Smart Sensing and Communications for UAVs/Drones

Wilson Ayyanthole Nelson

UiA University of Agder

Doctoral Dissertations at the University of Agder 485

Wilson A N

Smart Sensing and Communications for UAVs/Drones

Doctoral Dissertation for the Degree *Philosophiae Doctor (Ph.D.)*
at the Faculty of Engineering and Science, Specialisation in Information and Communication Technology

University of Agder
Faculty of Engineering and Science
2024

Doctoral Dissertations at the University of Agder 485
ISSN: 1504-9272
ISBN: 978-82-8427-202-3

Printed by Make!Graphics
Kristiansand

Preface

This dissertation marks the culmination of my research performed at the University of Agder, Grimstad campus between January 2021 and January 2024. I acknowledge the encouragement and assistance provided by my supervisors Professor Linga Reddy Cenkeramaddi (lead advisor), Associate Professor Ajit Jha, and Associate Professor Abhinav Kumar. I would also like to acknowledge all other collaborators and participants for their valuable contributions and support that have helped bring this dissertation to fruition.

This thesis is based on seven publications that are provided in the Appendix section. Papers A and B are literature surveys that have been carried out in conjunction with the research presented in subsequent publications. Papers C, D, and E provide practical approaches to realize reliable sensing for UAVs and ground control stations. The remaining publications, Papers F and G involve the design of an adaptively optimized switching for communication modules on board UAVs. The dissertation comprises of five chapters. Chapter 1 provides an introduction to the research presented in the thesis. Chapter 2 discusses necessary background material and state-of-the-art. Chapter 3 focuses on the proposed approaches concerning UAV sensing. Chapter 4 summarizes the design of an adaptive switching scheme for UAV communication. Finally, Chapter 5 marks the conclusion of the dissertation, providing a summary and holistic view of the research, and future directions.

This Ph.D. dissertation is a testament to the hardships, endurance, and sacrifice that I faced during the tenure of the Ph.D. journey. It also serves as a reminder that strict discipline enforced with hard work, dedication, and proper planning are crucial to achieving one's dreams. I am deeply grateful for all the lessons I have acquired throughout this journey and wish to rely on them in my future endeavours.

I believe that the outcomes of this research that is forged with hard work, dedication, and commitment will serve to be meaningful contributions to the scientific community.

Acknowledgments

First, and foremost, I would like to thank God almighty, for granting me this opportunity. The countless blessings and knowledge showered upon me have truly aided in the completion of this dissertation.

The success of this thesis largely depends on the efforts and commitment of my Ph.D. supervisor Professor Linga Reddy Cenkeramaddi. His immense knowledge, exceptional guidance, and mentorship have contributed significantly to the timely completion of this Ph.D. thesis. I would like to express my greatest appreciation for his constant motivation, unwavering support, and regular feedback. His continuous efforts and dedication have helped shape the structure of this Ph.D. dissertation. I would also like to extend my profound gratitude to my co-supervisors, Associate Professor Ajit Jha and Associate Professor Abhinav Kumar for their constructive feedback, encouraging discussions, and motivation that has helped me navigate the challenges of this research.

I would like to convey my heartfelt gratitude to Dr. Yeduri Sreenivasa Reddy for his invaluable suggestions, inspirational discussions, and constructive criticisms that have motivated me to think beyond my current capabilities. I would also like to acknowledge all the collaborators and participants whose support and contributions have had a significant impact on the formation of this Ph.D. dissertation. My sincere gratitude to all the members of the REC committee for their adequate and invaluable feedback in matters related to the Ph.D. research.

I am thankful to the Head of the Department, Folke Haugland for providing all the necessary infrastructure and research facilities. Special thanks to Emma E. Horneman, Katharina Pätzold, and Sissel Stenersen for their assistance in all the logistics related to the Ph.D. process at UiA. I would also like to acknowledge the immense support provided by all the administrative staff and members of UiA Hjelp.

I was fortunate to forge lasting friendships during my stay in Grimstad. My deepest gratitude to Talha Bilal, Sreenivasa Reddy, Amala, Sriram Reddy, Mouni, Michal, Sindhusha, Balu, Khushi, Vishnu, Aveen, Naveen, José Amendola, and Touseef for making my life at the F-building vibrant and cheerful. Special thanks to Nikhil and Ashfaq for their continuous calls and motivation during the difficult times of the Ph.D. journey.

This Ph.D. thesis would not have been possible without the unwavering love, support, and trust of my family. They have been with me through thick and thin and offered invaluable advice during my time of need. A special shoutout to my sister who has been instrumental in motivating me throughout this journey. Her

regular suggestions concerning my physical and mental well-being have truly helped to improve my health and productivity during this time. My sincere appreciation to my grandparents for their timely calls and inspiring discussions that have helped me take on the challenges encountered during this endeavor. Finally, I would also like to thank everyone who have contributed directly or indirectly to the success of this dissertation.

Wilson A N
Grimstad, Norway
28/ 8/ 2024

Sammendrag

Pålitelig sensing og energieffektiv kommunikasjon er en integrert del av UAV-baserte applikasjoner. Det er imidlertid utfordrende å sikre pålitelig sensing under alle værforhold. Dessuten er det like krevende å muliggjøre robust og energieffektiv kommunikasjon for sømløse autonome operasjoner. Lav-kompleksitet og lette algoritmer spiller en viktig rolle i denne forbindelse. Som et resultat av dette ble målet for denne Ph.D. Forskningsprosjektet er å designe, utvikle og implementere pålitelige sensing og energieffektive kommunikasjonsteknikker for å gi sømløse autonome operasjoner for UAV-er og bakkekontrollstasjoner. For å oppnå det tiltenkte målet, vil et av målene være rettet mot å utvikle nye lette, lavkompleksitets smarte sensoralgoritmer som oppnår forbedret deteksjonsytelse under ugunstige vær- og lysforhold. I denne forbindelse fokuserer denne forskningen på å bruke termiske, akustiske og mmWave FMCW-radarsensorer med lette maskinlæringsalgoritmer for å oppnå forbedret deteksjons- og lokaliseringsytelse. Den registrerte informasjonen sammen med telemetridata må overføres til andre UAV-er eller bakkekontrollstasjoner for å lette effektiv og sikker koordinering av autonome UAV-er. Derfor vil et tilleggsmål med forskningen innebære utformingen av en dynamisk og adaptivt optimalisert svitsjalgoritme for de forskjellige kommunikasjonsprotokollene utstyrt på UAV-er. Forskningen vil først og fremst ta for seg utfordringen rundt energieffektiv dataoverføring for et UAV-assistert trådløst nettverk.

Abstract

Reliable sensing and energy-efficient communication are an integral part of unmanned aerial vehicle (UAV)-based applications. However, it is challenging to ensure reliable sensing under all weather conditions. Furthermore, enabling robust and energy-efficient communication of this sensed data for seamless autonomous operations is equally demanding. Low-complexity and lightweight algorithms play a vital role in this regard. As a result, the goal of this Ph.D. research project is to design, develop, and implement reliable sensing and energy-efficient communication techniques to provide seamless autonomous operations for UAVs and ground control stations. To achieve the intended goal, one of the objectives is directed toward developing novel lightweight low-complexity smart sensing algorithms that attain improved detection performance in adverse weather and lighting conditions too. In this regard, this research focuses on utilizing thermal, acoustic, and mmWave frequency-modulated continuous wave (FMCW) radar sensors with lightweight machine learning algorithms to obtain improved detection and localization performance. The sensed information along with telemetry data is required to be transmitted to other UAVs or ground control stations to facilitate efficient and secure coordination of autonomous UAVs. Hence, the secondary objective of this research addresses the challenge of achieving energy-efficient data transmission for a UAV-assisted wireless network. The research for attaining energy-efficient data transmission focuses on the design of a dynamic and adaptively optimized switching algorithm that leverages the distinct features of the various onboard communication modules.

This thesis is a compendium of seven publications that are organized into five chapters. Overall the primary objective is to develop reliable smart sensing algorithms along with energy-efficient communication schemes to enable seamless autonomous UAV operations. The first chapter briefly introduces the various challenges dealt in the thesis and provides an overview of the thesis structure. Chapter 2 provides the necessary literary material to navigate the subsequent chapters. Chapter 3 discusses three sensing approaches using thermal, acoustic, and mmWave FMCW radar sensors to achieve improved detection and localization performance. In chapter 4, the notion of an adaptively optimized switching algorithm for energy-efficient communication is conceived. Two approaches to form the hybrid communication switching network are discussed. Finally, chapter 5 concludes the Ph.D. dissertation by summarizing the proposed approaches and providing avenues for future research.

Publications

The following publications are included in this dissertation and are provided in Appendices, A-G at the end of the Ph.D. Thesis.

- PAPER A: **A. N. Wilson**, A. Kumar, A. Jha and L. R. Cenkeramaddi, "Embedded Sensors, Communication Technologies, Computing Platforms and Machine Learning for UAVs: A Review," *IEEE Sensors Journal*, vol. 22, no. 3, pp. 1807-1826, Feb. 2022.

- PAPER B: **A. N. Wilson**, K. A. Gupta, B. H. Koduru, A. Kumar, A. Jha and L. R. Cenkeramaddi, "Recent Advances in Thermal Imaging and its Applications Using Machine Learning: A Review," *IEEE Sensors Journal*, vol. 23, no. 4, pp. 3395-3407, Feb. 2023.

- PAPER C: **W. A. N.**, A. Jha, A. Kumar and L. R. Cenkeramaddi, "Estimation of UAV Count Using Thermal Imaging and Lightweight CNN," in *Proc. ICCMA*, Grimstad, Norway, 2023, pp. 92-96.

- PAPER D: **Wilson A. N.**, Ajit Jha, Abhinav Kumar, Linga Reddy Cenkeramaddi, "Estimation of number of unmanned aerial vehicles in a scene utilizing acoustic signatures and machine learning," *J. Acoust. Soc. Am.*, vol. 154, no. 1, pp. 533–546, Jul. 2023.

- PAPER E: **A. N. Wilson**, A. Kumar, A. Jha and L. R. Cenkeramaddi, "Multitarget Angle of Arrival Estimation Using Rotating mmWave FMCW Radar and Yolov3," *IEEE Sensors Journal*, vol. 23, no. 3, pp. 3173-3182, Feb. 2023.

- PAPER F: **W. A. N.**, Y. S. Reddy, A. Jha, A. Kumar and L. R. Cenkeramaddi, "Hybrid BLE/LTE/Wi-Fi/LoRa Switching Scheme for UAV-Assisted Wireless Networks," in *Proc. IEEE ANTS*, Hyderabad, India, 2021, pp. 78-83.

- PAPER G: **W. A. Nelson**, S. R. Yeduri, A. Jha, A. Kumar and L. R. Cenkeramaddi, "RL-Based Energy-Efficient Data Transmission Over Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network," *IEEE/ACM Transactions on Networking.*

Contents

List of Figures

List of Tables

List of Abbreviations

ADC:	Analog-to-digital converter
AI:	Artificial intelligence
AoA:	Angle of arrival
AWGN:	Additive white gaussian noise
BLE:	Bluetooth low energy
BSLIM:	Block sparse learning via iterative minimization
BSS:	Blind source separation
CFAR:	Constant false alarm rate
CNN:	Convolutional neural networks
CRNN:	Convolutional recurrent neural networks
CSS:	Chirp spread spectrum
CWT:	Continuous wavelet transform
D2D:	Device-to-device
DADS:	Drone acoustic detection system
DL:	Deep learning
DOA:	Direction of arrival
DQN:	Deep Q-network
DRL:	Deep reinforcement learning
FAR:	False alarm rate
FFT:	Fast fourier transform
FLOPS:	Floating point operations
FMCW:	Frequency modulated continuous wave
FML:	Fast maximum likelihood
FoV:	Field of view
FS:	Free space
FSMP:	Free space and multipath
GCS:	Ground control station
GIP:	Generalised inner product
GLRT:	Generalised likelihood ratio test
GMM:	Gaussian mixture model
IF:	Intermediate frequency
IoT:	Internet-of-things
k-NN	k-nearest neighbors

LiDAR: Light detection and ranging
LoRa: Long range
LoS: Line-of-sight
LPCC: Linear predictive cepstral coefficients
LPWAN: Low power wide area network
LTE: Long term evolution
MC-IWM: Multi-class inverse wishart mixture
MFCC: Mel-frequency cepstral coefficients
OFDM: Orthogonal frequency division multiplexing
PC: Pulse compression
PCA: Principal component analysis
PIL: Plotted image learning
PSD: Power spectral density
RCNN: Region-based convolutional neural network
RF: Radio frequency
RL: Reinforcement learning
RML: Regularized maximum likelihood
RMSE: Root mean square error
RNN: Recurrent neural network
RPAS: Remotely piloted aircraft systems
SIR: Signal-to-interference ratio
SLA: Stochastic learning automata
SRP-PHAT: Steered-response phase transform
STFT: Short-time fourier transform
SVM: Support vector machine
UAV: Unmanned aerial vehicles
V2X: Vehicle-to-everything
VMD: Variational mode decomposition
VoIP: Voice over internet protocol
Wi-Fi: Wireless fidelity
YOLO: You look only once
ZCR: Zero crossing rate

Chapter 1

Introduction

The increasing demand for unmanned aerial vehicle (UAV) or drone-based applications has been significantly influenced by growing technological advancements in sensor miniaturization, ubiquitous wireless connectivity, and greater processing capacity. According to market forecasts, it is anticipated that the revenue generated by UAV-based applications will surpass 8.5 billion dollars by the year 2027 [1]. This anticipated rise can be attributed to the relaxation of restrictions pertaining to UAVs and their subsequent availability for commercial applications.

Initially employed in military applications for surveying and targeting hostile areas, UAVs have since been utilized in a diverse range of applications, spanning from agriculture to space research. UAVs are employed in agriculture for irrigation and monitoring crop vitality [2]. They are employed in defense for detecting intruders and launching attacks [3]. UAVs are utilized extensively in surveillance applications to map large regions of land [4]. UAVs equipped with remote sensing capabilities are valuable for surveying large geographical regions for archaeological studies [5]. In the e-commerce sector, Amazon has showcased a package delivery system called Prime Air [6], which utilizes UAVs to transport products to clients in remote locations within a designated time frame. Furthermore, there is an ongoing exploration of architectures based on UAVs to develop and implement advanced technologies such as 5G and vehicle-to-everything (V2X) communications.

The widespread use of UAVs across varied applications can be attributed to a multitude of factors. UAVs offer advanced airborne inspection, better line-of-sight (LoS) connectivity, reliable data acquisition, and seamless mobility. The superiority and flexibility of UAVs over ground-based systems can be attributed to their capability to access remote locations and capture images. Furthermore, the availability of advanced processing units, easily accessible components, and cost-effectiveness have further solidified the suitability of UAVs for a diverse array of applications.

Thus it can be perceived that the growing appeal for UAVs can be attributed to their capacity to effortlessly sense information from their surrounding environment. UAVs are outfitted with a diverse array of onboard sensor modules, enabling them to effectively detect and comprehend their surroundings. The sensor modules encompass a range of technologies, such as red-green-blue (RGB) cameras, infrared cameras, global positioning system (GPS) modules, radar sensors, acoustic sensors,

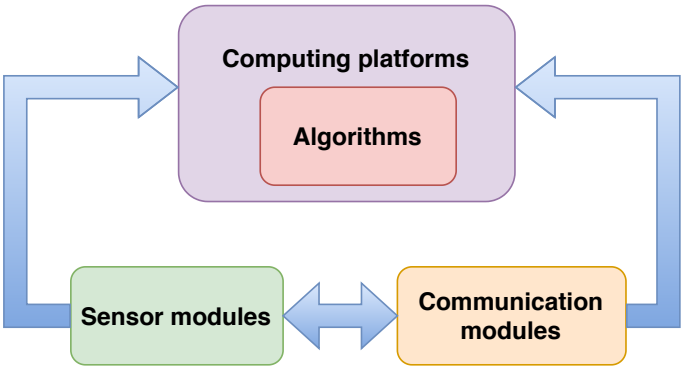


Figure 1.1: Components of a UAV system: Sensors, communication, and computing [7].

and others. Each sensor module possesses distinct features that enable them to perceive the surroundings from different perspectives. In a similar vein, UAVs transmit the acquired data to other UAVs or ground control station (GCS) to guarantee safe and secure UAV operations. The seamless connectivity is facilitated by various communication modules such as bluetooth low energy (BLE), long term evolution (LTE), wireless fidelity (Wi-Fi), Sigfox, and long range (LoRa). Hence, it is apparent that sensing and communication are of utmost importance in the functioning of a UAV system. This thesis examines the challenges related to sensing and communication in UAVs and proposes various approaches to tackle them.

1.1 Overview

In this section, a brief overview of the thesis is presented. First, we provide some background information relevant to navigate the subsequent chapters. Next, a summary of the research undertaken in this dissertation is outlined. This is followed by defining the overall structure of the thesis. Finally, in the later part of this section, we provide insight into the various challenges addressed in the thesis and their proposed solutions.

1.1.1 Components of a UAV System: Sensors, Communication, and Computing

To foster a deeper appreciation for the subject matter of this thesis, it is important to acquire a comprehensive understanding of the UAV system and its diverse modules, particularly concerning sensing and communication aspects. The UAV system can be conceptualized abstractly, as depicted in Fig. 1.1. This perspective is derived by considering the adaptability and mobility offered by different UAV components in accommodating a wide range of applications.

UAVs are outfitted with hardware sensor modules that enable them to detect and interpret the environment. Temperature and vision-based sensors play a crucial role in monitoring crop health in agricultural applications, whereas camera and light

detection and ranging (LiDAR)-based modules are typically utilized in surveillance and remote sensing. Every sensor possesses distinct characteristics in terms of its functionality, physical structure, cost, and performance. The data collected from each of these sensors is distinct and provides us with unique insights into the observed surroundings. Utilizing the data obtained from these hardware sensors and extracting essential information about the observed surroundings is a key aspect that can be found throughout the chapters of this thesis.

Onboard communication modules for UAVs aid in transmitting the sensed information and telemetry data to other UAVs or GCS. Each individual communication module functions within a predetermined frequency, bandwidth, power, and coverage range. The selection of an appropriate communication technology is determined by the application requirements, taking into account the diverse capabilities and characteristics of the technology. Wi-Fi modules are employed to transmit data between UAVs in scenarios necessitating a high data transmission rate. Conversely, LoRa modules are utilized in search and rescue operations that demand the transfer of information across long distances. The later part of the thesis examines the optimal utilization of several onboard communication systems to facilitate energy-efficient data transmission.

The integration of sensor and communication modules into UAVs is facilitated through the utilization of hardware computing platforms. Hardware computing platforms are computational devices that facilitate the processing of data from different sensors and communication modules. Further, they establish connection and serve as a medium for processing the acquired data. The selection of a computing platform is mostly decided by the specific requirements of the application. When selecting a hardware computing unit for UAV-based applications, several aspects such as processing power, interface choices, form factor, weight, and other relevant considerations are taken into account. Computing systems host a diverse range of algorithms that play a crucial role in guaranteeing the safe and secure operation of UAVs. These algorithms require low power consumption and low computational complexity. The algorithms are specifically developed to address a range of application scenarios, such as collision avoidance, control and stabilization procedures, and interference mitigation. Furthermore, computing systems have showcased to accommodate a diverse range of machine learning algorithms, which have played a crucial role in enhancing the efficiency of UAV operations.

By decoupling the UAV system into the individual blocks as illustrated in Fig. 1.1, it becomes possible to selectively analyze and learn about the various issues inherent in currently available UAV systems. Therefore, in this thesis, the following chapters specifically concentrate on these blocks. Specifically, the first part of the thesis focuses on the sensing aspects and the second part deals with the communication aspects prevalent in UAV systems.

1.1.2 Research Overview

In this subsection, we provide a comprehensive overview of the research that is presented in this thesis. UAVs cater to applications in several diverse sectors. One such application can be seen with UAV light shows. Multiple UAVs are programmed to work in tandem to provide visually appealing scenes in the night sky during various events [8]. Another scenario can be observed in disaster management situations where the use of a multi-UAV system can increase the probability of success in a search and rescue operation [9]. Fig. 1.2 provides an abstract view of how a multi-UAV system is used in a disaster management scenario for search and rescue operations of stranded individuals. The research presented in this thesis focuses on the sensing and communication of UAVs, particularly in a multi-target or multi-UAV environment.

