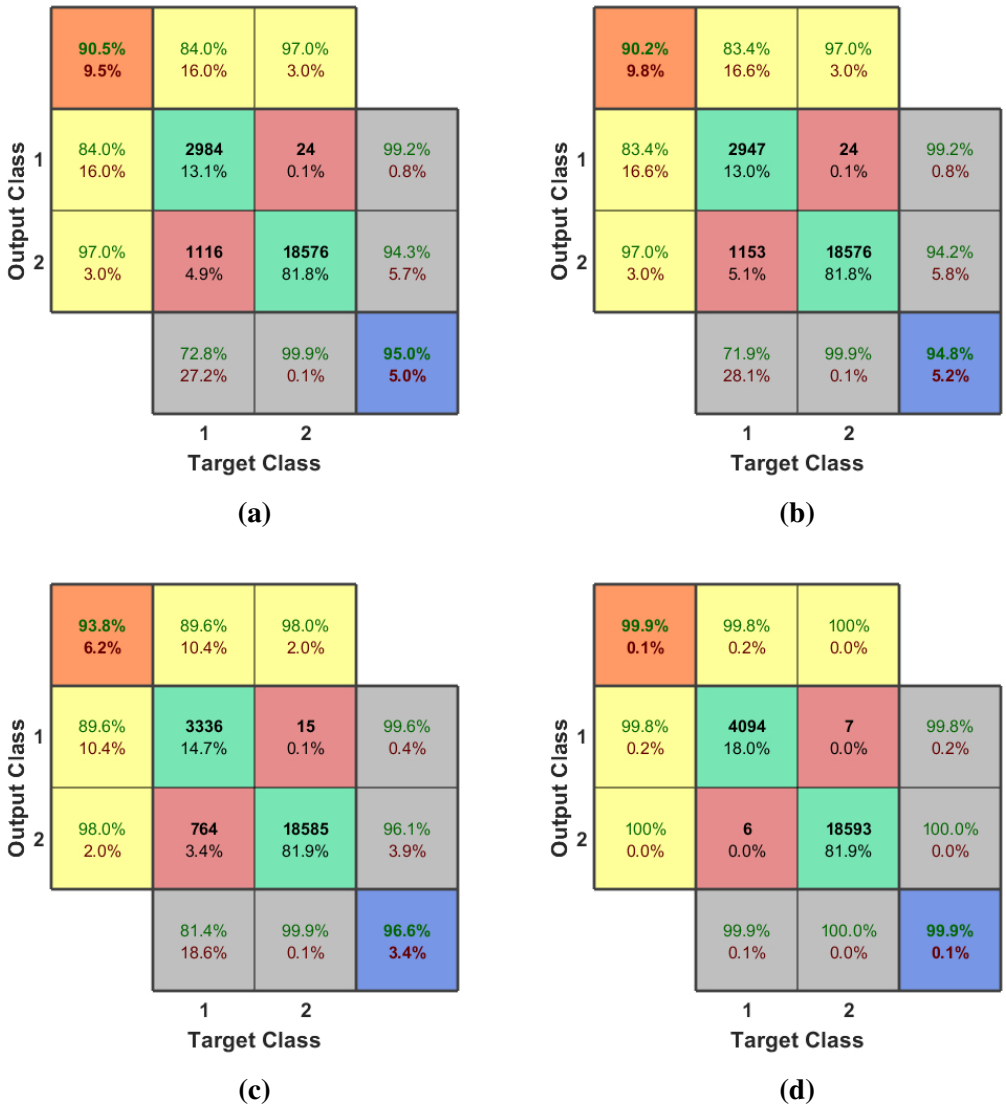


Fig. A.1. Confusion Matrices of remaining results, regarding Detection using Dilation Rate 2, as applied: exclusively to layer (a) 1, (b) 2, (c) 3, and (d) 4.

A.2. Detection using Dilation Factor 3

Moreover, the ensuing results refer to the experiment concerning detection using a dilation factor of 3, when applied to the layers as expressed in section 4.2.2.