In a multi-UAV system, individual UAVs utilize various onboard sensors to sense their surrounding environment. The sensed information can include various nearby objects present in the vicinity of the UAV. These objects can include birds, trees, aircrafts, building infrastructure, and even other UAVs. The sensed information has to be adequately processed to derive meaningful information that further helps to ensure safe and secure UAV operations. However, the detection of multiple nearby objects is a challenging task. As such detecting multiple small-sized UAVs adds further complexity. As UAVs have a small form factor, the effective radar cross-section available for UAVs is small as compared to traditional aircrafts. Hence, small-sized UAVs are undetectable using conventional aircraft detection methods. We attempt to address this issue of detecting multiple UAVs by utilizing the data from various onboard sensors and machine learning algorithms. We first propose a thermal-based approach to detect and estimate the presence of multiple UAVs. The proposed approach utilizes thermal images captured from various measurement scenarios. Each measurement scenario contains multiple small-sized UAVs maneuvering in arbitrary directions. Our proposed machine learning architecture is trained with the captured thermal images to detect and estimate the exact number of UAVs in the vicinity. Next, an acoustic-based approach was proposed. Acoustic signatures of multiple small-sized UAVs maneuvering in random directions were captured and processed. The processed information was fed to the proposed machine learning models to accurately detect and estimate the number of UAVs in the vicinity. It is shown that the proposed approach is a low-cost and portable solution that can be deployed easily on edge-computing devices. One of the limitations of the acoustic-based approach is that the detection is localized to short distances. Hence, we attempted to use the mmWave frequency modulated continuous wave (FMCW) radars to detect and localize target objects. mmWave FMCW radars provide accurate range and velocity estimates making them a prime candidate for accurate detection and localization. However, their angle of arrival estimation is poor. We proposed a rotating mmWave FMCW radar setup that can improve the angle of arrival estimation as well as classify the targets based on their range-angle signatures. The proposed approach requires a small form factor and can be integrated on UAVs and GCS.

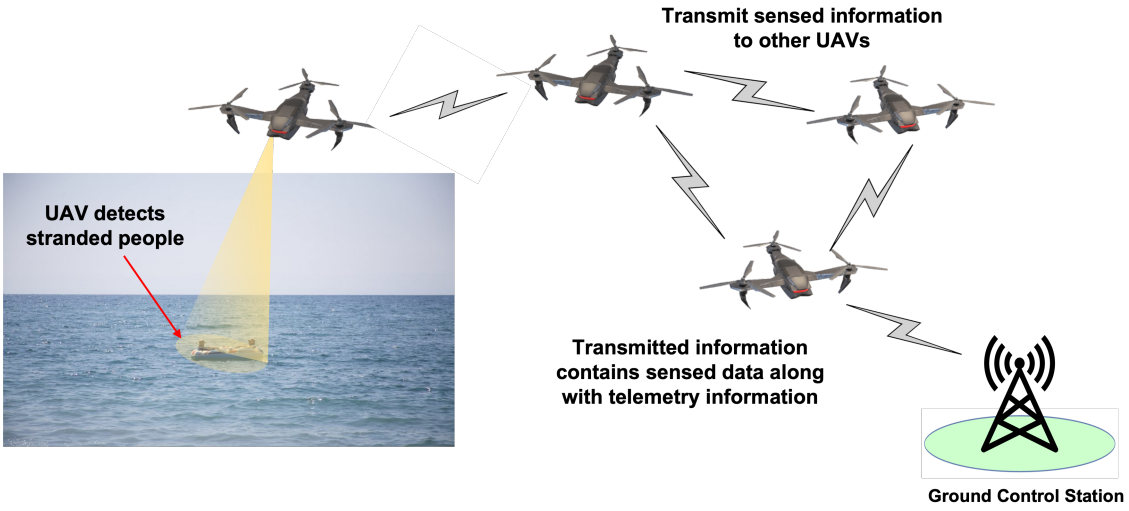


Figure 1.2: Search and rescue operation with the help of a multi-UAV system

To facilitate better decision-making and control, there is a requirement for information to be transferred between UAVs and GCS or other devices. UAVs are equipped with various communication modules that can aid in the transfer of information. However, utilizing a single communication protocol in all scenarios for information transfer may not be energy efficient. In this scenario, switching between different communication technologies based on the application requirements seems to be way in the right direction. Initially, we propose a hybrid communication switching scheme for UAV-assisted wireless networks that can transfer information from UAVs to GCS by utilizing minimum transmission energy. The proposed scheme uses the free space (FS) and free space and multipath (FSMP) formulations to arrive at the optimum switching scheme. To incorporate additional parameters and scale, we further proposed Q-learning and deep Q-network (DQN) algorithms to form the optimum switching scheme.

Based on the above, this thesis is structured into two parts, with the first part discussing the proposed solutions for achieving smart sensing using cutting-edge hardware sensor modules. The second part of the thesis provides insights into transmitting the sensed information and telemetry data in an energy-efficient manner using the proposed hybrid communication switching scheme. Thus, the major contributions of this thesis are outlined as given below:

- A simple cost-effective approach to detect and estimate the UAV count using thermal and acoustic-based sensors. The proposed approach utilizes cutting-edge signal processing and machine learning techniques to accurately determine the exact number of UAVs in the vicinity. The proposed approach is lightweight and low in complexity enabling them to be deployed easily on edge computing devices.

- A radar-based approach to improve the localization of UAVs and target objects in a multi-target environment. The proposed approach improves the angle of arrival estimation of mmWave FMCW radar using a programmable rotor

and state-of-the-art computer vision algorithms. Additionally, the proposed approach classifies the target objects based on the range-angle signatures.

- A novel hybrid communication switching scheme for UAV-assisted wireless network. A framework to switch between different communication technologies based on energy-efficient data transmission is proposed. Two approaches are proposed to aid in the formation of the hybrid network, one using the FS and FSMP path loss models and the other with the help of RL algorithms such as Q-learning and DQN.

1.2 Organization of the thesis

This Ph.D. thesis is a compilation of contents from seven publications, all of which are attached in the Appendix. These publications Paper A, Paper B, Paper C, Paper D, Paper E, Paper F, and Paper G are referenced as [7], [10], [11], [12], [13], [14], and [15] respectively. The thesis is structured into five chapters as follows:

- *Chapter* 1 introduces the reader to the primary content presented in the dissertation. A brief research overview is presented, after which the focus of the thesis is outlined. This is followed by summarizing the primary contributions presented in the dissertation.

- *Chapter* 2 provides the relevant background material required while navigating the subsequent chapters in the thesis. Additionally, a literature survey of the current research associated with each of the issues tackled in the dissertation is also summarized. The contents of this chapter are selectively compiled from all the publications listed in the Appendix section of the thesis.

- *Chapter* 3 proposes various approaches to detect and estimate the presence of UAVs in a multi-UAV system. Various sensors such as thermal, acoustic, and radars are utilized in conjunction with state-of-the-art machine learning techniques to detect and localize UAVs and other object targets. This chapter of the thesis is composed from Paper C, Paper D, and Paper E.

- *Chapter* 4 discusses a novel hybrid BLE/LTE/Wi-Fi/LoRa switching scheme for UAV-assisted wireless networks. The chapter provides a theoretical framework and analytical expressions for energy-efficient data transmission over the hybrid UAV-assisted wireless network. Additionally, two approaches are proposed to aid in the formation of the network, one using the FS and FSMP path loss model, and the other using RL algorithms such as Q-learning and DQN. The chapter is a summarized version of Paper F and Paper G that is provided in the Appendix.

- *Chapter* 5 concludes the dissertation by highlighting the major contributions. Additionally, future avenues for research in conjunction with the thesis topic are also presented and discussed.

Chapter 2

Background Theory and Literature Survey

2.1 Introduction

As mentioned in previous chapters, sensing and communication are vital to maintaining safe and secure operational services for UAVs. This chapter provides relevant background information and literature on the current state-of-the-art concerning the various issues addressed in this thesis. We first present a brief summary of the necessary background theory about different sensors and communication technologies that can be encountered in future chapters. Afterward, a summarized literature review is provided for each of the issues discussed in the succeeding chapters. The various algorithms, machine learning models, and hardware technologies used throughout the thesis are also briefed. This chapter comprises background theory and literature from all the publications, Paper A [7], Paper B [10], Paper C [11], Paper D [12], Paper E [13], Paper F [14], and Paper G [15].

2.2 Background Theory

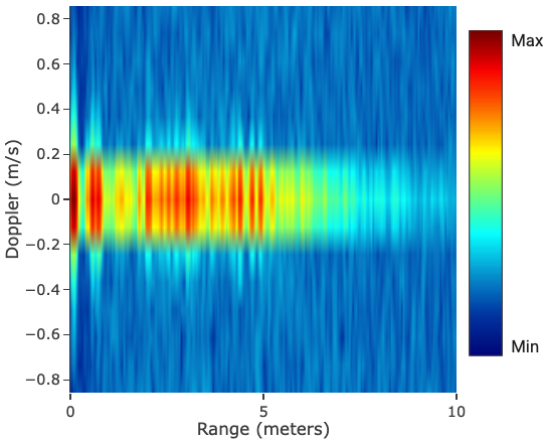
This section provides fundamental information required to access the subsequent chapters in the thesis. First, a summary of different hardware sensors is presented. Afterward, an outline of the various communication technologies utilized in the thesis is presented.

2.2.1 Hardware Sensors

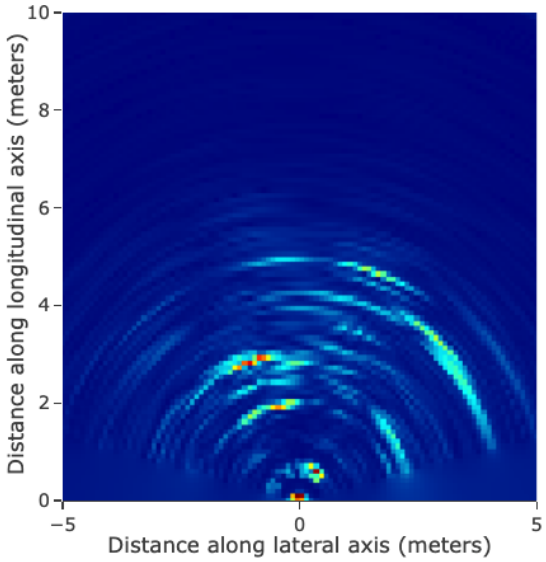
mmWave FMCW Radars

The mmWave FMCW radars [16], [17] are renowned for their ability to deliver precise range and velocity information [18]. mmWave FMCW radars are the natural choice for UAV-related applications because of their robustness against harsh weather and lighting conditions. They are also attractive for use in detection and collision avoidance scenarios due to their high bandwidth and precise range and velocity resolution

(a) Range profile plot



(b) Range-doppler plot



(c) Range-azimuth plot

Figure 2.1: Plots obtained from mmWave FMCW radar [7] as (a) range plot, (b) range-doppler plot, and (c) range-azimuth plot.

[19]. The operation of mmWave FMCW radars is similar to that of conventional radars. The radar emits a frequency-modulated continuous chirp waveform, which is reflected by surrounding objects. The receiver processes the reflected waveform to estimate the radial range, radial velocity, and angle of arrival (AoA) of the target object.

The raw intermediate frequency (IF) samples obtained from the mmWave FMCW radar are processed to create the range-plot, range-doppler, and range-azimuth heatmaps, as illustrated in Fig. 2.1a, 2.1b, and 2.1c respectively. The plots in Fig. 2.1 are obtained when the TI AWR1843 mmWave FMCW radar is utilized in conjunction with the TI mmWave demo visualizer [20]. It can be inferred from Fig. 2.1a that four objects have been detected by the radar at a radial distance of less than 3 meters. The range-doppler heatmap in Fig. 2.1b indicates that these

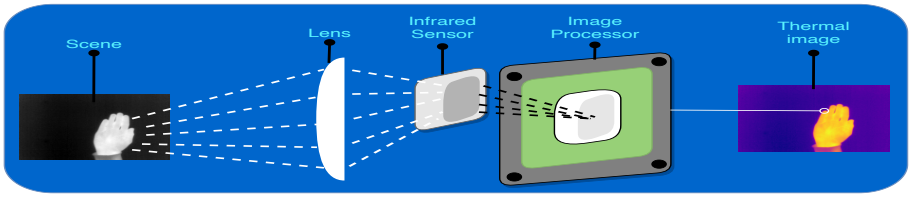


Figure 2.2: Thermal camera operating principle [7].

four objects are stationary as the higher intensity red hue is located in proximity to 0 m/s. Further, it can be deduced from the range-azimuth heatmap in Fig. 2.1c that the objects are positioned at an angle of 10° from the radar.

Because of their superior sensing capabilities, mmWave FMCW radars are employed in UAVs for object recognition, identification, and tracking. In [21], it has been suggested to use the micro-Doppler spectrum derived from mmWave FMCW radars for the detection and identification of birds and UAVs. In a similar vein, the work in [22] uses the mmWave FMCW radar micro-Doppler spectrum to help identify micro-UAVs in low grazing angle settings. This is particularly useful in electronic warfare, where detecting micro-UAVs can be challenging due to the topographical clutter. Furthermore, the authors in [23] have used mmWave FMCW radars in conjunction with convolutional neural networks (CNNs) for activity classification and localization of objects. Similar to this, Siddharth *et al.* in [24] achieved accurate target classification in the range of $87.68\% - 99.7\%$ by combining you look only once (YOLO) and faster region-based convolutional neural networks (RCNN) with range-angle images obtained from mmWave FMCW radars.

Inaccurate AoA estimation is one of the shortcomings of mmWave FMCW radars. The AoA estimation and field of view (FoV) in both azimuth and vertical directions is found to be improved by combining a mechanically rotating radar and novel machine learning algorithms as demonstrated in [25] and [26]. By utilizing a single transmitter and receiver, this approach has the benefit of accurate AoA estimation. Furthermore, in [26], the authors presented techniques for elevation and angle estimation of UAVs from the GCS using mmWave FMCW radars.

Thermal Camera

UAVs equipped with thermal cameras can be useful for surveillance, disaster management, and search and rescue missions. These cameras can operate in harsh weather and low-light conditions. Thermal cameras utilize specialized sensors that can record infrared radiation. After processing the collected infrared radiation, a temperature profile is generated that provides color-specific information on the temperature levels within the captured frame. The generated temperature profile is then utilized to enhance detection and classification capabilities. The operation of thermal cameras is illustrated in Fig. 2.2. Image resolution, range, refresh rate, and lens focal length are a few of the important thermal camera parameters [27], [28].

There are numerous applications for thermal imaging when combined with UAVs. To detect heat leaks in structures, the authors of [29] processed thermal images taken

by a UAV fitted with thermal cameras. Thermal camera-mounted UAVs are used in [30] to automatically monitor and identify wildlife. Moreover, UAVs fitted with thermal cameras are employed for tracking, identifying, and object detection in the ocean [31]. In our work, we detect and estimate the presence of numerous hovering UAVs using the FLIR Lepton 3.5 [32] thermal camera.

2.2.2 Communication Technologies

UAVs must constantly update the GCS about their position, battery health, on-board sensor data, and other critical information to enable and maintain seamless autonomous operations. The transfer of telemetry or sensed data necessitates the use of efficient and reliable communication technology for UAVs. The following paragraphs provide insight into some of the characteristics associated with various communication protocols that are used as a part of this thesis. Table 2.1 highlights the key attributes of these communication protocols.

Long Range (LoRa)

LoRa, developed by Semtech Corporation, is a low-power, long-range communication technology that is predominantly used for internet-of-things (IoT) applications [33]. The MAC layer of LoRa, known as LoRaWAN, is open source and maintained by the LoRa Alliance [34], while the LoRa physical layer is proprietary and utilizes the chirp spread spectrum (CSS) modulation [35]. LoRa has a maximum range of approximately 10 to 15 km and transmits via the unlicensed spectrum at frequencies 433, 868, 915, and 923 MHz. However, one of the primary drawbacks of employing LoRa in UAVs can be attributed to its data rate that can only support up to a maximum of 50 kbps [33].

Bluetooth Low Energy (BLE)

Designed to meet the needs of short-range, low-power applications, Bluetooth Smart, often known as BLE, is an improved version of the original Bluetooth technology [36]. The Bluetooth special interest group (SIG) developed the protocol to meet the demands of low-power applications in healthcare, fitness, and beacons. In addition, BLE is comparable to conventional Bluetooth in that it has a data throughput of 1 Mbps and a range of roughly 50 meters. BLE supports a multitude of topologies, including broadcast, mesh, p2p, and star. BLE utilizes the 2.4 and 2.48 GHz spectrum just like the conventional Bluetooth, however, it is not backward compatible with its predecessor. Due to its high data rate and low power consumption, BLE is an excellent candidate for UAV-based applications [37], [38], [39].

Wireless Fidelity (Wi-Fi)

Widely used in computers, tablets, cellphones, digital televisions, and other devices, Wi-Fi is a short-range communication technology that is based on the IEEE 802.11

Table 2.1: Key metrics for communication technologies [7].

Protocol	Range (km)	Throughput (Mbps)	Power (W)	Frequency (GHz)	Topology
Wi-Fi	≈ 0.1	$\approx 10^4$	≈ 2	2.4, 5	Star, Mesh
BLE	≈ 0.1	0.125 - 1.36	0.01 - 0.5	2.4	Star, Mesh, Broadcast, P2P
LTE-M	≈ 10	≈ 1	0.1 - 0.2	LTE bands	Star
LoRa	≈ 10	0.01 - 0.05	≈ 0.025	0.433, 0.868, 0.915, 0.923	Star

protocol stack. Wi-Fi is available in multiple variants with different power consumption, data rate, and bandwidth specifications. 2.4 GHz and 5 GHz are the unlicensed spectrum bands in which Wi-Fi operates [40]. Orthogonal frequency division multiplexing (OFDM) is the modulation scheme used by Wi-Fi. OFDM equips Wi-Fi with high data rate capability and immunity to interference issues. Due to its modular design, Wi-Fi can be configured to operate in both infrastructure and ad-hoc modes. Due to the above-mentioned features, Wi-Fi has become a promising solution for UAV-based applications. As observed in [41], Wi-Fi is deployed in ad-hoc mode to be used in UAV relay networks for various applications.

Long-Term Evolution for Machine-Type Communication (LTE-M)

LTE-M is developed by the 3rd generation partnership project (3GPP) to facilitate machine-to-machine communications. LTE-M is a low-power wide-area communication standard that provides a high data rate and increased bandwidth to support IoT applications [42]. LTE-M utilizes the licensed spectrum provided by 3GPP. Its unique architectural design has enabled seamless integration with existing cellular networks paving the way for greater adoption in UAV applications. Additionally, the growing popularity of LTE-M can also be attributed to its extended range, low latency, tolerance to interference, and weather conditions.

2.2.3 Edge Computing Platforms

In this section, we provide relevant information on some of the edge computing devices that have been used in our experiments. These are Raspberry Pi 4 Model B, Nvidia Jetson Nano, and Nvidia Jetson Xavier. The proposed machine learning models have been deployed onto these edge computing devices to analyze deployment feasibility and inference time.

In chapter 3, section 3.3, to obtain the inference time on various edge computing devices, the proposed model and benchmark models (refer section 2.4) are first converted to their equivalent TensorFlow Lite versions. TensorFlow Lite [43], developed by TensorFlow, is an open-source software that is used to optimize and deploy machine learning models onto edge computing devices. The TensorFlow Lite model is optimized for model size and inference time and enables seamless deployment on

a variety of embedded devices. In our work, the TensorFlow Lite versions of the proposed model and benchmark models are first deployed onto the above-mentioned embedded devices to determine the inference time.

Raspberry Pi 4 Model B

The latest model in the Raspberry Pi lineup, the Raspberry Pi 4 Model B, has a quad-core ARM Cortex-A72 processor and 2, 4 or 8 GB of RAM. The board is equipped with 2.4/5.0 GHz wireless LAN compatibility, Bluetooth 5.0, BLE, and Gigabit Ethernet. With a 40-pin GPIO header, the board also supports CSI, DSI, USB 2.0, and USB 3.0 interfaces. The Raspberry Pi 4 board is also well-suited for UAV and autonomous applications owing to their compliance-certified Bluetooth and wireless LAN protocols [44].

Nvidia Jetson Nano

Nvidia's Jetson Nano is a compact GPU platform that is designed for edge computing applications. The board has a quad-core ARM Cortex A5 CPU and a 128-core Maxwell architecture GPU [45]. In addition to offering 4 GB of RAM, the Jetson Nano also supports a variety of peripherals such as USB 3.0, USB 2.0 Micro-B, GPIO, I2S, I2C, UART, SPI, and Gigabit Ethernet interfaces. Further, it is also reported that the Jetson Nano can accelerate deep learning frameworks with an AI performance of about 472 GFLOPS [46].

Nvidia Jetson AGX Xavier

The Nvidia Jetson AGX consists of a 512-core Volta architecture GPU and a 64-bit 8-core Carmel ARM architecture CPU [47]. Primarily developed for various autonomous applications, the platform has 64 tensor cores and offers 32 GB RAM. Additionally, the board also comes with a dedicated vision accelerator, deep learning accelerator, and encoder/decoder units for executing various image processing tasks. The Jetson AGX Xavier supports RJ45, USB-C, USB 2.0, UART, and other peripheral interfaces. According to [48] and [46], the Jetson AGX Xavier can accelerate deep learning algorithms by an amount of 32 TFLOPS.

2.3 You Look Only Once (Yolo) Architecture and Working

The Yolov3 is a cutting-edge multi-scale object detection algorithm with a very short inference time [49]. Yolov3 operates by performing a single-stage forward pass over the entire image. The forward pass helps in predicting the class categories and their associated probabilities. Yolov3 is based on the Darknet-53 architecture which combines skip connections and convolution layers [50]. Drawing inspiration from the ResNet architecture [51], the Darknet-53 architecture is mostly composed

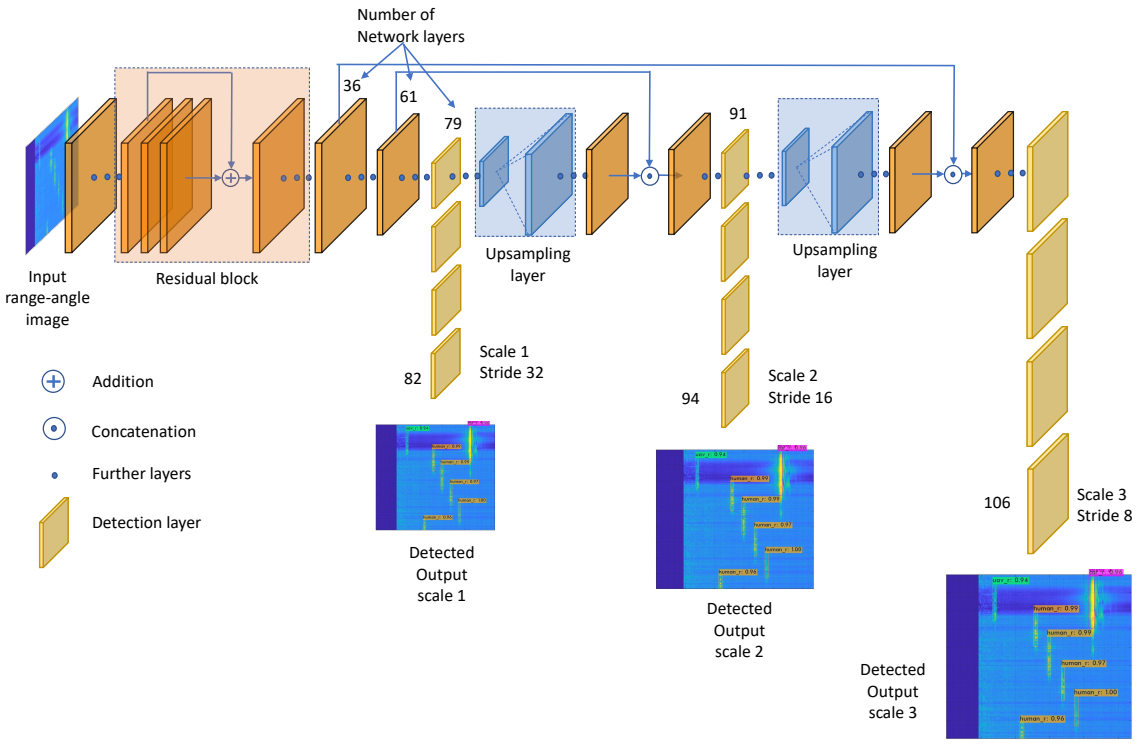


Figure 2.3: The Darknet architecture for Yolov3 [7].

of feature extraction and feature detection stages. In the feature extraction stage, 53 convolution layers are arranged as 3×3 and 1×1 successive layers. This is followed by a skip connection. In contrast, the feature detection stage has 53 layers that contribute to improving the prediction accuracy of the algorithm. As a result, the algorithm utilizes 106 convolutional layers in total for feature identification and feature extraction.

To obtain multi-scale feature embeddings, the Yolov3 algorithm performs an initial pass of the input image through a feature extractor. Yolov3 supports 13×13, 26×26, and 52×52 feature scales. To draw bounding boxes on the identified object, the acquired feature maps are subsequently passed into the feature detection stage. Here, every feature map is separated into a group of grid cells and each grid cell is responsible for predicting almost three bounding boxes. Each bounding box consists of class probabilities and class attributes. The likelihood that a specific class occurs in the corresponding grid cell is given by the class probabilities. Each bounding box 'B' containing '$5+C$' attributes has 'C' number of classes. In our work, the proposed model is being used to identify three classes, namely UAV, car, and humans, and so $C = 3$. The number '5' in '$5 + C$' denotes the bounding box attributes (t_x, t_y, t_h, t_w) and the objectness score. Yolov3 computes bounding box dimensions in relation to anchor box dimensions where the anchor boxes are prior boxes with fixed aspect ratios. These fixed aspect ratios of anchor boxes are computed by first executing a k-means algorithm on the complete dataset before training. Finally, the objectness score is a metric that determines the presence of an object in the relevant grid cell. Convolution of the downsampled feature maps with a 1×1 detection kernel is

performed for feature detection. Here, the dimension of the detection kernel is given by $1 \times 1 \times (B * (5 + C))$. Yolov3 has three layers at which feature detection occurs, namely the 82^{th}, 94^{th}, and 106^{th} layers.

Following object detection, each grid can contain up to three bounding boxes, depending on the object's relative location within the grid cell. A non-maximum suppression is performed to prevent multiple bounding boxes for the same detected object. The bounding box with the largest area of overlap with the detected object is retained, while the remaining bounding boxes are discarded. In our work, we employ the Yolov3 version to estimate the AoA from range-angle images. In Fig. 2.3, a high-level architecture of the Yolov3 version is provided. Further, Fig. 2.3 also illustrates how Yolov3 generates three distinct resolution outputs for a given input range-angle image.

2.4 Benchmark Models

Benchmark models are advanced machine learning models with unique architectural characteristics. For example, DenseNets [52] preserves the feed-forward nature of the network by feeding the feature maps from each layer to all subsequent levels. To enhance performance, residual blocks are introduced by the ResNet architecture [53]. The skip connections that make up the residual blocks retain the lost abstractions in the original path thereby improving the performance. Further, to improve accuracy, the EfficientNet family of CNNs [54] utilizes uniform scaling of the network's width, depth, and resolution. On the other hand, MobileNet [55] architectures are designed to offer fast computations on embedded and mobile devices. These models with their low memory footprint offer a better trade-off between accuracy and resource consumption. In chapter 3, section 3.3.7, we assess the performance of the proposed model by comparing it with these existing benchmark models. DenseNet [52], EfficientNet [54], Inception [56], MobileNet [55], ResNet [53], NASNetMobile [57], VGG [58], and Xception [59] are among the 23 benchmark models that have been used. The benchmark models accept an input image resolution of $224 \times 224 \times 3$ and are pre-trained using the ImageNet dataset [60].

2.5 Continuous Wavelet Transform (CWT)

The CWT splits a signal into its time and frequency counterparts [61]. Similar to the short-time fourier transform (STFT) [62], the CWT operates by measuring the correlation between the analyzing wavelet ψ and the original signal $f(t)$. The analyzing wavelet is scaled and dilated according to the parameters p and q respectively. To calculate the CWT of a signal $f(t)$, assuming $p > 0$, then,

$$C(p, q; f(t), \psi(t)) = \int_{-\infty}^{\infty} f(t) \frac{1}{p} \psi^* \left(\frac{t - q}{p} \right) dt, \tag{2.1}$$

where the * operator represents the complex conjugate [63]. The output obtained from CWT depends on the nature of the original signal. The output is real-valued if

the input signal on which CWT is applied is real. The $C(p, q)$ coefficients obtained after continuously varying p and q are used to plot the spectrogram of the signal. In chapter 3, section 3.3, CWT is applied over the trimmed audio clips to generate the audio spectrograms. The audio clips are sampled at 48000 Hz for a 3 second duration. The resulting spectrogram displays time and frequency components associated with the acoustic signature of UAVs.

2.6 Literature Survey

This section predominantly focuses on the current state-of-the-art prevalent for the various issues outlined in this thesis.

2.6.1 UAV Count Estimation Using Thermal Imaging and Lightweight CNN

As evidenced by the literature, there has been prior research on the detection and estimation of UAVs using thermal imaging. A noteworthy example of this can be seen in [64], where the authors use deep learning techniques to create a UAV tracking and detection system. Adversarial data augmentation methods were used to develop a dataset with UAVs operating in both thermal and visual modes. Despite being trained on artificial data, the proposed model demonstrated good performance on real-world UAV images with complex environments. In [65] and [66], UAV detection systems using thermal sensors are developed to detect obstacles at night. Specifically, the proposed approach in [65] utilizes an RGB camera and ADS-B signals to validate the results. Additionally, the reported detection accuracy was 100% for all nighttime cases and extreme illumination scenarios. In [67], an alternative method for developing a UAV detection system made use of faster-RCNN, a saliency map, and an MSO module. Meaningful features from the thermal images were extracted with the use of the thermal saliency map. Before supplying the thermal image to the model, the MSO module improves the resolution of small objects, which improves the model accuracy. According to reported results, the proposed approach achieved an accuracy of more than 93%. Using an alternate strategy, the authors of [68] examine the performance of a UAV detection system by training with a limited number of thermal images. To improve model accuracy, the features visible in thermal images are preserved in their preprocessed RGB image counterpart. Afterward, the previously processed RGB images and thermal images are then used to train the model. According to the reported results, it was inferred that the type of preprocessing performed on the RGB images significantly impacted the model accuracy.

As evident from previous research, the majority of the literature on thermal imagery-based UAV detection lacks real-world datasets. To imitate the features of thermal images, RGB images or other sources are utilized to create synthetic thermal images. Moreover, the majority of the literature concentrates on the detection of a

single UAV. Simultaneous detection of several UAVs is essential to assure safety and security in real-world trespassing and intrusion scenarios. Section 3.2 of chapter 3 investigates this problem and attempts to provide a potential solution. Our proposed approach to precisely calculate the number of UAVs in a multi-UAV scenario seeks to close this gap. Furthermore, this work will serve as a guide for future UAV detection systems that embark on concurrently detecting several UAVs.

2.6.2 Acoustic and Machine Learning Based UAV Count Estimation

Extensive works have been published on the topic of UAV detection using acoustic signatures. In this section, we provide a brief overview of the current state-of-the-art on acoustic-based UAV detection. However, for a detailed account of the topic, the interested reader is encouraged to refer to section D.2 of this thesis. Previous approaches to detect UAVs based on acoustic signatures can be classified into two, conventional methods and machine learning-based methods.

Conventional methods

The conventional method of utilizing acoustic data to detect UAVs is based on signal processing techniques such as correlation. One such study in [69] utilizes a modified cross-correlation approach for UAV detection. The differential Doppler shift produced by the rapid motion of the UAV and the spacing of the microphone array is utilized by the proposed approach to detect UAVs. The incoming signals are successfully decorrelated from the background noise using the differential doppler shift feature. The measurements were performed in a quiet, controlled environment. The collected data was then combined with ambient noise to assess the performance of the proposed approach. In another study [70], a drone acoustic detection system (DADS) employing microphone nodes was developed for UAV detection and classification. The steered-response phase transform (SRP-PHAT) approach is used in the DADS to detect UAVs. Additionally, the propeller frequencies from the spectrogram of the measured acoustic signatures are used to classify the drones. Based on the reported results, the SRP-PHAT approach works well with real UAVs in practical situations. However, based on the classification algorithm, the UAVs must reach a threshold distance to attain improved performance. In [71], the acoustic spectrum produced by UAVs is studied and analyzed using the Barlett, capon, and cross-correlation approach. To achieve performance comparable to the mel-frequency cepstral coefficients (MFCC) method, an additional high-pass filter is employed in the proposed methodology. Subsequent experiment results also show that the cross-correlation approach performed better when noise was removed using low-pass filtering. In contrast, [72] makes use of the intrinsic harmonics present in the acoustic field signatures for 3D localization and detection. Zero-phase bandpass filters in conjunction with a pitch recognition algorithm are used to extract the fundamental and other significant harmonics of the acoustic input signal. To assess

the performance, experiments have been performed outdoors and in anechoic conditions. It was found that the proposed method performs admirably well when the UAV follows plain vertical trajectories. However, the performance was inadequate for complicated trajectories involving several UAVs and calls for additional research. Another study that sheds light on using acoustic signatures to detect UAVs is [73]. In this study, scientists use a sound level meter to gauge the noise level in a confined space with just a single UAV. Following the acquisition of the noise-free measurements, UAV noise was combined with ambient music and anthropogenic noise from humans in the surrounding area. The frequency spectrum was analyzed using spectral methods, and it was revealed that the detection of UAVs was confirmed upon noticing a peak at $5,000$ Hz in the frequency spectrum. Nevertheless, the study is unable to replicate and validate the findings in a multi-UAV real-world setting. The authors of [74] used a biologically inspired vision approach to solve the UAV detection challenge. Time-frequency analysis of the audio signals is performed to obtain spectrogram images. Using a hoverfly vision model to preprocess these spectrogram images, meaningful representations can be extracted that can aid in UAV detection. In comparison to conventional narrowband and broadband methodologies, it was demonstrated through outdoor field trials that the bioinspired technique can increase the maximum UAV detectable distance between 30% and 50%. Nevertheless, further validation of the proposed method is required through additional UAV trials and flight scenarios.

Machine learning methods

New capabilities for UAV detection and classification have been made possible by the development of machine learning algorithms [75], [76]. To aid in UAV detection and classification, machine learning approaches extract the data's innate hidden patterns [77]. A notable increase in detection and classification accuracy has been reported by combining machine learning techniques with other preprocessing methods such as principal component analysis (PCA) and STFT [78]. In the following paragraphs, we examine some of the literature on machine-learning techniques for UAV detection and classification that utilize acoustic signatures.

In [79], the authors examine the acoustic fields produced by several miniature quadcopter UAVs. The acoustic field data is then provided to COMSOL Multiphysics simulation software to perform numerical analysis and simulations. The simulation results ascertained the impact of directional patterns, blade defects, and propeller-induced pressure changes on the generated UAV acoustic fields. Further, to detect UAVs, the acquired acoustic fingerprints were then fed into a neural network model that was trained using the cepstrum coefficients. In another work [80], the authors detected and tracked a UAV in a noisy real-world environment using several microphone nodes. MFCC and STFT were employed in the data preprocessing stage. CNN and support vector machines (SVM) were then used for training. Results indicate that when a UAV is in close proximity to a microphone node, the STFT-SVM model performs better at detecting it. Subsequent research endeavors

may involve testing various UAVs and mitigating ambient noise while preprocessing. A multi-class SVM is used in [81] to recognize UAVs under various environmental situations. As part of the work, a dataset comprising five 70 minute audio files is recorded. The recordings included audio signals captured during the daytime from nature, street traffic noise, train audio, audio from crowd, and flying UAVs. To perform further analysis, the audio recordings are then divided into parts of 5 and 20 milliseconds. Next, preprocessing is performed to extract features such as MFCCs, zero crossing rate (ZCR), spectral centroid, spectral roll-off, and temporal centroid. Afterward, the SVM classifier uses these extracted features to output a UAV detection accuracy as high as 96.4%. The authors of [82] use k-nearest neighbors (k-NN) and plotted image learning (PIL) as two classifiers to detect UAVs. Audio clips of 1 second duration are recorded independently in a noise-free indoor and outdoor setting using DJI Phantom 1 and DJI Phantom 2. Subsequently, the audio from the outdoor environment is included to replicate real-world situations. The audio snippets then undergo an FFT transformation before being sent to various classifiers. According to the authors, PIL demonstrated 83% accuracy in detecting UAVs, while k-NN only accounted for 61%. In [83], the authors created a distributed system for UAV detection and localization utilizing an acoustic wireless sensor network. Trial investigations revealed that there was a considerable difference between the background spectrum and the power spectral density (PSD) of UAV sound. A UAV audio dataset was developed based on this idea. The dataset included ambient background noises added to UAV audio. Upon low pass filtering of the audio samples at 15 kHz, FFT is then applied to generate the PSD of the signals. Additional PCA transformation is performed to reduce the dimensions. The preprocessed signals are then sent to the SVM classifier after being split up for training, testing, and additional testing using overlapped signals. Reported results indicate that successful UAV identification occurred when the signal-to-interference ratio (SIR) was above 10 dB. In the work [84], the blind source separation (BSS) method is used to detect UAVs in the presence of multiple source interference. The audio signatures are recorded using three distinct UAVs, each operating independently. The proposed approach begins by estimating the total number of sources. Depending upon the type of source separation required (overdetermined, positive-definite, or underdetermined), either ICA, PCA, or variational mode decomposition (VMD) is applied after source estimation. To assess the performance, the extracted features are then fed to various machine learning algorithms, including SVM, k-NN, and decision trees. Reported results indicate that SVM and k-NN performed similarly, with SVM performing marginally better. Both algorithms outperformed conventional filtering and mixed-signal techniques, displaying an accuracy of more than 90% for UAV detection. An alternate approach for detecting UAVs using acoustics was carried out in [78]. Firstly, the sounds of thunderstorms, birds, airplanes, and amateur UAVs were recorded in a noisy setting. Next, for feature extraction, the authors employed linear predictive cepstral coefficients (LPCC) and MFCC. To recognize and identify UAV acoustics, the collected features are then fed as input to SVMs with linear, cubic, and quadratic kernels. The findings indicate that SVM using a cubic kernel

in conjunction with MFCC features outperforms LPCC, achieving a UAV detection accuracy of approximately 96.7%.

Supervised learning [85] uses labeled datasets as inputs and outputs, in contrast to the previously stated unsupervised approaches. The labeled data helps the model to understand the structure of the data by taking on a supervisory role. Based on the amount of labeled data and its underlying structure, supervised algorithms learn to become increasingly accurate over time. The authors of [86] conducted a comparison analysis to identify the optimal classifier for acoustic UAV identification. To replicate real-world UAV scenarios, acoustic signatures from various UAVs are recorded and then these audio signals are supplemented with various distinct environmental noises. Next, the MFCCs features are extracted from these acoustic signatures and then provided to various classifiers to evaluate their effectiveness. With an F-score of 80%, the recurrent neural network (RNN) was found to perform the best, followed by the gaussian mixture model (GMM) at 68% and CNN at 58%. In [87], normalized STFT is applied to UAV acoustic signals. The authors used the DJI Phantom 3 or 4 versions to record the UAV acoustic signatures. Once recorded, the captured audio snippets are trimmed to 20 ms duration with a 50% overlap. The trimmed audio clips are then subjected to normalized STFT to produce 41958 non-UAV and 68931 UAV audio frames. There were acoustic signatures from motorbikes and scooters in the non-UAV audio frames. Additive white gaussian noise (AWGN) is added to the output of the STFT before it is fed into the proposed CNN architecture. The findings showed a false alarm rate (FAR) of 1.28 and an accuracy of 98.97% for UAV detection. In this study [88], remotely piloted aircraft systems (RPAS) in urban contexts are detected using CNN and sensory substitution of ambient and pre-existing microphones. A diverse set of RPASs were used to conduct separate indoor and outdoor trials. The captured audio snippets are then used to create spectrogram images. Next, these spectrogram images are fed to the Inception CNN model for further training via transfer learning. For every RPAS class, the results revealed an RPAS detection accuracy of more than 90%. Mel-spectrograms are used in [89] to extract features from UAV audio signals. In this work, with the goal of UAV classification, the authors employed CNNs and convolutional recurrent neural networks (CRNN) to extract the relevant features. From the obtained mel-spectrograms, it was determined that CNNs performed better than other methods for UAV classification. The study also investigated ways to enhance UAV detection performance by combining ensemble techniques with late fusion methods. UAV detection in [90] also uses audio spectrograms in conjunction with CNN, RNN, and CRNN. In a controlled setting, the authors used two distinct UAVs to perform experimental studies. Realistic audio data that can be used for inference was obtained by augmenting real-world background noises. According to the published results, CNN and CRNN exhibited accurate detection and classification of UAVs outperforming RNN models. Our proposed method referenced in section 3.3.4 of chapter 3 also revolves around a similar approach in which we utilize audio spectrograms to perform multiple UAV detection.

The most recent studies on machine learning-based acoustic UAV detection are

Table 2.2: Overview of recent research on machine learning-based acoustic detection of UAVs [12].