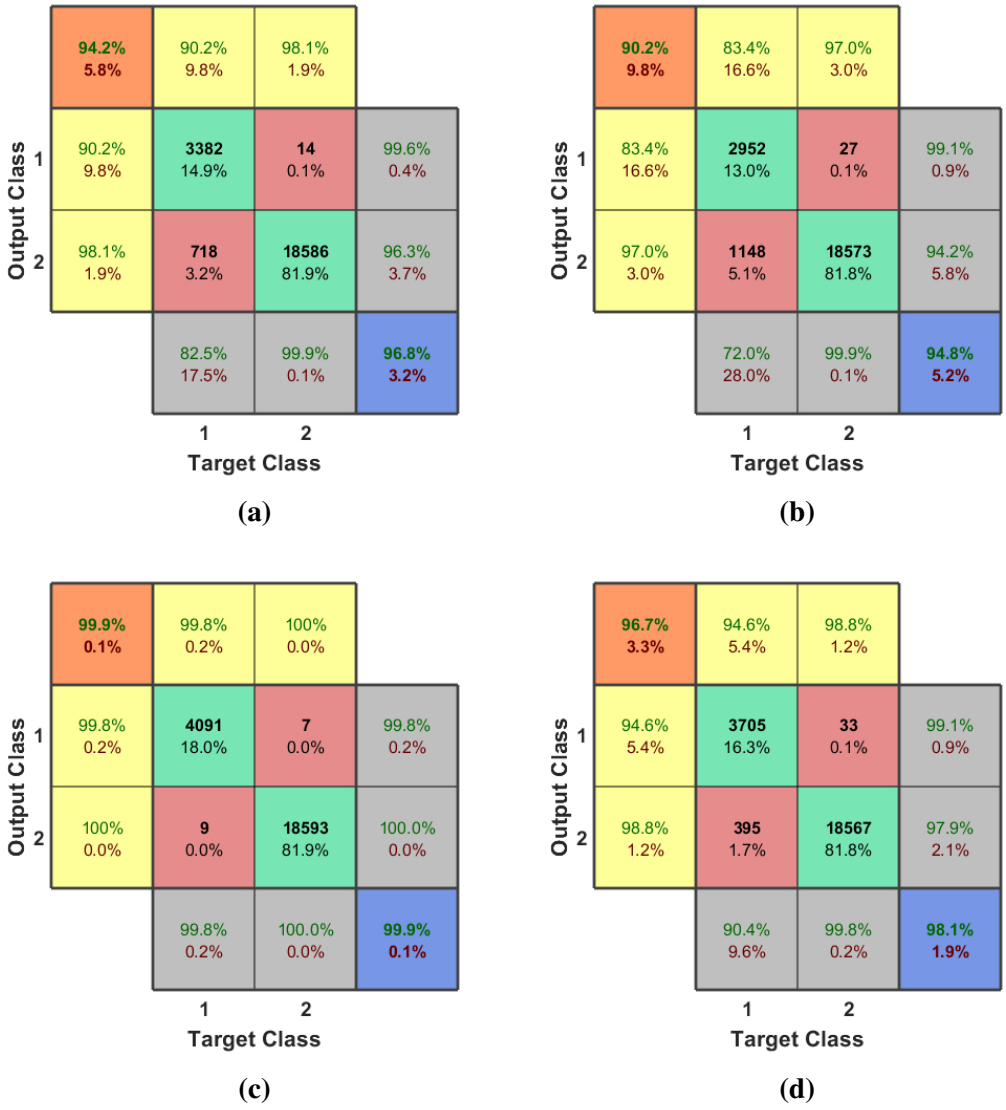


Fig. A.2. Confusion Matrices of remaining results, regarding Detection using Dilation Rate 3, as applied: exclusively to layer (a) 1, (b) 2, and (d) 4; and (c) across all layers simultaneously.

A.3. Detection using Depthwise Separable Convolutions

Furthermore, the succeeding results concern the experiment regarding detection using DSCs, and applied to the layers as described in section 4.3.1.