Reference	Method	Results	Limitations
[81]	Preprocessing using ZCR, MFCC, spectral centroid, etc. Extracted features are fed to multi-class SVM.	UAV detection accuracy - 96%	• Single UAV case. • Lacks real-world experiments and background noise.
[86]	MFCC coefficients are fed to RNN, GMM, and CNN.	RNN showed best F-score with 80% followed by GMM with 68% and CNN with 58%.	• Augmented environmental noise. • Multi-UAV detection is absent.
[83]	PSD using FFT followed by PCA for dimension reduction. Output fed to SVM.	Best accuracy when SIR was greater than 10 dB.	• SVM is more sensitive to bit error rate.
[87]	Normalized STFT features with CNN.	UAV detection accuracy of 98.97%	• Considered only single UAV scenario. • AWGN is added to simulate a noisy environment.
[82]	Preprocesing using FFT. Trained using PIL and KNN.	PIL - 83% and KNN - 61% accuracy	• Sound clips from 2 UAVs recorded separately.
[80]	Preprocessing with MFCC and STFT. Obtained features fed to SVM and CNN.	eSTFT-SVM reported best performance.	• Considered only single UAV case. • Model accuracy is low.
[88]	Audio spectrograms with CNN. Used different RPAS classes individually for measurements.	Greater than 90% detection accuracy.	• Multiple RPAS scenario is absent.
[90]	Audio spectrograms with CNN, RNN, CRNN.	CNN reported best detection accuracy with 96.38% followed by CRNN with 94.72%. Experimented with two different types of UAVs.	• Lacks real-world experiment scenarios. • Doesn't estimate the number of UAVs.
[78]	MFCC and LPCC for feature extraction. Features are fed to SVM with linear, cubic, and quadratic kernels.	MFCC with SVM cubic kernel achieves 96.7% detection accuracy.	• Considers only single UAV scenario.
[89]	Mel-spectrograms for feature extraction followed by CNN and CRNNs.	CNN (94.7% accuracy) outperformed CRNN (94.1% accuracy). Experimented with real-world scenarios.	• Multiple UAV scenarios are absent.
[79]	Trains neural network on cepstrum coefficients.	Relatively high UAV detection rate.	• Multiple UAV scenarios are absent.
[84]	BSS using ICA, PCA, or VMD features. Obtained features are fed to SVM, KNN, and Decision trees	SVM and KNN reported more than 90% accuracy	• Lacks real-world scenarios with background noise.

compiled in Table 2.2. As observed from Table 2.2, a major portion of the existing literature investigates the detection of a single UAV [80], [81], [86], [87], [88], [78], [89]. In scenarios involving several UAVs, the results obtained for detecting a single

UAV can differ significantly. Comparably, the scenarios taken into consideration in the literature roughly duplicate well-defined and controlled UAV trajectories [81], [90], [86], [87], [84]. Thus, such controlled scenarios may not provide a practical UAV flight trajectory and hence can adversely affect real-time detection accuracy. Moreover, several discussed techniques call for the utilization of extremely sophisticated hardware infrastructure, which may not be always accessible [83].

2.6.3 Angle of Arrival Estimation and Classification of Targets Using Rotating mmWave FMCW Radar and Yolov3

As mentioned in section 2.2.1, mmWave FMCW radars are low-cost sensors that can provide accurate range and velocity estimates. However, their angle of arrival estimates are poor. This limitation prevents the radar from being used in critical detection and localization operations. As such there have been several attempts in the literature to improve the AoA estimation. In [91], an adaptive radar signal processor is proposed by the authors for the detection of multiple UAVs. The received signal samples are first subjected to pulse compression (PC), fast fourier transform (FFT), and beamforming by the radar signal processor. Afterward, for each range-Doppler angle bin the processed signal is assessed for potential detection using the constant false alarm rate (CFAR) logic. The angular characteristics of the targets are collected and both single and multi-target scenarios are taken into consideration. Based on simulation and real-world test results, it was found that the proposed method can detect and resolve two UAVs situated $\approx 8^{\circ}$ apart with satisfactory performance. Aubry *et al.* attempts to solve the adaptive radar detection problem in the presence of limited training data in [92]. Three detection schemes based on the generalized likelihood ratio test (GLRT) are proposed. The detection schemes are based on the assumption of linear combination for inverse covariance matrices. Based on the published results, even with a limited number of training samples, the proposed GLRT-1 and GLRT-3 outperform traditional adaptive detectors by a significant performance margin. A novel approach for DoA estimation based on 2-D spectrum sensing is developed in [93]. To gain 2-D occupancy awareness, the proposed method takes advantage of the inherent block-sparsity of the 2-D space-frequency profile. By introducing a variable for block-sparsity, the block sparse learning via iterative minimization (BSLIM) approach is used to solve a non-convex regularized maximum likelihood (RML) estimation problem to obtain the space-frequency profile. It was observed from the reported results that the BSLIM algorithm outperforms conventional methods in terms of false alarms and detection rate. The authors of [94] introduce a novel class of covariance matrix estimators called the median matrices, that are independent of the sample probability distribution. The construction of the median matrices involves utilizing the positive definite properties of matrices. Depending upon the output obtained from the new estimators, the secondary outliers in the training data are eliminated using a generalized inner product (GIP) selection criterion. It was observed that the log-

euclidean median-based estimator performs better than other estimators in terms of the training data's secondary outlier selection probability. The multi-class inverse wishart mixture (MC-IWM) filter [95] and the fast maximum likelihood (FML) covariance estimator [96] are two additional covariance estimators that can be used to infer radar disturbances and support precise UAV detection and localization.

To precisely estimate the AoA, the authors in [97] use a rapid iterative adaptive algorithm that makes use of a time-shared technique. Using a single mmWave FMCW radar, the technique's initial step requires estimating the coarse azimuth angle. To precisely estimate the DoA within a limited area, the intermediate results are then fed into a more sophisticated iterative algorithm. However, this approach is computationally demanding because it depends on several mmWave FMCW radars. An alternate approach in [98] simultaneously calibrates the antennas and estimates the AoA using linear algebraic techniques. Under the presence of an effective coupling between the antennae elements, the iterative optimization algorithm showcases excellent performance. Furthermore, it is demonstrated that the proposed method operates effectively with several sensors and transmitter-receiver pairs. In [99], Daegun *et al.* presented a joint angle and delay estimation technique that extends the one-dimensional pseudospectrum searching by making use of the dual-shift-invariant nature of the received signal. Monte Carlo simulations were employed to evaluate the performance of the proposed approach. Results indicate that the proposed approach performed better than the state-of-the-art. However, the proposed approach is computationally demanding and requires additional testing with challenging real-world scenarios. In [100], the authors develop a low-complexity estimating technique that approximates different FMCW radar characteristics, including the AoA. The proposed approach relies on combining the benefits of the MUSIC and 2D-FFT algorithms. However, under the presence of several targets, the proposed approach fails to yield any significant performance gain. The notion of rotating the radar about its axis to scan an area is used in [24] and [25]. The range profile is constructed using the signals that are acquired in each frame of the rotating radar. The range-angle maps, which provide a 180° field-of-view of the scene, are then constructed by stitching and combining all of the individual range profiles. Nevertheless, [24] is primarily focused on target detection and classification and is not intended for AoA estimation. In contrast, Linga *et al.* have presented a novel AoA estimation technique in [25], that performs reasonably well for AoA estimation. Our work will be an extension to this existing work, in which the proposed method estimates the AoA from the range-angle maps by utilizing cutting-edge machine learning techniques.

2.6.4 Hybrid Communication Switching Scheme for UAV-Assisted Wireless Networks

To ensure reliable and robust information transfer, UAVs are outfitted with various communication technologies. BLE [101], Wi-Fi [102], LTE-M [103], and LoRa [35] are some of these communication modules. Every communication technology is different in terms of transmit power, bandwidth, range, and other attributes. To fulfill

various application requirements, the use of just one communication technology may not suffice and calls for a hybrid and adaptive communication switching scheme for UAV-assisted wireless network. Thus, in this thesis, we present the formation of a hybrid communication switching scheme for UAV-assisted wireless networks.

The amount of literature prevalent on the topic of hybrid switching schemes for communication systems is currently scarce and sparse. To increase the sum rate capacity, the authors of [104] have proposed an indoor hybrid RF/PLC/VLC communication system that switches the device connections between the RF, PLC, and VLC. Moreover, [105] has formulated and studied the transmit power minimization problem. In [106], a dual radio IoT architecture has been proposed for the use of wildlife monitoring systems. Based on the herd's closeness, the proposed approach utilizes BLE in low power wide area networks (LPWAN). To assess the effectiveness of the proposed approach, an analytical model for energy consumption has also been provided. Research has demonstrated that the proposed dual radio network outperforms a network that just uses LPWAN in terms of energy efficiency. The authors of [107] have examined a multi-layer ad-hoc UAV network that utilizes BLE and Wi-Fi to enhance the network's throughput and latency. The UAV-assisted network is first split up into several clusters. BLE is considered to be the primary choice when communication takes place within clusters and also between gateway UAVs and cluster heads. For the communication between gateway UAV and GCS, Wi-Fi is the preferred option. The performance of the proposed scheme against standalone communication protocols is assessed through simulations using the optimized network engineering tool (OPNET). It has been determined that in terms of throughput and latency, the proposed scheme outperforms standalone communication protocols. Nevertheless, the proposed scheme has not given enough attention to the UAV energy consumption. Furthermore, this scheme has not considered additional communication protocols that can support long-range communication between UAVs and GCS. Moreover, the proposed scheme focuses only on the free space path loss model which may be inadequate in real-world situations where multipath propagation dominates. Currently, to the best of our knowledge, no work is available in the literature that focuses on the topic of hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network for energy-efficient data transmission.

The use of reinforcement learning (RL) in UAV-assisted wireless networks has been the subject of numerous works published in the literature. In [108], a deep learning (DL) technique has been presented for collecting data in a wireless network aided by UAVs. A number of DL-based artificial intelligence (AI) techniques, including federated DL, multi-agent deep deterministic policy gradient, and point learning, have been proposed in [109] to address the problem of optimizing energy efficiency in UAV-assisted wireless network. For UAV-assisted wireless networks operating in the TeraHertz band, the joint flight control and spectrum access problem is solved in [110] using a combination of RL and echo state learning. In [111], the authors have proposed a deep reinforcement learning (DRL) method for allocating resources in UAV-assisted wireless networks in terms of throughput, bandwidth, and power consumption. In [112], the authors jointly optimize channel and relay selec-

tion in UAV-assisted device-to-device (D2D) networks using the stochastic learning automata (SLA) algorithm. However, none of the above-mentioned works focuses on leveraging RL algorithms to form a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network that supports energy-efficient data transmission. Our work referenced in section 4.8 of chapter 4 of this thesis focuses on this aspect.

Chapter 3

Smart Sensing for UAV Networks

3.1 Introduction

In this chapter, we discuss some of the inherent issues in the detection of multiple UAVs in a multi-UAV system. We primarily focus on achieving practical solutions to the problem using state-of-the-art low-cost sensors. The proposed approaches in each case are intended to be lightweight and computationally less demanding. This is to ensure portability and deployability on edge computing devices.

This chapter is a summarized version of the contents found in Paper C [11], Paper D [12], and Paper E [13]. The chapter starts by first introducing an approach to estimate the number of UAVs in the vicinity using thermal imaging. Next, we employ acoustic sensors and lightweight CNNs to detect and estimate the UAV count. In the third section, we look at how mmWave FMCW radar sensors can be used to localize and classify multiple targets by utilizing machine learning techniques. Finally, the last section provides an overview of the work and future research directions.

3.2 UAV Count Estimation Using Thermal Imaging and Lightweight CNN

3.2.1 Motivation

Recent technology advancements have generated interest in using UAVs for a range of applications, including disaster management, package delivery, law enforcement, surveying, and defense [7]. Nevertheless, the improper and illicit usage of UAVs in recent times has increased the need to detect, monitor, and track UAV-related activities [113]. In this regard, techniques to identify and detect unauthorized and intruding UAVs are crucial. In our work, we tackle this problem by employing a thermal imaging-based technique. The principal incentive for employing a thermal imaging-based solution is their enhanced visibility in intensely bright and dim environments. Moreover, unlike acoustic sensors, thermal sensors are not affected by background noise [114], [10].

3.2.2 Contributions

The major contributions of this work as provided in Paper C [11] are as follows:

- Proposed a simple cost-effective approach to estimate the count of UAVs in multi-UAV setting by utilizing thermal imaging.
- Developed a lightweight machine learning model that can be used on the edge to estimate the UAV count in a multi-UAV scenario.
- Developed a thermal dataset that contains a total of 10 UAVs flying simultaneously in all directions in a random manner.

3.2.3 System Model

Batteries are the primary power sources that enable UAVs to perform various tasks including flight, landing, and hovering. During these tasks, UAVs dissipate heat energy. Propeller rotation and other aerodynamic motion also results in energy dissipation for UAVs. Some part of the energy thus dissipated from the above-mentioned dynamic motions is converted to thermal energy. This thermal energy is characteristic to each UAV and depends on factors such as UAV form factor, UAV internals, and UAV battery source. Hence, by carefully processing the thermal characteristics of UAVs, these thermal signatures can be leveraged to detect UAVs. Therefore, the main goal of this work is to leverage the unique thermal signatures of UAVs along with state-of-the-art machine learning techniques so as to estimate the total UAV count.

3.2.4 Methodology

The proposed method makes use of the distinctive thermal signatures produced by UAVs carrying out diverse tasks. In this case, we use the FLIR Lepton thermal camera [32] to capture the thermal signatures. The recorded thermal images are then fed to a customized lightweight CNN model to determine the number of UAVs in the captured frame. The steps are outlined as follows:

Step 1: Use the FLIR Lepton thermal camera to capture thermal images of UAVs.
Step 2: Resize and normalize the captured thermal images before providing them as input to the proposed CNN architecture.
Step 3: Utilizing the processed images, train the proposed lightweight CNN model to determine the approximate number of UAVs present in the scene.

3.2.5 Measurement Setup

The experiment was carried out primarily in an enclosed laboratory setting with a radius of 5 meters that was semicircular in shape. The thermal images of UAVs

Table 3.1: Various parameters used for the experiment [11].

Parameters	**Details**
Measurement duration	5 minutes
Semicircular area	Radius: 5 meters
UAV model	**Count**
DJI Mavic 2 Enterprise	1
DJI Mini SE	2
DJI Mini 2	1
SYMA X30	1
DJI Mini 3 Pro	1
Tello EDU	4

Table 3.2: Specifications of thermal camera [11].

Parameters	**Value**
Size (mm)	$10.5 \times 12.7 \times 7.14$
Resolution (pixels)	160×120
Frame Rate (Hz)	8.7 (effective)
Lens Type	f/1.1
Thermal sensitivity (mK)	< 50
Spectral Range (microns)	$8 - 14$ (nominal)
Horizontal Field of View (°)	57

were captured in each measurement of the experiment. Each measurement was taken with 1, 2, 3, $\cdots$ upto 10 UAVs randomly maneuvered in all directions. For the measurements, a variety of UAV models were used, including the Tello EDU series [115], the DJI series [116], and the SYMA series [117]. Table 3.1 contains information on the different UAV model types used in this thesis. Tello EDU UAVs were also utilized in the planned experiments as it can be challenging to detect small-sized UAVs. Each Tello EDU was configured to follow predetermined flight paths. On the other hand, the remaining UAVs were manually controlled by humans. More information with respect to the experiment setup can be obtained from Table 3.1.

In this work, the FLIR Lepton 3.5 [32] camera is used to capture the thermal images of UAVs. With an effective frame rate of 8.7 Hz, the FLIR Lepton 3.5 produces a thermal image with a resolution of 160×120 pixels. With a nominal spectral range of 8–14 microns, the FLIR Lepton provides a thermal sensitivity of less than 50 mK. We use the FLIR Lepton due its excellent performance and compact form factor. Table 3.2 provides additional information about the FLIR Lepton thermal camera.

A static tripod is used to mount the thermal camera, as illustrated in Fig. 3.1. A USB A connection from the Raspberry Pi Model 4B provides power to the thermal camera [44]. Furthermore, a Picamera [118] is installed directly beneath the FLIR

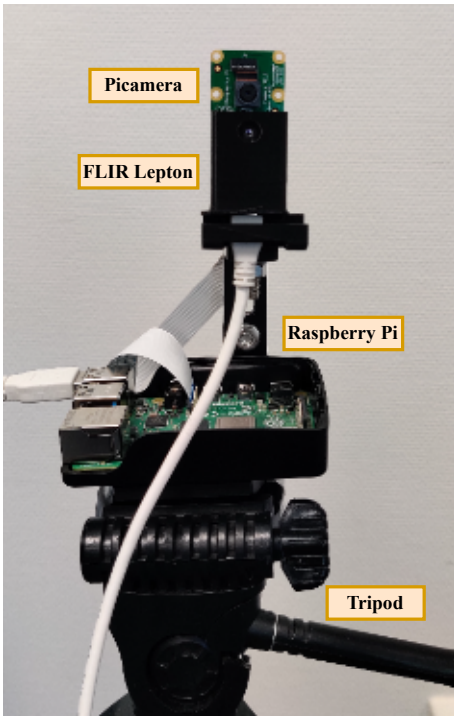


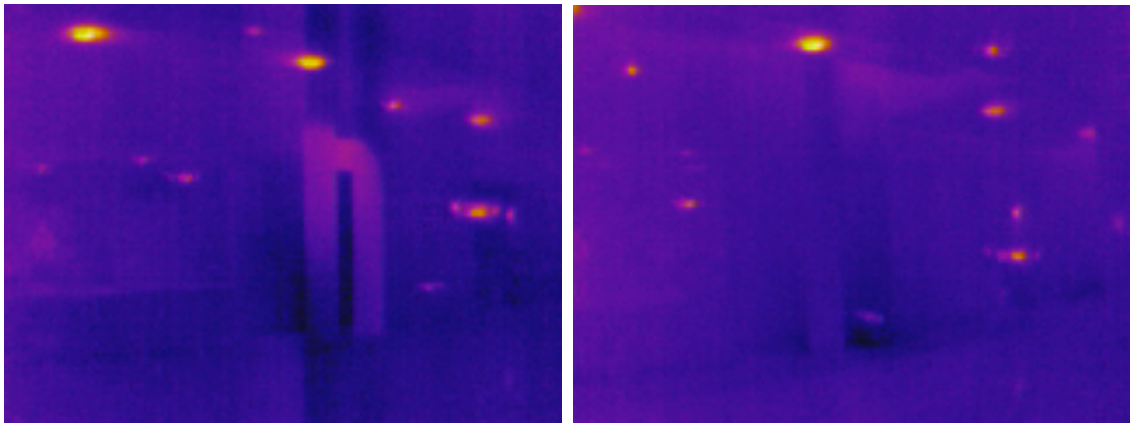
Figure 3.1: Experimental setup used for collecting thermal images of UAVs.

Lepton. The Picamera is used to capture RGB pictures that will act as ground truth for the thermal images. The Raspberry Pi powers the Picamera as well, and both devices work in tandem with the thermal camera. Apart from acting as power sources to the two cameras, the Raspberry Pi is also responsible for capturing and archiving the RGB and thermal images that are collected from the FLIR Lepton and Picamera, respectively. The operating system that is installed on the Raspberry Pi is Ubuntu 18.04 [119]. Ubuntu 18.04 was chosen, so as to ensure smooth installation and integration of ROS Melodic [120] and other necessary software packages onto the Raspberry Pi board. Data from both the FLIR Lepton and Picamera can be simultaneously captured with the aid of ROS Melodic. The Raspberry Pi board is powered by a portable battery bank.

Each measurement involved flying a predetermined number of UAVs in every direction. For every one of these flight paths, thermal and RGB images are captured and saved. Ten measurements are taken, with each measurement consisting of 1, 2, 3, and so on up to 10 UAVs flown in arbitrarily different directions. Each measurement was designed for 5 minutes so that at least 2500 thermal images were captured. To increase variability, several UAV models were employed in the planned measurements. In order to introduce additional unpredictability and noise to the data set, one of the measurements was performed outdoors. This in turn offers a means of comprehending the performance of the proposed approach in outside environments.

(a) (b)

Figure 3.2: The RGB images obtained by using Picamera with the configured experimental setup for measurement scenarios [11] with (a) 6 UAVs and (b) 8 UAVs.



(a) (b)

Figure 3.3: The thermal images obtained by using FLIR Lepton with the configured experimental setup for measurement scenarios [11] with (a) 6 UAVs and (b) 8 UAVs.

3.2.6 Dataset Details

The FLIR Lepton camera produces thermal images with an image resolution of $n_{row} \times n_{col} = 160 \times 120$ pixels. We have constructed a data set $D_j = \{y_j^i\}_{i=1}^{N=1000}$ of thermal images using the configuration mentioned in the previous section. In this data set, y denotes each thermal image within a set of 1000 images belonging to the class j. In this case, the class j represents the images containing j UAVs. Class 2, for instance, represents the thermal images that contain 2 UAVs. The proposed CNN model receives a subset of this data set as training input. The RGB images of the measurement scenarios with 6 and 8 UAVs flown randomly are displayed in Fig. 3.2. The equivalent thermal images for the 6 and 8 UAV scenarios are shown in Fig. 3.3.

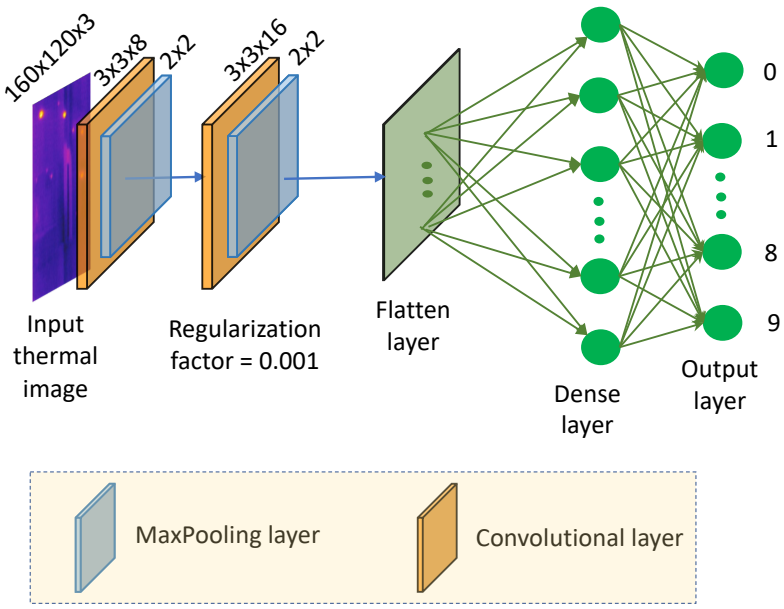


Figure 3.4: Proposed CNN architecture for predicting the number of UAVs based on thermal images [11].

Table 3.3: Details on the architecture of the proposed CNN model [11].

No.	Layer	Output Size	Parameter
1	Input	[(None, 160, 120, 3)]	0
2	Conv2D (1)	[(None, 160, 120, 8)]	224
3	MaxPooling2D (1)	[(None, 80, 60, 8)]	0
4	Conv2D (2)	[(None, 80, 60, 16)]	1168
5	MaxPooling2D (1)	[(None, 40, 30, 16)]	0
6	Flatten	[(None, 19200)]	0
7	Dense	[(None, 10)]	192010

3.2.7 Proposed CNN Architecture

Fig. 3.4 shows the proposed CNN model. The thermal image with a resolution of 160×120 serves as the input to the CNN model. After the input layer, a combination of max pooling and convolution (Conv2D) layers are followed. Next, the image is passed into a convolutional layer with 8 filters and a 3×3 kernel size. This layer extracts features, and the reduced image is sent into a 2×2 max pooling layer. The reduced image is then subjected to a second convolutional process using an L2 regularizer with a 0.001 regularization value. Next, other layers such as the max pooling, flattening, and dense layers are followed. We use the ReLu non-linear activation function to activate each convolutional layer. Lastly, the output layer estimates the likelihood of UAVs in the image using softmax activation. Table 3.3 provides further architectural details.

The UAV-thermal dataset contains $10,000$ thermal images. The thermal images are sorted into 10 folders where each folder corresponds to a class that indicates the number of UAVs in the thermal image. Train, test, and validation is performed

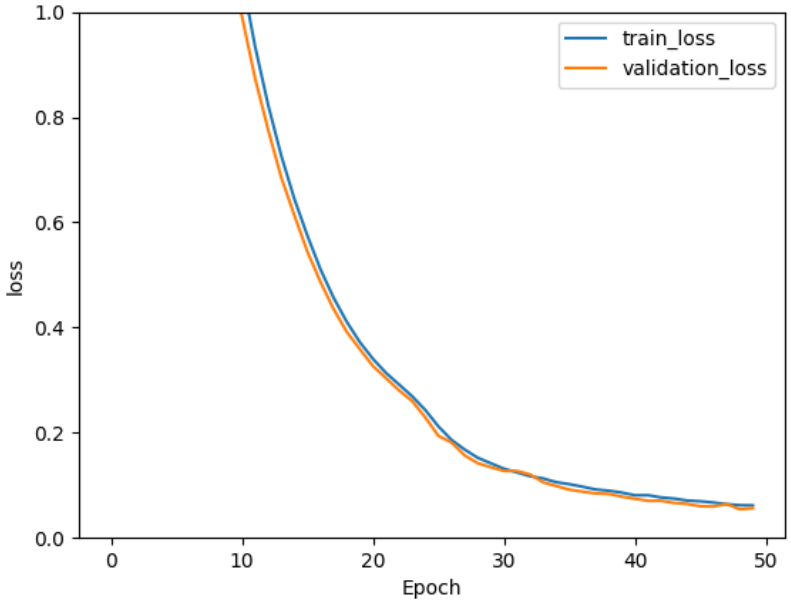


Figure 3.5: Plots depicting the curves for training loss and validation loss [11].

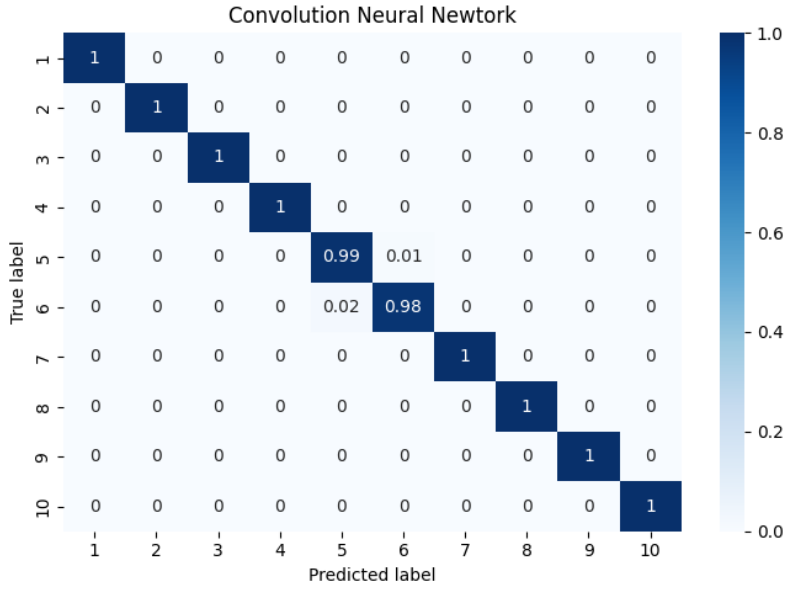


Figure 3.6: Classification performance depicted with the confusion matrix [11].

with ratios of 80%, 10%, and 10%, respectively. To reduce the loss, the sparse categorical cross-entropy function is employed. Additionally, during training, the model employs an adaptive momentum (Adam) optimizer with a value of 10^{-4} as the learning rate.

3.2.8 Results

The thermal images from the dataset are used to train the proposed CNN model. The proposed CNN model is trained by 10-fold cross-validation. The dataset is divided into train, test, and validation subsets with a split ratio of $(80, 10, 10)\%$. The loss curve obtained during training is shown in Fig. 3.5. It can be observed

from Fig. 3.5 that after 40 epochs, both the training and validation losses converge to a minimum. With the 10-fold hold-out validation, it can be observed that the model achieves 99.91% validation accuracy. Additionally, to assess the model performance following training, previously unseen data (10% from the dataset) is used for testing. On the test set, we achieved an accuracy of 99.9% on average. Fig. 3.6 depicts the confusion matrix that represents the performance of the proposed model.

3.3 Acoustic and Machine Learning Based UAV Count Estimation

3.3.1 Motivation

Various sensors can be used to detect and track small-sized UAVs. Radar sensors, vision-based sensors, and acoustic sensors [121], [91] are few of them. However, it has been shown that utilizing acoustic sensors for UAV detection offers a number of benefits over other types of sensors. Compact and inexpensive, acoustic sensors gauge the pressure changes caused by sound waves to measure sound intensity. Acoustic sensors, in contrast to conventional vision and radar-based sensors, are usually omnidirectional. Due to their ability to detect disturbances in all directions, acoustic sensors are a perfect fit for collision-avoidance systems. Additionally, passive acoustic sensors are environmental friendly as they do not emit any radiation. Radar sensors rely on a threshold radar cross-sectional area to ensure detection. In contrast, acoustic sensors are independent of the form factor of UAVs, thereby enhancing their detection performance compared to radar systems. When compared to vision and radar-based systems, the output data rate of acoustic sensors is comparatively modest, which makes data processing and acquisition relatively easier. Thus a major part of this section is focused on the detection and estimation of several UAVs utilizing only their generated acoustic signatures.

3.3.2 Contributions

Prior research in the field has primarily concentrated on identifying a single UAV rather than several UAVs. Moreover, the research predominantly focused on controlled environments and predefined trajectories. This work is different from previous works such that, the primary focus is directed towards acquiring acoustic signatures from a maximum of 10 UAVs. The UAVs are configured to navigate in a random manner across all directions, with different velocities. Our study additionally incorporates the natural background noise throughout the detection process. One outdoor measurement has also been incorporated, that includes noise originating from wind and bird chirping. Supervised learning techniques are employed because of their exceptional efficacy in target detection and classification. While unsupervised techniques have the benefit of extracting intrinsic properties from unlabeled data, they are ineffective in identifying certain specific classes of targets.

Further, the proposed CNN architecture demonstrates superior performance compared to existing machine learning models in terms of both accuracy and model size. Due to its comparatively compact model size, the proposed CNN architecture exhibits reduced resource consumption, facilitating its deployment on lightweight edge-computing devices as Raspberry Pi 4, NVIDIA Jetson Nano, and others. The model is evaluated on these edge computing devices, and the corresponding inference time is also noted. As such the major contributions of this work outlined in Paper D [12] are as follows:

- An acoustic-based UAV dataset is created by utilizing a total of 10 UAVs. An arbitrary number of UAVs are flown randomly within the measurement area and the acoustic field signatures are captured using a cardioid unidirectional microphone.
- Time-frequency algorithms such as CWT are applied to transform the recorded acoustic field signatures to spectrogram images.
- A custom lightweight CNN architecture is designed to estimate the number of UAVs in the scene. The performance of the proposed model is compared with state-of-the-art benchmark machine learning models in terms of accuracy and model size.

3.3.3 System Model

Motorization and the propeller motion of small multirotor UAVs can produce complicated acoustic fields. Thus, UAVs produces acoustic fields that are arguably unique and has complex harmonics. The literature contains various works such as [122], [123], and [124] that address the study of the UAV acoustic fields. Our proposed approach leverages these distinct harmonic acoustic signatures produced by several UAVs to identify and quantify their presence in an indoor environment.

3.3.4 Methodology

Figure 3.7: Schematic representation of the proposed methodology [12].

The proposed method makes use of the acoustic field produced by the UAV propellers and rotors. An acoustic field is generated due to the high-speed rotation of UAV rotors and propellers, creating pressure differentials. In this work, the generated acoustic field signatures are recorded using a cardioid microphone. Audio spectrogram images are produced by capturing and processing acoustic signatures

Table 3.4: Experiment details and measurement settings [12].

Experiment Parameters	**Details**
Measurement duration	5 minutes
Semicircular area	Radius: 5 meters
Microphone	**Shure MV7**
Sampling rate	48000 Hz
Output impedance	314 ohms
Frequency response	50 Hz - 16000 Hz
Total UAVs	**10**
DJI Telllo EDU	4
DJI Mini SE	2
DJI Mini 2	1
DJI Mini 3 Pro	1
DJI Mavic 2 Enterprise	1
SYMA X30	1

from 10 UAV combinations. CNN models are then trained using these audio spectrogram images to determine the approximate number of UAVs present in the scene. Fig. 3.7 depicts a condensed flow diagram of the methodology.

3.3.5 Measurement Setup and Dataset

Measurement setup

Measurement is carried out in an indoor laboratory setting that spans a semicircular area with radius of 5 meter. Acoustic signatures from a maximum of 10 UAVs that are flown arbitrarily in all directions within the measurement area were recorded. The DJI Mavic 2 Enterprise [125], DJI Mini 2 [126], DJI Mini SE [127], DJI Mini 3 Pro [128], DJI Tello EDU [129], and SYMA X30 [117] are the UAV models used in the experiment. With the exception of the DJI Mavic 2 Enterprise, all of the remaining UAVs weighs less than 250 and have a relatively small form factor (about 251 $\times$ 362 $\times$ 70 mm [128]). Small-sized UAVs serve as an excellent candidate to evaluate the estimation performance of the proposed approach in a multi-UAV setting. The DJI Tello UAVs were configured to follow a predetermined flight path. However, the remaining UAVs were human controlled to maneuver in random trajectories so as to replicate near real-time scenarios. Additional information on the experiment is given in Table 3.4.

The Shure MV7 microphone is used to record the UAV acoustic signatures [130]. The microphone is unidirectional with a cardioid type polar pattern and has an output impedance of 314 Ω. Its adjustable gain spans between 0 and +36 dB, and offers a frequency response that spans from 50 Hz to 16000 Hz [131]. Further specifications of the microphone can be found in Table 3.4. The microphone is fixed atop a tripod stand and oriented toward the measurement region. The unidirec-

Figure 3.8: Shure MV7 mounted on a tripod to measure UAV acoustic signals [12].

tional cardioid polar pattern of the microphone allows it to pick up and magnify the acoustic disruptions caused by UAVs flying in front of it. Disturbances originating from the rear end of the microphone are however reduced in power and hence do not affect the output signal. The microphone configuration utilized to record the acoustic signatures from UAVs is shown in Fig. 3.8.

For each measurement, an arbitrary number of UAVs were flown for 5 minutes in the designated area. For instance, acoustic field signatures from 5 randomly flown UAVs were recorded in the fifth measurement of the experiment. The sixth measurement consisted of randomly flying 6 UAVs. Each measurement of the experiment has been planned to use as much different UAV models as feasible in order to increase variability. For measurements that required additional UAVs, however, we had to resort to similar UAV models due to availability limits. A 5 UAV measurement example is illustrated in Fig. 3.9. We performed the 2nd measurement outdoors in order to introduce additional variability to the acoustic field readings. The outdoor measurement also in turn adds noise from wind and bird chirping. The outdoor measurement area is a semicircular area of radius 5 meters as like the indoor measurement area. Moreover, 3 distinct UAV models were used for the initial measurement so as to increase the number of sample values.

Figure 3.9: Recording UAV acoustic signals for a scenario with 5 UAVs [12].

Dataset details

A dataset consisting of UAV audio snippets is created using the acoustic signatures that are recorded from the performed experiments. The duration of each recorded acoustic signature is 5 minutes. The recorded signatures are meticulously cropped to preserve only the audio related to the UAV. With a sampling rate of 48000 Hz, each trimmed audio signal has a duration of 4 minutes and 45 seconds (285 seconds). For each trimmed signal (285 second duration), we divide it into 95 equal parts in order to minimize latency and ensure seamless processing. Each part is a 3 second audio clip that is sampled at 48000 Hz. Thus, the dataset has 1140 three second UAV audio samples in total. The UAV audio clips with a duration of 3 seconds for the 1, 2, 4, 6, 8, and 10 UAVs are provided in Fig. 3.10. As seen in Fig. 3.11, the CWT transform is applied to these audio samples in order to further generate the audio spectrogram images.

3.3.6 Proposed CNN Architecture

CNNs are special deep-learning architectures that identify and categorize objects in images using artificial neural networks. The proposed approach involves using a unique CNN architecture to extract feature data from spectrogram images.

We initially tested the performance of the custom CNN architecture by changing the number of layers as 5, 10, 15, and 20. With a data set split of (80, 10, 10) for training, testing, and validation, we employed a 10-fold cross-validation. It was found that CNNs with 20 layers or more outperformed those with less layers in terms of performance. To find the best-performing architectures, we then adjusted the image resolution and layer count. The performance of the CNN architectures

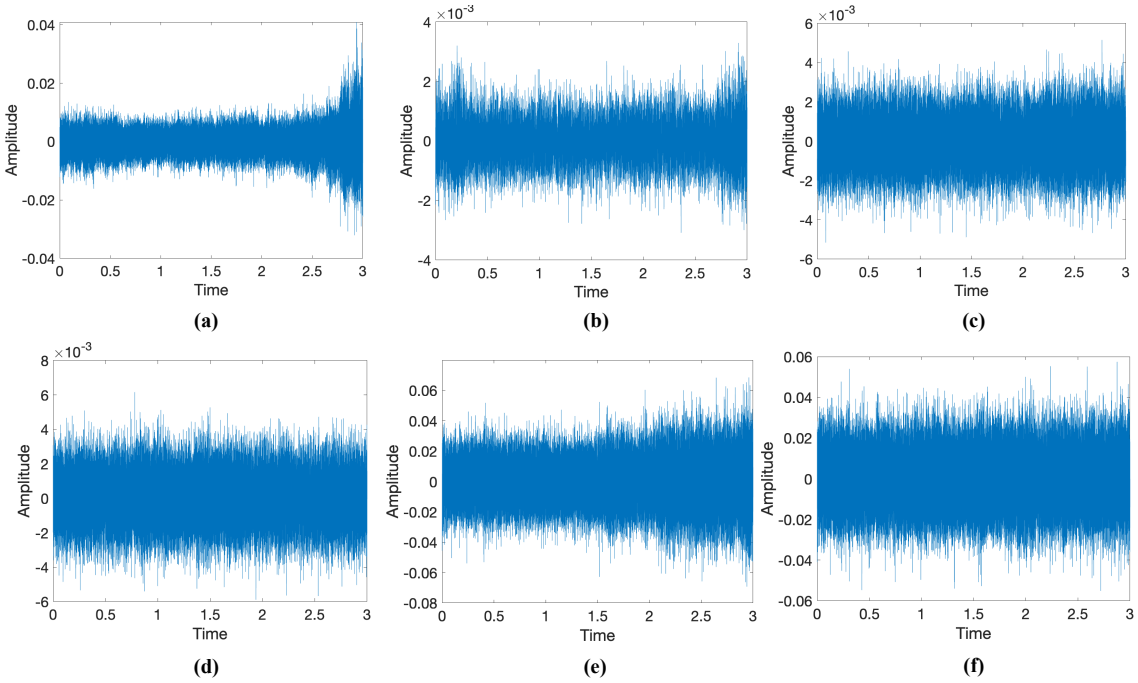


Figure 3.10: Recorded UAV acoustic signatures for (a) 1 UAV, (b) 2 UAVs, (c) 4 UAVs, (d) 6 UAVs, (e) 8 UAVs, and (f) 10 UAVs measurement scenarios [12].

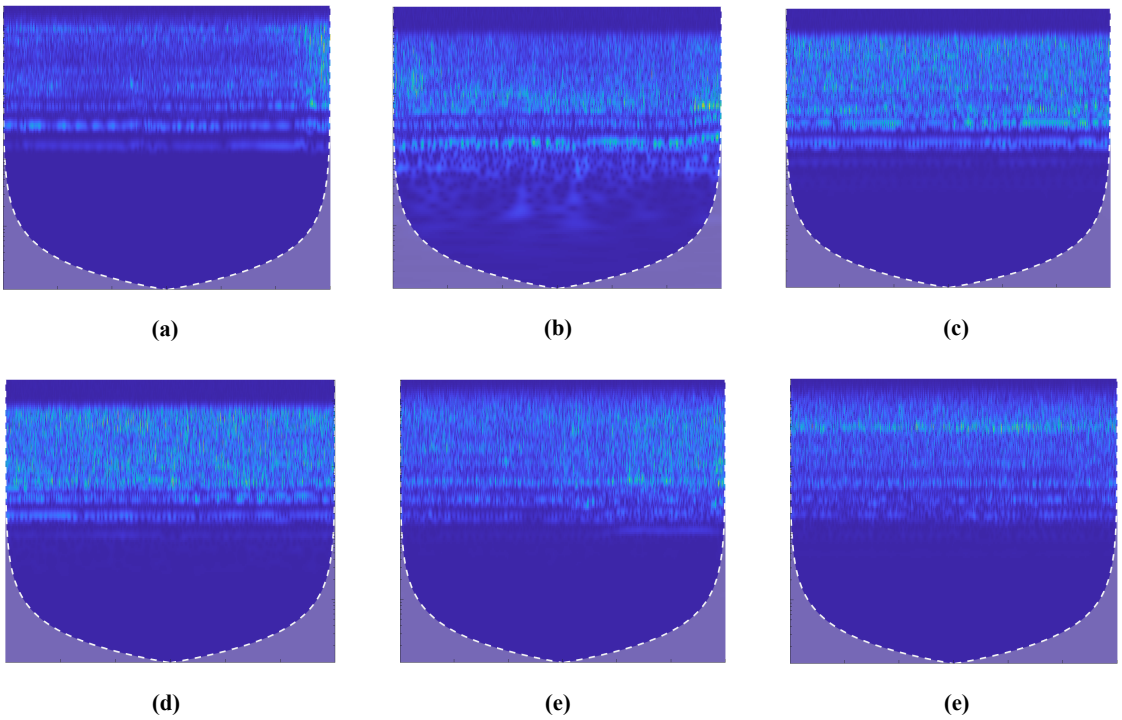


Figure 3.11: Plotted spectrogram images for (a) 1 UAV, (b) 2 UAVs, (c) 4 UAVs, (d) 6 UAVs, (e) 8 UAVs, and (f) 10 UAVs measurement scenarios [12].

with 18 layers and above is given in Table 3.5. The performance variation by altering the image resolution is also noted. Table 3.5 shows that performance starts to progressively rise from 18 layers, peaks at 22 and 24 layers, and then gradually

Table 3.5: Various CNN architectures obtained by varying image resolution and number of layers [12].