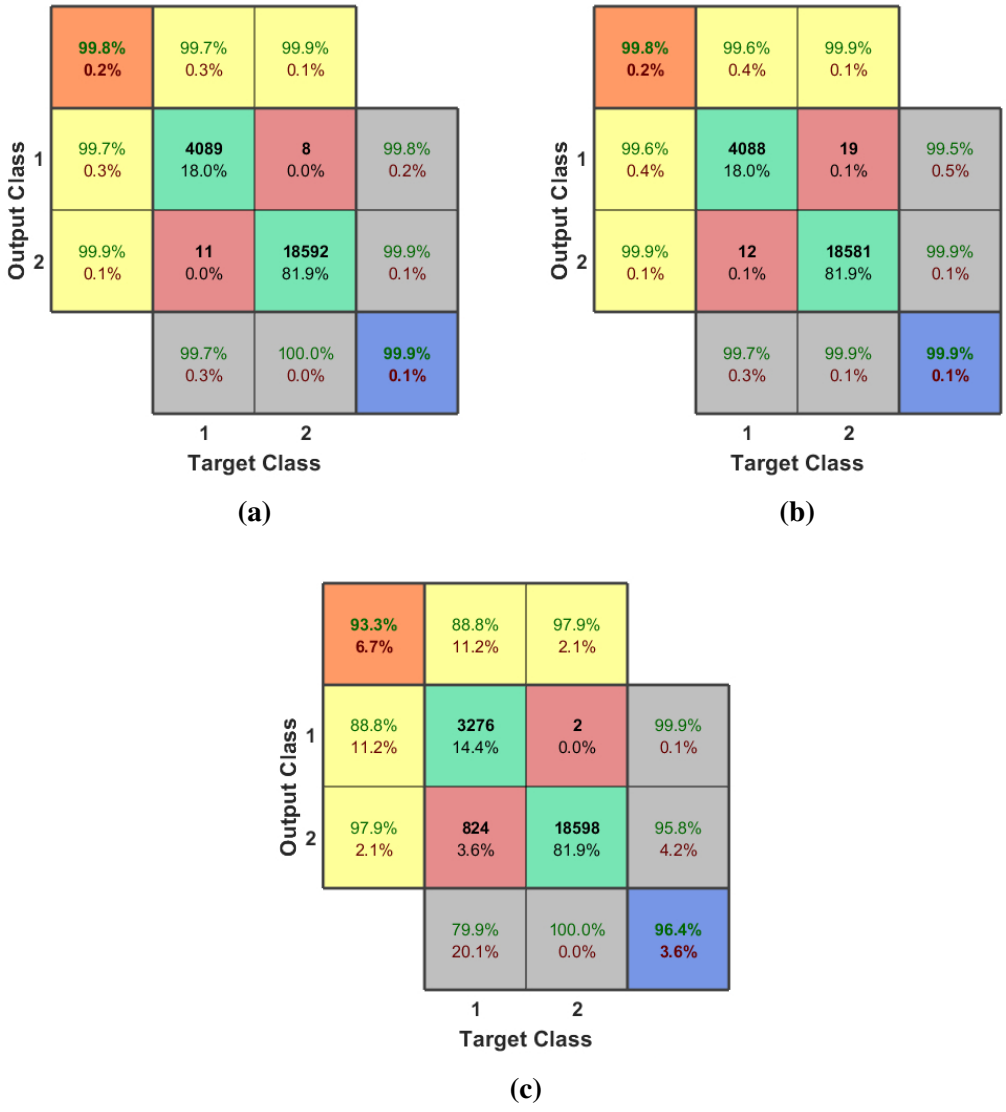


Fig. A.3. Confusion Matrices of remaining results, regarding Detection using Depthwise Separable Convolutions, as applied: exclusively to (a) layer 4, and (b) layers 4, 3 and 2; and (c) across all layers simultaneously.

APPENDIX B

Research Article

This section exclusively contains and presents the research article [92], planned for submission to a peer-reviewed journal, for scientific contribution; and mainly derived in support from this thesis. In particular, only the first page is reproduced.