Sl. No.	**No of Layers**	**Image resolution**	**Avg. Test Accuracy (%)**
1	18	$300 \times 200 \times 3$	92.29
2	18	$400 \times 300 \times 3$	92.40
3	18	$500 \times 400 \times 3$	86.05
4	18	$600 \times 500 \times 3$	**94.73**
5	18	$700 \times 600 \times 3$	93.55
6	20	$300 \times 200 \times 3$	89.37
7	20	$400 \times 300 \times 3$	88.50
8	20	$500 \times 400 \times 3$	89.56
9	20	$600 \times 500 \times 3$	88.89
10	20	$700 \times 600 \times 3$	88.00
11	22	$300 \times 200 \times 3$	92.19
12	22	$400 \times 300 \times 3$	**94.24**
13	22	$500 \times 400 \times 3$	**94.74**
14	22	$600 \times 500 \times 3$	93.86
15	22	$700 \times 600 \times 3$	**94.05**
16	24	$300 \times 200 \times 3$	92.98
17	24	$400 \times 300 \times 3$	92.98
18	24	$500 \times 400 \times 3$	93.56
19	24	$600 \times 500 \times 3$	93.86
20	24	$700 \times 600 \times 3$	**94.24**
21	26	$300 \times 200 \times 3$	91.13
22	26	$400 \times 300 \times 3$	91.61
23	26	$500 \times 400 \times 3$	89.76
24	26	$600 \times 500 \times 3$	86.34
25	26	$700 \times 600 \times 3$	88.11

declines. It was found that five CNN architectures (shown in bold in Table 3.5) performed with an average test accuracy of greater than 94%. Out of the five, the CNN architecture with 22 layers and supporting input image resolution of $500 \times 400 \times 3$ was selected after taking into account other performance metrics and parameters.

Table 3.6 and Fig. 3.12 depict the 22-layer proposed CNN architecture. After applying CWT on the audio clips, spectrogram images with a resolution of $836 \times 716 \times 3$ pixels were produced. These spectrogram images are then further scaled to $500 \times 400 \times 3$ pixels to train the proposed CNN architecture. The input layer of the proposed CNN architecture then receives the rescaled spectrogram images as input. The convolutional layers in the proposed architecture have kernel dimensions of 4×4. To extract embedded feature representations, 8 or 16 convolutional kernels

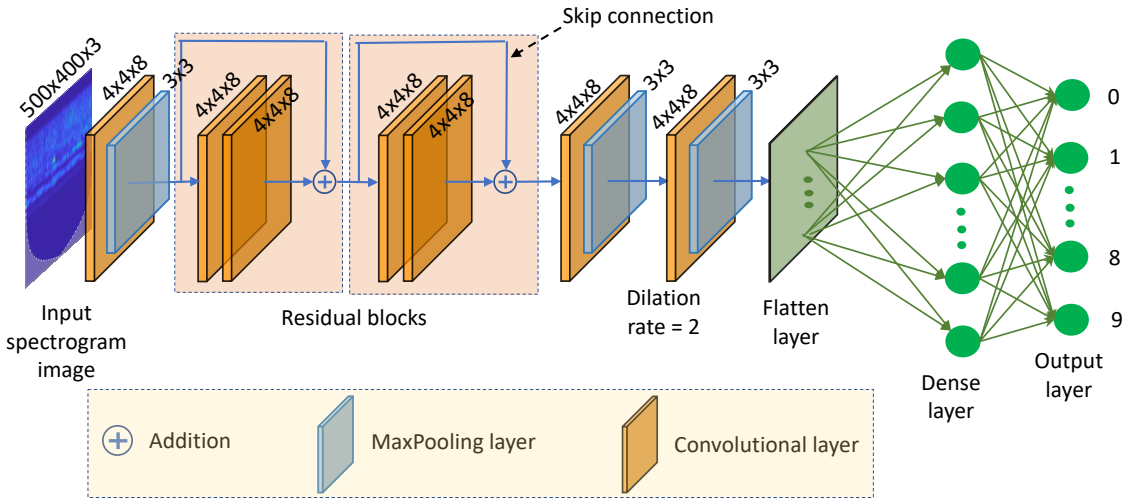


Figure 3.12: The proposed CNN architecture [12].

are primarily employed. In the final convolutional layer, we also employ dilation with a dilation factor of 2. Dilation introduces holes between neighbouring elements thereby expanding the kernel size. As a result, it is possible to capture intrinsic sequence information with a wider field of view [132].

In order to increase the number of layers in the network without sacrificing performance, the proposed CNN architecture utilizes residual blocks. A total of 2 residual layers are used in our proposed model. Fig. 3.12 shows the architecture of the residual blocks employed in the proposed model. We have also introduced 3×3 max pooling layers throughout the proposed architecture. The input feature representation is downsampled with the help of the max pooling layer [133]. Max pooling improves computing performance for subsequent layers by effectively removing translational invariances from the input representation. Batch normalization is also used at the input layer in the proposed model, as illustrated in Table 3.6. The problem of internal covariate shift [134] is resolved by batch normalization by normalizing the input distribution through a process of rescaling and re-centering. In the final layers of the proposed CNN architecture, we use the flatten and dense layers. The input vector is converted into a 1-dimensional output by the flatten layer, which is then supplied to the dense layer. The output of the dense layer are class probabilities that can be utilized for detection and classification tasks. As we are predicting a maximum of 10 UAVs, there are 10 outputs from the dense layer in the proposed architecture.

The Adam optimizer [135] is utilized to train the proposed CNN model using the tanh activation function. The batch size has been increased to 16 and the learning rate to 0.001 in order to reduce the training-related variations in the accuracy/loss curve. Using two Tesla V100-SXM3 GPUs with 32 GB RAM [136], the Keras deep learning library [137] is used for training the proposed CNN model.

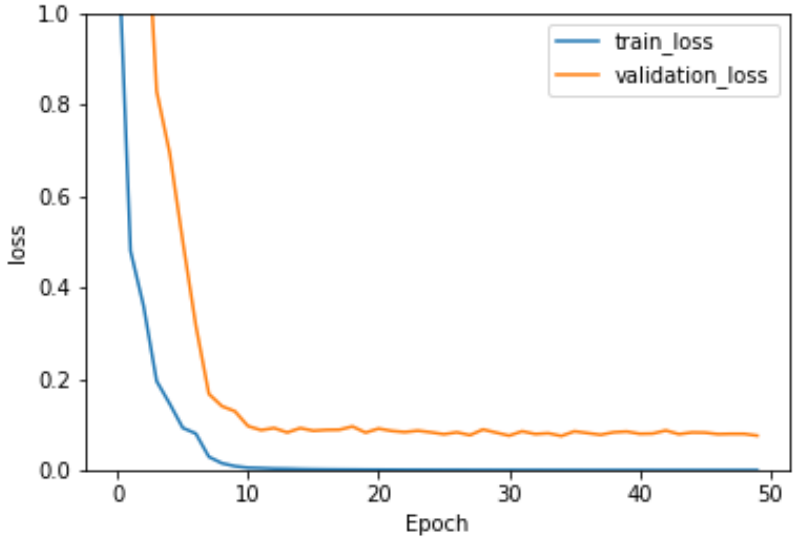
Table 3.6: Architecture details of the proposed CNN model arranged layerwise [12].

No.	Layer	Output Size	Parameter
1	Input	[(None, 500, 400, 3)]	0
2	Batch Normalization (1)	[(None, 500, 400, 3)]	12
3	Conv2D (1)	[(None, 500, 400, 8)]	392
4	MaxPooling2D (1)	[(None, 166, 133, 8)]	0
5	Conv2D (2)	[(None, 166, 133, 8)]	1032
6	Batch Normalization (2)	[(None, 166, 133, 8)]	32
7	Conv2D (3)	[(None, 166, 133, 8)]	1032
8	Batch Normalization (3)	[(None, 166, 133, 8)]	32
9	Add (1)	[(None, 166, 133, 8)]	0
10	Activation (1)	[(None, 166, 133, 8)]	0
11	Conv2D (4)	[(None, 166, 133, 16)]	1032
12	Batch Normalization (4)	[(None, 166, 133, 8)]	32
13	Conv2D (5)	[(None, 166, 133, 16)]	1032
14	Batch Normalization (5)	[(None, 166, 133, 8)]	32
15	Add (2)	[(None, 166, 133, 8)]	0
16	Activation (2)	[(None, 166, 133, 8)]	0
17	Conv2D (6)	[(None, 166, 133, 16)]	2064
18	MaxPooling2D (2)	[(None, 55, 44, 16)]	0
19	Conv2D (7), Dilation rate = 2	[(None, 55, 44, 64)]	16448
20	MaxPooling2D (3)	[(None, 18, 14, 64)]	0
21	Flatten	[(None, 16128)]	0
22	Dense	[(None, 10)]	161290

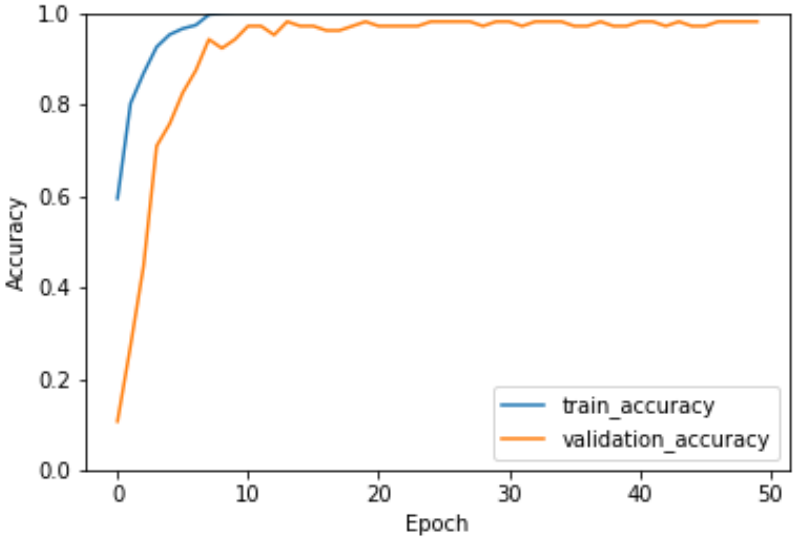
3.3.7 Results

The proposed CNN model is trained using the spectrogram images that are produced after applying the CWT transform. 80% of the dataset is reserved for training, 10% for validation, and the remaining 10% is used for testing. We trained each fold for 50 epochs with a batch size of 16 and a learning rate of 0.001 using 10-fold cross-validation [138]. For the 10-fold cross-validation, we split the dataset as 80% for training and 10% for validation. The remaining 10% of the dataset acts as a test set so as to compute the detection accuracy. The loss and accuracy curves obtained during training are displayed in Fig. 3.13a and 3.13b respectively. Fig. 3.13a shows that after 10 epochs, there is a significant drop in both training and validation loss. As a result, after 10 epochs, the accuracy curves for training and validation converge close to 1, suggesting that the proposed model necessitate less training time.

The model performance on the test set was assessed after training. The confusion matrix that was plotted after testing the proposed model on the test set is shown in Fig. 3.14. It is evident from Fig. 3.14 that in more than 90% of the cases, the proposed model can accurately forecast the number of UAVs. For the remaining 10% of the cases, the proposed model is inaccurate in its prediction. This may be because



(a) Loss curve



(b) Accuracy curve

Figure 3.13: Loss curve and accuracy curve obtained after training [12].

the resulting signal can become unresolvable due to the superposition of acoustic signatures from comparable UAV types. As seen in Table 3.7, the performance of the proposed model is also contrasted with that of common benchmark machine learning models. For the purpose of computing the performance metrics on the benchmark models, we split the data set in the same split ratio as before. The spectrogram images are then provided as input to benchmark models after undergoing rescaling to $224 \times 224 \times 3$ pixels. In comparison to the benchmark models, the proposed model achieves a comparatively high test accuracy of 93.33%, as shown in Table 3.7. Table 3.7 further shows that, in comparison to the benchmark models, the proposed model requires a storage space of just 2.34 MB, allowing for seamless deployment

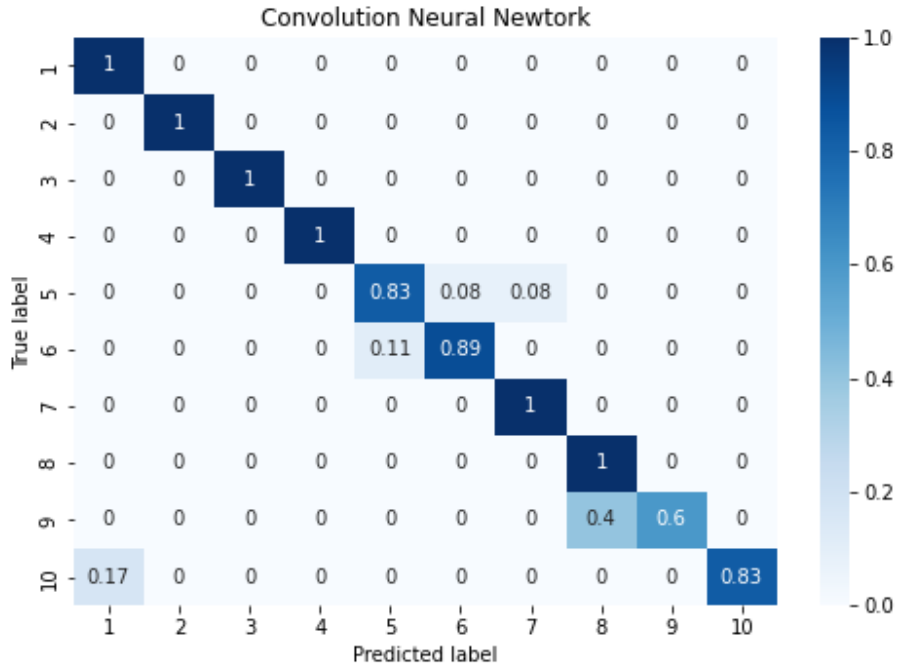


Figure 3.14: Classification performance depicted with the confusion matrix for the proposed model [12].

and portability across a wide range of edge computing devices. Furthermore, when compared to other benchmark models, our approach uses fewer number of total parameters. Additionally, the majority of the total parameters that the proposed model utilizes are trainable parameters, demonstrating effective parameter usage. In terms of the number of floating point operations (FLOPs) [141], Table 3.7 compares the computational performance of our model to that of other benchmark models. GFLOPs are the metric used to gauge the number of FLOPs, with 1 GFLOP equal to 10^9 FLOPs. Standard open-source software from TensorFlow [142] and PyTorch [143] is used to acquire the FLOP count. As can be seen, our model has fewer FLOPs than the standard benchmark models with the exception of MobileNetV3Large and MobileNetV3Small models. This difference may be attributed to the width and resolution multiplier parameters introduced in the MobileNet series [55] that are in turn responsible for decreasing the computing cost.

We have also deployed the proposed model on to edge computing platforms such as NVIDIA Jetson Nano, Raspberry Pi 4 Model B, and NVIDIA Jetson AGX Xavier. On each of these devices, we compute the inference time for the proposed model. For time-critical applications, the inference time computation can aid in determining feasibility to deploy the proposed model. We used a randomly chosen 10% of the dataset as test set for each of the three devices. With the Raspberry Pi 4 board, the trained model was executed for an average inference time of approximately 127 milliseconds over 10 iterations. However, with the Jetson Nano, the average inference time was approximately 219 milliseconds. This is to be expected as the GPU resources of the Jetson Nano are not being completely utilized by the TensorFlow Lite models. For GPU operations, even though CUDA is not supported

Table 3.7: Comparison of various performance metrics between benchmark models and the proposed CNN model [12].

Sl. No.	**Model**	**Total Parameters**	**Avg. Test Accuracy (%)**	**Model Size (MB)**	**Floating Point Operations (GFLOPs)**
1	DenseNet121 [52]	7,047,754	83.77	28.98	2.88
2	DenseNet169 [52]	12,659,530	87.28	51.76	3.42
3	DenseNet201 [52]	18,341,194	88.68	74.68	4.37
4	EfficientNetB0 [54]	4,062,381	83.85	16.74	0.4
5	EfficientNetB1 [54]	6,588,049	81.05	27.01	0.59
6	EfficientNetB2 [54]	7,782,659	80.08	31.80	0.68
7	EfficientNetB3 [54]	10,798,905	84.21	43.94	0.99
8	EfficientNetB4 [54]	17,691,753	78.24	71.66	1.54
9	EfficientNetB5 [54]	28,534,017	79.73	115.20	2.41
10	EfficientNetB6 [54]	40,983,193	82.45	165.18	3.43
11	EfficientNetB7 [54]	64,123,297	81.40	257.98	5.27
12	InceptionResnetV2 [139]	54,352,106	77.63	218.77	6.55
13	InceptionV3 [56]	21,823,274	78.42	87.97	2.89
14	MobileNetV2 [55]	2,270,794	86.75	9.49	0.32
15	MobileNetV3Large [140]	4,239,242	86.92	17.42	0.23
16	MobileNetV3Small [140]	1,540,218	81.84	6.55	0.06
17	NASNetMobile [57]	4,280,286	77.71	18.48	0.27
18	ResNet101V2 [53]	42,647,050	89.64	171.37	8.28
19	ResNet152V2 [53]	58,352,138	87.63	234.50	12.5
20	ResNet50V2 [53]	23,585,290	90.35	94.82	3.97
21	VGG16 [58]	14,719,818	87.98	58.98	15.5
22	VGG19 [58]	20,029,514	87.98	80.22	19.6
23	Xception [59]	20,881,970	82.89	83.96	0.36
24	**Proposed model**	**184,462**	**93.3**	**2.34**	**0.25**

with TensorFlow Lite [144], the Jetson devices provide the necessary CUDA support [145]. Hence, the observed increase in inference time is expected. Similarly, when executed on the NVIDIA Jetson AGX Xavier, the trained model was able to output an average inference time of approximately 81.4 milliseconds. Comparing the Jetson Nano and Raspberry Pi 4 to the Jetson AGX Xavier, the latter was faster in inferring the predictions. The faster inference time, despite the GPU not being used, can be ascribed to the availability of additional RAM, around 32 GB, which can subsequently improve system performance. Moreover, compared to the other two devices, the Jetson AGX Xavier has access to additional computing resources. The inference time obtained for the proposed model and benchmark models on different edge computing devices are listed in Table 3.8. For the purpose of determining the inference time for the benchmark models, we divide the data set based on same split ratio as mentioned before. On all three embedded devices, it is evident that the proposed model outperforms majority of the benchmark models with a faster inference time. However, the MobileNet models exhibit lower inference time as compared to the proposed model. This may be because of the introduced width and resolution multiplier parameter in the MobileNet series that further lowers the computational

Table 3.8: Computed inference time values when deployed on various edge computing devices [12].

Sl. No	Model	Inference time (seconds)		
		Raspberry Pi	**Jetson Nano**	**Jetson AGX Xavier**
1	DenseNet121 [52]	0.692	0.841	0.413
2	DenseNet169 [52]	0.832	1.009	0.483
3	DenseNet201 [52]	1.088	1.275	0.614
4	EfficientNetB0 [54]	0.389	0.372	0.116
5	EfficientNetB1 [54]	0.591	0.546	0.178
6	EfficientNetB2 [54]	0.624	0.592	0.180
7	EfficientNetB3 [54]	0.843	0.794	0.248
8	EfficientNetB4 [54]	1.162	1.147	0.359
9	EfficientNetB5 [54]	1.724	1.672	0.528
10	EfficientNetB6 [54]	2.339	2.173	0.832
11	EfficientNetB7 [54]	3.252	2.971	1.158
12	InceptionResNetV2 [139]	1.637	1.828	0.797
13	InceptionV3 [56]	0.700	0.819	0.378
14	MobileNetV2 [55]	0.088	0.112	0.046
15	MobileNetV3Large [140]	0.072	0.090	0.036
16	MobileNetV3Small [140]	0.022	0.028	0.011
17	NASNetMobile [57]	0.320	0.256	0.125
18	ResNet50V2 [53]	0.912	1.008	0.440
19	ResNet101V2 [53]	1.879	2.053	0.858
20	ResNet152V2 [53]	2.820	3.159	1.285
21	VGG16 [58]	3.903	4.042	1.699
22	VGG19 [58]	5.031	6.981	2.184
23	Xception [59]	1.236	2.206	0.520
24	**Proposed model**	**0.127**	**0.219**	**0.081**

cost of the model.

3.4 Angle of Arrival Estimation and Classification of Targets Using Rotating mmWave FMCW Radar and Yolov3

3.4.1 Motivation

There is an increasing need for improved detection, localization, and tracking techniques as UAVs become more and more popular. Advanced sensor technologies are fitted onto UAVs and GCSs so as to provide better target localization. LiDARs, RGB cameras, and ultrasonic sensors are among them. However, the sensing capabilities of majority of these sensors are inadequate under harsh weather and insufficient lighting conditions. In this regard, when compared to other sensors, the mmWave FMCW radars have demonstrated relatively better performance. The mmWave FMCW radars are compact lightweight sensors that offer exceptional target detec-

tion capabilities. The radars offer good velocity estimation capability together with a radial distance measurement range of 0.2 meters to 300 meters. Further, they also provide accurate range and velocity resolutions. In addition, they are a promising solution for GCS and UAV applications due to their performance in harsh weather and environmental situations.

Despite the aforementioned benefits, the AoA estimation and AoA resolution of mmWave FMCW radars are inadequate [7]. For the radar to estimate the AoA accurately from a single target, at least one transmitter and two receivers are required. The AoA estimation can be successfully improved by increasing the number of transmitter and receiver pairs, given that the trade-off between hardware complexity and larger form factor are considered. Further, it is even more challenging to estimate AoA for several objects in a multi-target environment. Therefore, our goal in this work is to improve the AoA estimation for mmWave FMCW radars, so as to enable accurate target detection and localization.

3.4.2 Contributions

Even though there have been several attempts to improve the AoA estimation capabilities of mmWave FMCW radars as mentioned in section 2.6.3 of chapter 2, most of these approaches demand high computational complexity and additional hardware resources. Our work aims to resolve some of these issues. As such the major contributions of this work are:

- We propose a single rotating mmWave FMCW radar setup to obtain accurate multi-target AoA estimation.
- Our proposed method utilizes computer vision algorithms such as Yolov3 to estimate the AoA of multiple targets with a 180° field of view.
- In addition to AoA estimation, the proposed approach also provides an accurate classification of the targets based on the obtained range-angle signatures.

3.4.3 System Model and Problem Formulation

The operation of mmWave FMCW radars involves transmitting high-frequency chirp signals. When these chirp signals are obstructed in its path by obstacles, they are reflected back. The receiving antennae captures these reflected chirps and utilizes them for additional processing. The IF signal is then obtained by feeding the transmitted and received chirp signals into a mixer. The analog IF signal is then forwarded to a high-speed ADC for sampling and generating the corresponding digital samples. To determine the radial range, radial velocity, and AoA of the target, these raw IF samples then undergo further processing [146].

To estimate the range of the target, FFT transformation is applied on the obtained IF samples [147]. Thus, the range denoted by R is computed as,

$$R = \frac{cf_{IF}}{2S}, \tag{3.1}$$

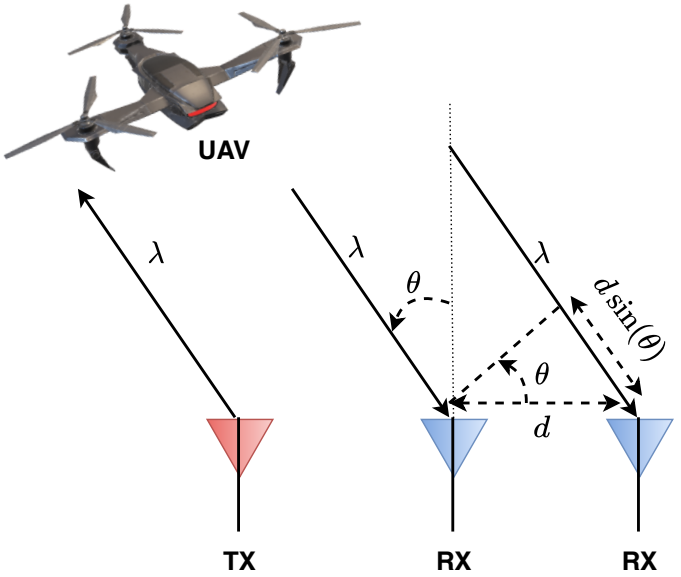


Figure 3.15: Estimating the angle of arrival [13].

where S is the slope of the mmWave FMCW radar, f_{IF} is the IF signal frequency, and c denotes the velocity of light in vacuum ($3\text{x}10^8$ m/s).

AoA estimation depends directly on the number of receiver antennas on the radar. The mmWave FMCW radar is manufactured such that there exists between a target object and the individual receiver antennas a differential distance. The differential distance is directly proportional to the phase change that is obtained by applying a second FFT transformation on the output range profile. The angle-FFT is obtained by performing the second FFT operation over the different receiver antennas. Now, the AoA estimation of target objects in the scene can then be obtained utilizing the angle-FFT plot. Note that the 2D-FFT is applied over spatially separated receiver antennas to obtain the angle-FFT, whereas, the 2D-FFT leverages the phase difference between time-separated chirp waveforms to estimate the velocity [148]. Considering the distance between adjacent receiving antennas is d and the phase difference between the received chirp waveforms is ϕ, then the following is true,

$$\phi = \frac{2\pi d \sin\theta}{\lambda}, \tag{3.2}$$

where the wavelength of the chirp signal is denoted by λ. Hence, the estimated value for AoA given by θ is computed as,

$$\theta = \sin^{-1}(\frac{\lambda\phi}{2\pi d}) \tag{3.3}$$

Fig. 3.15 provides a visual depiction of the AoA estimation. As observed from (3.2) and (3.3), the value of ϕ is more sensitive to changes in the value of θ, especially when $\theta = 0°$. Further, the sensitivity of ϕ to θ falls rapidly with increasing θ value. This can be attributed to the non-linear relationship that exists between ϕ and θ with $\phi \propto \sin\theta$. Thus, with the increase in θ, the sensitivity of $\sin\theta$ decreases. As a result, when the target is positioned perpendicular to the radar, AoA estimation is more accurate [148].

As observed, the AoA estimation requires the presence of atleast one transmitter and two receivers. In this work, we attempt to obtain accurate AoA estimation with the use of a single transmitter and receiver antennae.

3.4.4 Measurement Setup and Dataset

Measurement setup

The measurement equipment comprises of a mmWave FMCW radar that is securely installed atop a programmable rotor. The rotor is fixed firmly on stationary tripod stand. The rotor is portable and uses battery sources for power. The mmWave FMCW radar is installed on the rotor head. Depending on the requirements, the rotor head is capable to rotate the firmly fixed radar in the azimuth for the whole 360°. The programmable parameters of the rotor include the rotational velocity, the rotational direction, and the rotational angle. The rotor can be rotated at a velocity that can be adjusted based on the quantity of chirp frames the radar transmits in a second. Here, the set of K equally spaced chirp waveforms sent by the radar is referred to as a chirp frame, or simply a frame [146]. The rotor's rotational velocity has a direct impact on the accuracy of the AoA estimation. If the rotor rotates at high rotational velocity, then the radar captures fewer frames in each second. On the otherhand, a low rotational velocity corresponds to a higher number of radar frames collected each second. Higher frames per second are required to obtain better AoA estimate performance, since higher the number of collected frames, larger the amount of information that can be retrieved from multiple targets present in the scene. In this work, in order to record at least one frame for each degree of radar rotation, we have configured the rotor's rotating velocity to be low.

Let θ_{FoV} be the intended FoV that the rotor should cover in t seconds. The radar transmits n_f frames in t seconds. Taking into account that the whole θ_{FoV} is split into smaller angle bins (θ_b), we get the subsequent relationship:

$$\theta_b = \frac{\theta_{FoV}}{n_f} \tag{3.4}$$

The desired FoV in our experiment is 180°. For the radars AWR2243 [149] and AWR1843 [150], the n_f is 800 and 200 frames, respectively. A batch of 128 chirp waveforms called a frame are transmitted by the mmWave FMCW radar. For AWR2243 and AWR1843, this translates to 0.225° and 0.9° per frame, respectively. In our experiment setup, the radar is oriented vertical relative to the ground plane in order to maximize the FoV at elevation.

Our experiment uses two mmWave FMCW radars from Texas Instruments: AWR2243 [149] and AWR1843 [150]. The frequency ranges of both radars are the identical, ranging from 77 to 81 GHz. They also have the identical chirp slope, sampling rate, ADC samples, RF bandwidth, and number of transmitter and receiver pairs. However, the AWR1843 uses 800 frames, while the AWR2243 uses 200 frames. The key specifications for both the radars is provided in Table 3.9.

Table 3.9: Specifications for mmWave FMCW radars, AWR1843 and AWR2243 [13].

Radar Parameter	**Value**
No. of ADC samples	256
Sampling Rate	10 MSPS
No. of Chirps	128
Frequency Range (RF)	$77 - 81$ GHz
No. of Tx antennas	3
No. of Rx antennas	4
No. of frames (AWR2243)	200
No. of frames (AWR1843)	800
Transmission Power	12 dBm
Bandwidth (RF)	1798.92 MHz
Chirp Slope	29.982 MHz/μs
Frame periodicity	40 ms
Rx Noise Figure	14 dB (76 to 77 GHz) 15 dB (77 to 81 GHz)

Table 3.10: Experiment settings [13].

Experiment Parameter	**Value**
Measurement Range	upto 26 meters
Car Size	$431.5 \times 178.0 \times 160.5$ cm^3
Human Height	172 cm
UAV Size	$32.24 \times 24.2 \times 8.4$ cm^3

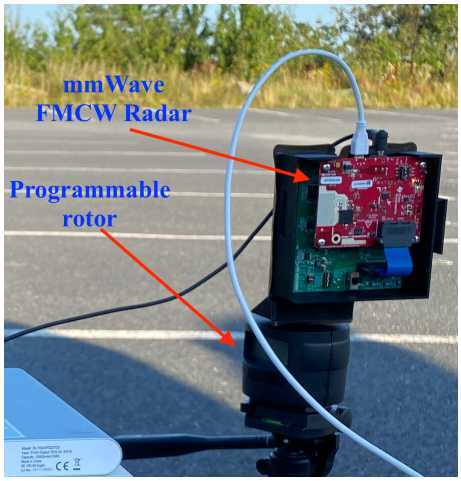


Figure 3.16: mmWave FMCW radar mounted on the rotating rotor [13].

The experiment is conducted at an outdoor parking lot with a car, a UAV, and people positioned at different angles to the radar. For each measurement, the UAV, car, and people are positioned at various points within a radial distance of $\approx$ 26 meters and 180° FoV in front of the radar. We restrict the radial distance within

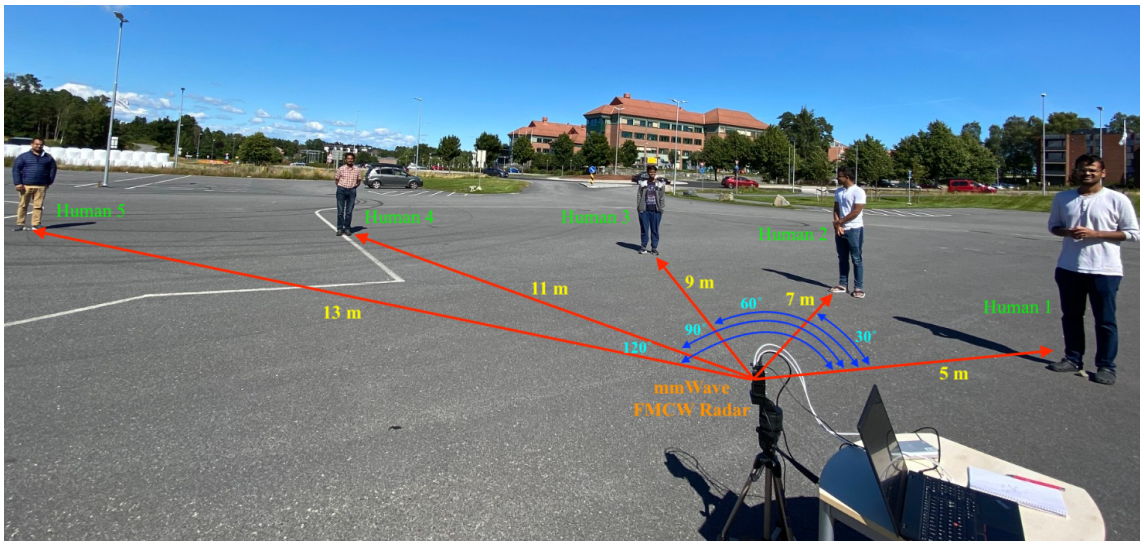


Figure 3.17: Measurement case AA from Set3_OnlyHumans [13].

the allowed measurement range by capping it at ≈ 26 meters. Nevertheless, objects up to 300 meters away can be measured by the mmWave FMCW radars [149], [150]. For every measurement, a total of 800/200 frames containing the raw IF signal are recorded, depending on the radar version (AWR1843/AWR2243) that is being used. Each frame of the radar comprises of 128 chirps with a duration of about 40 ms. Additionally, depending on the radar model, each frame of the radar delivers raw IF signal data corresponding to a 0.225°/0.9° FoV. To acquire the range profile for every frame, the gathered raw IF signal data is post-processed in MATLAB [151]. In order to create a 180° FoV range-angle map of the entire measurement scenario, the range profiles for each frame are finally stitched together [24]. The experimental configuration of the rotating mmWave FMCW radar, which records the raw IF signals, is depicted in Fig. 3.16. Additional experiment parameters are listed in Table 3.10.

Dataset details

Based on the measurement setup described above, we have collected the raw IF signal data for various target objects from the mmWave FMCW radars, AWR2243 and AWR1843. The target objects consist of human, car, and UAV combinations that are positioned at different locations in front of the radar. The AWR2243 mmWave FMCW radar was utilized to record the IF signals from humans, cars, and UAVs. In a similar vein, the IF signals from humans were obtained using the AWR1843 mmWave FMCW radar. MATLAB is utilized to process the captured raw IF signals and derive the corresponding range profile for every frame. The radar range-angle images are created by stitching together the range profiles for each frame. On the basis of these range-angle images, three datasets have been developed. The first dataset includes humans, car, and UAV; the second dataset includes a mix of UAV and car. Range-angle images of just human targets are included in the third dataset. Set1_UavCarHumans, Set2_UavCar, and Set3_OnlyHumans are the names of the

Table 3.11: Measurement cases for Set1_UavCarHumans [13] (Angle in degrees).

Range (m) → / Cases ↓	5	7	9	11	13	15	17	19	21	23	25
AA	U(0°)		H1(30°)	H2(60°)	H3(90°)	H4(120°)	H5(150°)	C(0°)			
BB		U(30°)		H5(180°)	H1(60°)	H2(90°)	H3(120°)	H4(150°)	C(30°)		
CC			U(60°)	H1(0°)	H2(30°)		H3(90°)	H4(120°)	H5(150°)	C(60°)	
DD				U(90°)	H1(0°)	H2(30°)	H3(60°)		H4(120°)	H5(150°)	C(90°)
EE	U(120°)	H1(60°)	H2(90°)		H3(150°)	H4(0°)	C(180°)	H5(30°)			
FF		U(150°)	H5(0°)			C(150°)	H1(30°)	H2(60°)	H3(90°)	H4(120°)	
GG	H1(60°)	H2(90°)	U(180°)	H3(150°)	C(120°)				H4(0°)	H5(30°)	
HH	U(30°)	H4(0°)	H3(150°)	H2(120°)				H1(90°)		C(180°)	H5(60°)
II	H1(90°)	H2(120°)		U(30°)	H3(180°)	H4(60°)	H5(0°)				C(150°)
JJ	U(150°)	H3(180°)	H1(120°)						H2(60°)	H4(90°)	C(30°)
KK	H3(150°)	U(60°)			H4(90°)		H5(30°)		H1(180°)	H2(0°)	C(120°)
LL	H2(180°)	H1(120°)	U(30°)	H3(60°)		H4(90°)		H5(150°)			C(0°)
MM		H2(0°)	U(90°)	H3(30°)	H4(120°)	H1(180°)	H5(150°)				C(180°)
NN	U(90°)	H1(30°)	H3(60°)	H4(120°)	H5(150°)		H2(0°)	C(180°)			

U - UAV, C - Car, H1 - Human 1, H2 - Human 2, H3 - Human 3, H4 - Human 4, H5 - Human 5.

Table 3.12: Measurement cases for Set2_UavCar [13] (Angle in degrees).

Range (m) → / Cases ↓	5	7	9	11	13	15	17	19	21	23	25
AA		U(0°)									C(60°)
BB			U(0°)					C(60°)			
CC				U(0°)			C(90°)				
DD	U(60°)					C(120°)					
EE			U(60°)				C(150°)				
FF				U(60°)			C(180°)				
GG		U(90°)							C(180°)		
HH		U(120°)								C(150°)	
II			U(120°)			C(0°)					
JJ				U(120°)					C(0°)		
KK			U(150°)				C(0°)				
LL				U(150°)						C(90°)	
MM	U(180°)					C(60°)					
NN		U(180°)							C(120°)		
OO				U(180°)							C(30°)

U - UAV, C - Car.

first, second, and third datasets, respectively. After processing from MATLAB, the range-angle images for Set1_UavCarHumans and Set2_UavCar have a resolution of 875×656 pixels. In a similar vein, Set3_OnlyHumans contains range-angle images with a resolution of 1167×875 pixels.

In Tables 3.11, 3.12, and 3.13, the dataset measurement cases are described in detail. Depending on the dataset, the various cases are labeled as AA, BB, CC,....OO. An object that is positioned at a given distance and angle from the radar is represented by the entries of the table. The labels used in the table entries, together with their corresponding abbreviations, are provided at the bottom of each table. Accordingly, the first entry in Table 3.13 case AA shows a person at 5 m and 0° in front of the radar. The remaining entries can be deciphered as follows: a second person standing at 7 m and 30°, a third individual at 9 m and 60°, a fourth human

Table 3.13: Measurement cases for Set3_OnlyHumans [13] (Angle in degrees).

Range (m) → / Cases ↓	5	7	9	11	13	15	17	19	21	23	25
AA	H1(0°)	H2(30°)	H3(60°)	H4(90°)	H5(120°)						
BB		H3(0°)	H4(30°)	H5(60°)		H1(150°)	H2(180°)				
CC			H5(0°)		H1(90°)	H2(120°)	H3(150°)	H4(180°)			
DD				H1(30°)	H2(60°)	H3(90°)	H4(120°)	H5(150°)			
EE				H2(0°)	H3(30°)	H4(60°)	H5(90°)		H1(180°)		
FF					H4(0°)	H5(30°)		H1(120°)	H2(150°)	H3(180°)	
GG							H1(60°)	H2(90°)	H3(120°)	H4(150°)	H5(180°)
HH						H1(0°)	H2(30°)	H3(60°)	H4(90°)	H5(120°)	
II							H2(0°)	H3(30°)	H4(60°)	H5(90°)	H1(150°)
JJ								H2(0°)	H3(30°)	H4(60°)	H5(90°), H1(120°)
KK	H4(150°)	H5(180°)							H1(0°)	H2(30°)	H3(60°)
LL	H3(120°)	H4(150°)	H5(180°)							H1(0°)	H2(30°)
MM	H2(90°)	H3(120°)	H4(150°)	H5(180°)							H1(0°)
NN	H1(60°)	H2(90°)	H3(120°)	H4(150°)	H5(180°)						
OO	H1(30°)	H2(60°)	H3(90°)	H4(120°)	H5(150°)						

H1 - Human 1, H2 - Human 2, H3 - Human 3, H4 - Human 4, H5 - Human 5.