RF-Based Drone Detection: A Deep Learning approach

Miguel A. M. Romana[1] Nuno M. B. Souto[2,3], Renato Ferreira[2,3]
Iscte – University Institute of Lisbon, Lisbon, Portugal
[1]Telecommunications and Computer Engineering
[2]Instituto de Telecomunicações
[3]Department of Information Science and Technology
Corresponding author: Miguel Romana (e-mail: miguel_romana@iscte-iul.pt)

***Abstract*—Drones have become widely adopted across numerous domains due to their accessibility and versatility; however, associated incidents have raised significant security and privacy concerns, consequently motivating the development of accurate, efficient, and reliable solutions devised to recognise Unmanned Aerial Vehicles (UAV). This research proposes a drone detection system capable of detecting the presence, as well as identifying the types of drones, by utilising captured Radio Frequency (RF) signals, combined with a Deep Learning (DL) approach known as Convolutional Neural Network (CNN), particularly employing one-dimensional (1-D) layers. The system leverages an existing database composed of recorded drone RF signals from several drones, along with background noise, for UAV classification. Various experiments were investigated, focusing on the impact of dilation factors and depthwise separable convolutions; performance is evaluated through confusion matrices and standard metrics, including accuracy, precision, recall, and F1-score. Results demonstrate near-perfect detection of drones, achieving accuracy of approximately 100%, complemented by great performance identification of drone types, with accuracy exceeding 75%. Furthermore, a comparative analysis with the literature highlights the effectiveness of the proposed approach, achieving superior or similar results, whilst with preserved or reduced complexity.**

***Index Terms*—Drone detection, UAV identification, Deep Learning, Neural Networks, RF signals.**

I. Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are aircraft that operate without a human pilot on board. They can be remotely controlled by an operator, or function autonomously following pre-programmed flight paths, utilising onboard sensors along with advanced technology for navigation, control and perception [1], [2].

Although initially implemented for military applications, drones have become integral to modern technology, and are now widely used in various civilian and commercial sectors, ranging in applications from aerial cinematography and parcel delivery to environmental monitoring and emergency response operations, among a plethora of others [1], [3]–[7].

However, the growing accessibility and proliferation of drones has raised concerns for security and privacy. Unauthorised drones can be used for malicious activities, especially in restricted areas, for instance to interfere with air traffic, or violate personal privacy via surveillance [8]–[10].

Tragic episodes have highlighted the dangers posed by unauthorised drones, particularly airport disruptions caused by rogue UAVs, which have endangered lives and resulted in significant financial losses [8]. A notable example is the Gatwick Airport drone incident of 2018, in which the airport runway had to be closed in order to prevent the risk of collision with aircraft [11], [12]; moreover, drone usage has extended from civilian disruptions to active warfare, as evidenced in the Russia–Ukraine conflict initiated in 2022, where UAVs have become pivotal offensive and surveillance tools which have redefined strategic operations on the battlefield [13], [14].

To address these challenges, there is a pressing need to develop drone detection and identification systems that can trigger and apply appropriate countermeasures to unauthorised drones in real-time when necessary, ensuring safe and responsible integration of drones into the airspace [15], [16].

Various detection techniques have been developed to achieve UAV detection, with anti-drone systems employing either a single strategy or a combination of approaches [1], including acoustic, visual, radar, and radio detection methods, each with its own advantages and limitations [1], [3], [15], [17]. Acoustic methods are sensitive to noise and range and therefore cannot accurately detect drones equipped with nearly silent rotors or emitting masking and concealing sounds [1]. Visual methods are hindered by scenarios that hide or disguise the UAVs, such as illumination, design or structural obstructions; and adverse weather conditions, particularly storms, rain and fog [18]. Radar approaches – similarly to visual techniques – are compromised by cluttered environments, and the physical characteristics and flight behaviour of the drone; these methods usually struggle differentiating from birds with small or composite-built drones, or when they are hovering [16], [19]. Even though Radio Frequency (RF)-based detection is not without vulnerabilities – such as signal interference, for example – it has proven more resistant and robust to misleading and deceptive tactics [1].

Deep Learning (DL) represents a field of Machine Learning (ML) which is particularly effective in processing and analysing complex data patterns, and identifying non-linear relationships and hidden structures that would otherwise be too difficult or even impossible to capture through conventional methods such as manual feature selection [20], [21]. Thus, given the nature of drone RF signals, this approach proves ad-