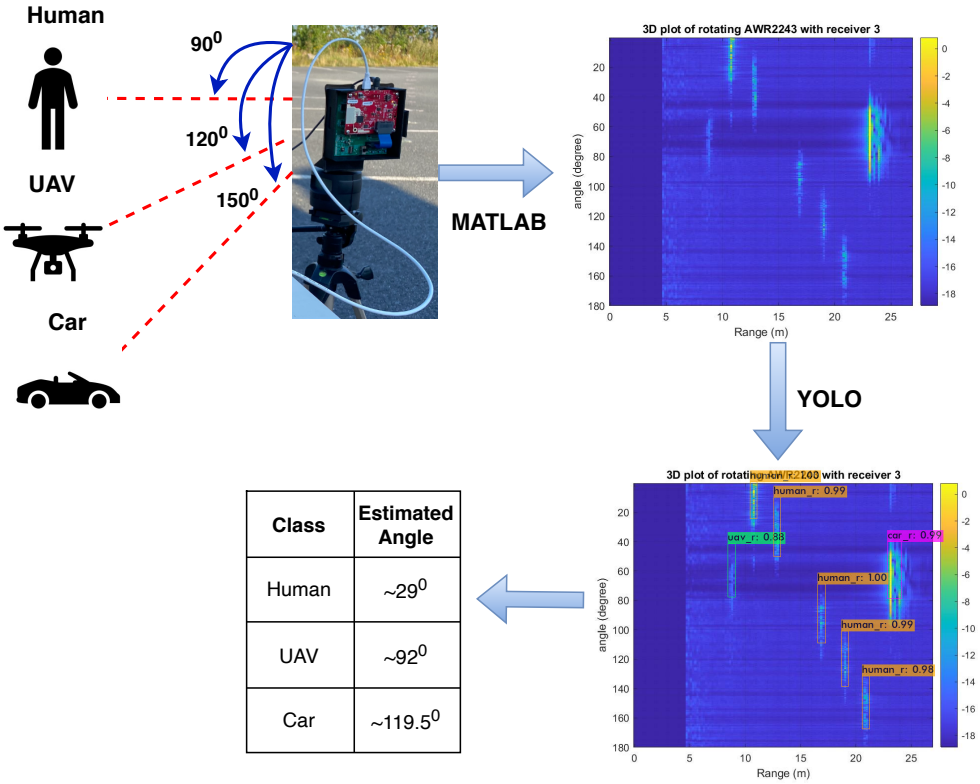


Figure 3.18: Block diagram of the proposed AoA estimation approach [13].

at 11 m and 90°, and a fifth at 13 m and 120° with respect to the radar. Fig. 3.17 illustrates one specific measurement case from the dataset Set3_OnlyHumans.

3.4.5 Proposed Approach

Fig. 3.18 depicts the block diagram of the proposed method. It can be inferred from Fig. 3.18 how the proposed method works by utilizing raw IF signal data collected

Table 3.14: Configuration parameters for Yolov3 [13].

Parameter	**Value**
Number of classes	3
Batch Size	64
Learning rate	0.001
Momentum	0.9
Decay	0.0005
Max Batches	6000
Subdivision	16
Filters	24
CUDA version	11.4
GPU version	Tesla V100-SXM3

from the rotating radar system. To generate the range profiles, additional processing is performed onto the recorded IF samples. Every range profile corresponds to a specific angle at which the radar is positioned. A 180° range-angle view of the surrounding area is then obtained by stitching all the range profiles associated with each rotational step of the rotor. MATLAB is then used to retrieve the range-angle images from the collected range profiles. As described previously, the range-angle images signify the signatures of the three target objects, car, UAV, and humans.

The Yolov3 algorithm is then trained using the obtained range-angle images. We use the LabelImg software [152] to draw bounding boxes on the range-angle images so as to train the Yolov3 algorithm. The centroid of the bounding boxes drawn indicate the target object's AoA. Further, the different parameters utilized by the Yolov3 algorithm to identify the target objects from the range-angle maps are listed in Table 3.14. In this work, we use a 10-fold cross-validation scheme. This scheme involves training for 10 folds of the experiment. For training, 80% of the dataset images are utilized in each fold of the experiment. For testing, the remaining part of the data set are designated as test images. To initiate the training, we make use of the pretrained weights that are available from the Darknet-53 repository. The weights are updated for each fold of the training so as to reach the optimal values. Consequently, the 1^{st} fold's weights are uncorrelated to those of the 2^{nd} fold, and so on. We are implicitly eliminating any bias connected to the training by employing this method. Finally, to obtain the predictions, the updated weights after training are used on the test images.

The Yolov3 algorithm outputs range-angle images with bounding boxes on the detected object. It should be noted that each pixel in the range-angle image is associated to a particular range and angle of the target object. As a result, in the measurement setup, the centroid coordinates of the detected bounding box match the estimated range and AoA of the target object. After obtaining the AoA estimation, the algorithm's performance can be determined by calculating the root mean square error (RMSE) between the target's ground truth angle and the estimated

Table 3.15: Accuracy and RMSE values for 10 folds [13].

Sl No.	UAV (RMSE)	Car (RMSE)	Human (RMSE)	Accuracy (%)
1	1.4746°	1.3234°	0.9644°	99.3902
2	0.7869°	1.4576°	0.8635°	98.4939
3	1.3661°	1.0667°	1.0702°	97.9166
4	0.8700°	1.3807°	1.2312°	98.6111
5	0.9305°	1.2518°	1.1380°	98.7012
6	0.7274°	1.2540°	0.8898°	98.4423
7	0.8963°	1.6049°	1.1868°	96.8838
8	1.4128°	1.2237°	1.1005°	98.6413
9	1.3441°	1.2035°	0.9511°	95.5882
10	0.9922°	1.1559°	0.8297°	99.3569

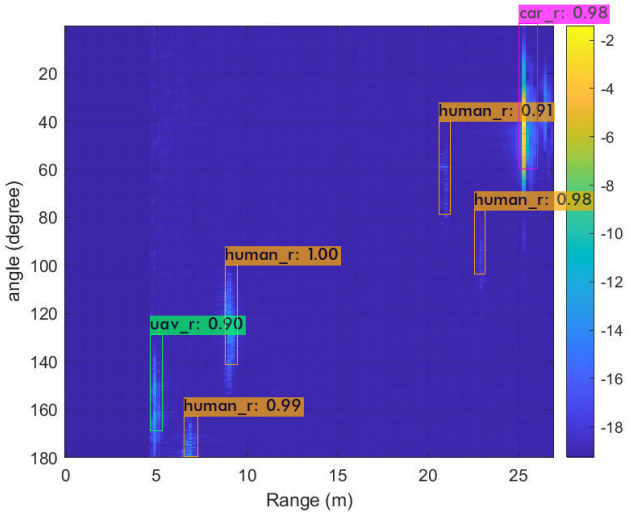
AoA value.

3.4.6 Results

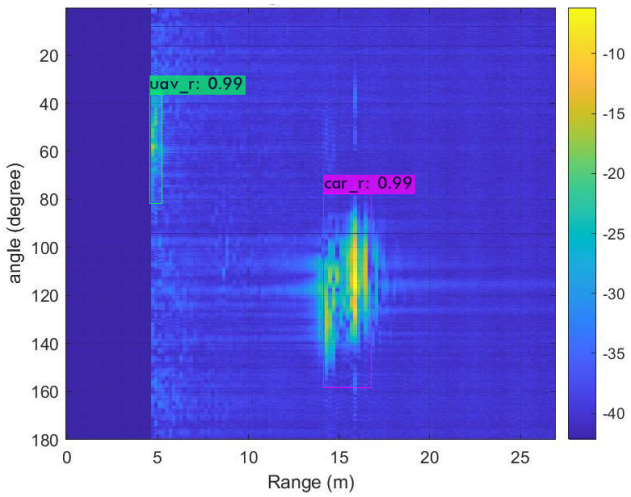
Table 3.16: Comparison of the proposed work with other methods [13].

Method	Number of antennas	Targets with same range or angle	Target classification
2D-ESPRIT [153]	1 Tx, 2 Rx	✗	✗
DFT-ESPRIT [154]	1 Tx, 2 Rx	✗	✗
Dual-Smoothing [155]	1 Tx, 2 Rx	✗	✗
Clustered ESPRIT [156]	1 Tx, 2 Rx or more. Rx antennas could be less than number of targets	✓	✗
Rotating Radar [25]	1 Tx, 1 Rx	✓	✗
Ours	**1 Tx, 1 Rx**	✓	✓

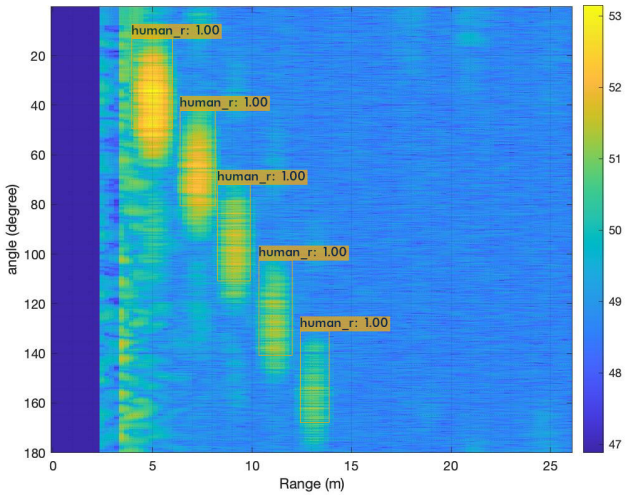
The Yolov3 algorithm is trained with the labeled range-angle images to determine the ideal weight parameters. The training is performed for 10 folds of the experiment containing 6000 iterations each so as to minimize the average loss. Fig. 3.20 shows the average loss vs iterations for the 7^{th} fold of the experiment. Additionally, Table 3.15 lists the accuracy for each experiment fold. Our approach is reliable, as evidenced by the comparatively high average classification accuracy of 98.20%, as observed from Table 3.15. As seen in Fig. 3.19, the classified output can be observed on the range-angle image. The confusion matrix shown in Fig. 3.21 also



(a) Set1_UavCarHumans



(b) Set2_UavCar



(c) Set3_OnlyHumans

Figure 3.19: Inference images after executing Yolov3 on all three range-angle image datasets [13].

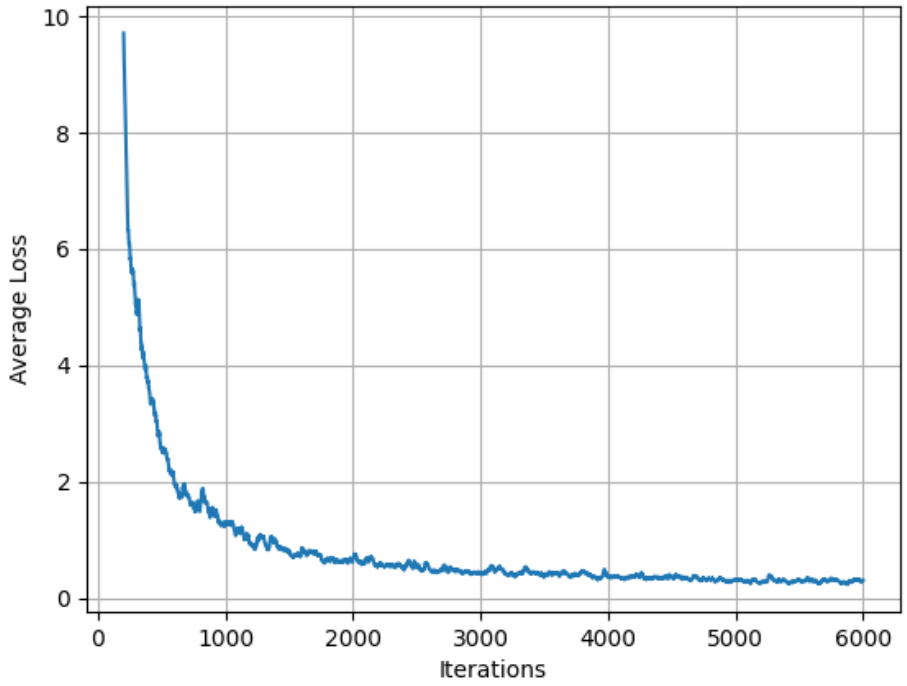


Figure 3.20: Plot depicting the average loss vs iteration for the 7^{th} fold [13].

Class	UAV	Car	Human	None	Total
UAV	650	0	0	14	664
Car	0	669	0	1	670
Human	0	0	1955	14	1969

Figure 3.21: Confusion matrix depicting the classification performance [13].

depicts the prediction statistics. According to the confusion matrix, the percentages of UAV, car, and humans are 97.89%, 99.85%, and 99.28%, respectively. It was also observed that the average precision, recall, and F1-score are 0.991, 0.992, and 0.991, respectively for the 10 folds of the experiment.

The RMSE value is used to determine the performance of the proposed AoA estimation approach. The various RMSE values for various test cases are listed in Table 3.15. As can be observed, the average AoA RMSE values for the humans, car, and UAV are 1.0225°, 1.2922°, and 1.0800° respectively. It is interesting to note that car has an average RMSE value higher than both humans and UAV. This is because the larger size of the car accounts for its greater AoA dispersion. Thus the proposed approach suggests that the large size of the car in relation to a UAV and humans makes it challenging to quantify its AoA to a single value.

The advantages of the proposed method over other similar techniques, including rotating radar [25], DFT-ESPRIT [154], dual-smoothing [155], clustered ESPRIT [156], and 2D-ESPRIT [153], are presented in Table 3.16. The proposed approach

is found to be comparable to the rotating radar technique in utilizing just a single transmitter-receiver antenna to identify a number of non-interacting targets. In addition to AoA estimate, the proposed approach also offers the benefit of target classification. Furthermore, the proposed approach exhibits a lower RMSE error when compared against the rotating radar method.

3.5 Summary

In this chapter, we delved into some of the inherent issues related to sensing in UAVs. Primarily, we focused on the detection of multiple UAVs or targets by utilizing different sensing technologies and machine learning techniques. In the first approach, we propose an end-to-end technique to estimate the UAV count by utilizing thermal sensors. The proposed method is deployed on embedded hardware so as to capture thermal images that are subsequently used to process and estimate the UAV count. Furthermore, a UAV-thermal dataset has been created, that comprises of a collection of 10 UAVs that are configured to traverse in various directions. Extensive experiments were conducted to illustrate a 99.9% accuracy for the proposed CNN architecture to estimate the number of UAVs present. In the future, our intention is to expand upon this research by directing our attention towards the identification of UAV models. This endeavor holds the potential to greatly influence the mitigation of UAV security risks within a multi-UAV setting.

In the second approach, acoustic sensors were utilized to precisely assess the UAV count in the surrounding area. A UAV acoustic dataset was created, consisting of 10 combinations of UAVs, covering both indoor and outdoor environments. The UAVs were configured to maneuver randomly in all directions. The dataset underwent preprocessing by utilizing time-frequency transformations so as to extract spectrogram images from the acoustic input. The spectrogram images that are produced are subsequently inputted into a specialized lightweight CNN model in order to approximate the UAV count. The proposed model demonstrates a high average test accuracy in effectively estimating the number of UAVs in the vicinity. Furthermore, the proposed model has been deployed on different edge computing platforms to assess its performance in terms of inference time. Potential future advancements of this research involve integrating additional sensors to enhance the UAV model identification and tracking.

In the third approach, we introduced a novel method for accurately estimating the AoA in mmWave FMCW radar. The proposed method aims to enhance the radar's ability to detect and locate targets in critical applications involving UAVs and GCS. In this work, the estimate of FoV and AoA of mmWave FMCW radars are enhanced by integrating a mechanical rotor configuration with advanced computer vision techniques like Yolov3. In each rotational step of the rotor, the radar generates range profiles. The range profiles are subsequently combined to provide range-angle images, which offer a comprehensive 180° perspective of the surrounding area. The range-angle images generated are subsequently inputted into the Yolov3

algorithm, which is responsible for the detection, classification, and localization of several targets inside the given scene. The proposed method demonstrated exceptional accuracy in target classification and surpassed previous techniques with its low AoA estimation error. A potential avenue for future research could involve assessing the real-time performance of the suggested approach on UAVs and GCS in order to mitigate delay in target detection and localization.

Chapter 4

Hybrid Communication Switching Scheme for UAV-Assisted Wireless Networks

4.1 Introduction

In this chapter, we discuss the communication aspects of a UAV-assisted wireless network. Specifically, we focus on the inherent issues of relying on a single communication technology to meet the various application use cases. In light of this, we introduce a novel hybrid communication switching scheme to incorporate various communication technologies and utilize them based on application requirements.

This chapter contains relevant content taken from Paper F [14] and Paper G [15]. The mentioned articles are provided in the Appendix section of this dissertation. Firstly, the chapter provides sufficient motivation for the notion of utilizing an adaptively switching communication framework. Afterward, we provide two approaches to form the energy-efficient data transmission hybrid network, one using the FS and FSMP path loss model and the other using RL algorithms. Additionally, we also present the derived analytical results for the formed hybrid network. The chapter concludes by providing a summary of the work and avenues for future research.

4.2 Motivation

BLE [101], Wi-Fi [102], LTE-M [103], and LoRa [35] are a few of the communication modules utilized in UAVs. For short distances (about 300 m), BLE can transfer data with low energy [36, 157]. As a result, BLE can be used for several low-power, short-range UAV applications, such as smart agriculture [38]. The authors of [101] have proposed an approach that uses UAVs to gather data from several sensor nodes spread across a sizable farmland area. The proposed approach utilizes BLE to send crop health data from the sensor nodes to the nearby UAV. For applications that demand high data rates, Wi-Fi is the preferred communication technology. In [102],

Figure 4.1: Pycom communication module.

a UAV system that can transmit voice over internet protocol (VoIP) data to the GCS in disaster-affected areas for monitoring purposes has been proposed. Wi-Fi is chosen for transmitting the video data due to its reliability and high-speed data rate. Conversely, LTE-M is an improved variant of the widely used LTE protocol that is intended to facilitate uninterrupted connection for IoT devices. In [103], it has been shown that LTE-M and other LPWAN protocols have demonstrated efficacy in facilitating dependable communication between high-speed moving entities. The LoRa communication protocol is frequently employed for long-distance communication purposes. The LoRa protocol offers a coverage range of approximately 10 km, accompanied by a relatively low power consumption of 0.025 watts [158, 33]. The authors in [159] have examined the dependability of several communication technologies, including LoRa, Wi-Fi, and LTE, from the standpoint of a UAV swarm. The LoRa protocol has been found to have improved dependability for long-range communications for a higher density of UAV swarms, in comparison to Wi-Fi. Therefore, this demonstrates that relying solely on a single communication protocol may not be optimal in meeting the requirements of an adaptive wireless network supported by UAVs. For instance, the LoRa protocol has the capability to facilitate long-distance communication. Nevertheless, the UAV's movement hinders its ability to provide higher data rates while it is in close proximity to the GCS. Opting for either BLE or Wi-Fi in this scenario could potentially result in increased data rates over the same distance. Driven by the aforementioned, the primary goal of this chapter is to address the aforementioned limitations associated with the utilization of a single communication technology for wireless networks assisted by UAVs.

Chip miniaturization and technological advances have resulted in reducing the form factor of existing technologies. For example, the Pycom communication module shown in Fig. 4.1 houses five communication technologies namely Sigfox, LoRa, Wi-Fi, Bluetooth, and LTE packed into a single chip. UAVs equipped with these chips can switch between these technologies to provide either low-energy data transmission or high throughput or low latency communication. Drawing inspiration from this aspect, our work proposes to provide a detailed analysis of a novel hybrid communication switching scheme employing such communication modules for UAV-assisted wireless networks.

4.3 Contributions

Based on the discussions above, the major contributions of this work as summarized in Paper F[14] and Paper G[15] are as given below:

- A hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network is formed with FS and FSMP energy models.
- Analytical expressions corresponding to network energy consumption and average network delay are derived for the data transmission over the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network.
- Further, two RL algorithms namely, Q-learning and DQN are proposed for energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. The performance of the proposed RL algorithms is compared with the rule-based algorithm and random hybrid scheme.
- Through extensive numerical results, it is shown that the proposed RL algorithms result in energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. We also compare the performance of the proposed hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network in terms of the network energy consumption and average network delay with conventional shortest path routing algorithm considering individual communication technology.

4.4 System Model

We examine a UAV-assisted wireless network with N UAVs randomly placed over a B^2 area. The UAVs are deployed over l layers of height $h_1, h_2, \cdots, h_l$ as illustrated in Fig. 4.2. UAVs gather data and transmit it to the GCS that is located at coordinates $(B/3, B/3)$. GCS is deemed to possess comprehensive access to various communication technologies for the purpose of data collection and processing. In this scenario, it is assumed that the data arrived at each UAV follows a Poisson distribution. Upon successful data transmission, the UAV proceeds to randomly relocate to another location in order to collect data.

The selection of a communication technology from BLE, LTE, Wi-Fi, and LoRa signifies the establishment of a UAV-GCS link. After the establishment the UAV-GCS link, the transmission of data will occur at a transmit power of P_T and a data rate of R. A delay of T_d seconds will be incurred. Moreover, this work examines two path loss models, namely FS and FSMP, in the context of establishing a communication technology for a UAV-GCS link. In the FS model, a UAV-GCS link has the ability to choose a communication technology if the radial distance of the link is smaller than r_τ, where τ denotes one of the communication technology from BLE, LTE, Wi-Fi, or LoRa. The energy consumption in the FS model adheres to d^2 model. In the FSMP model, however, a communication link is assigned to a

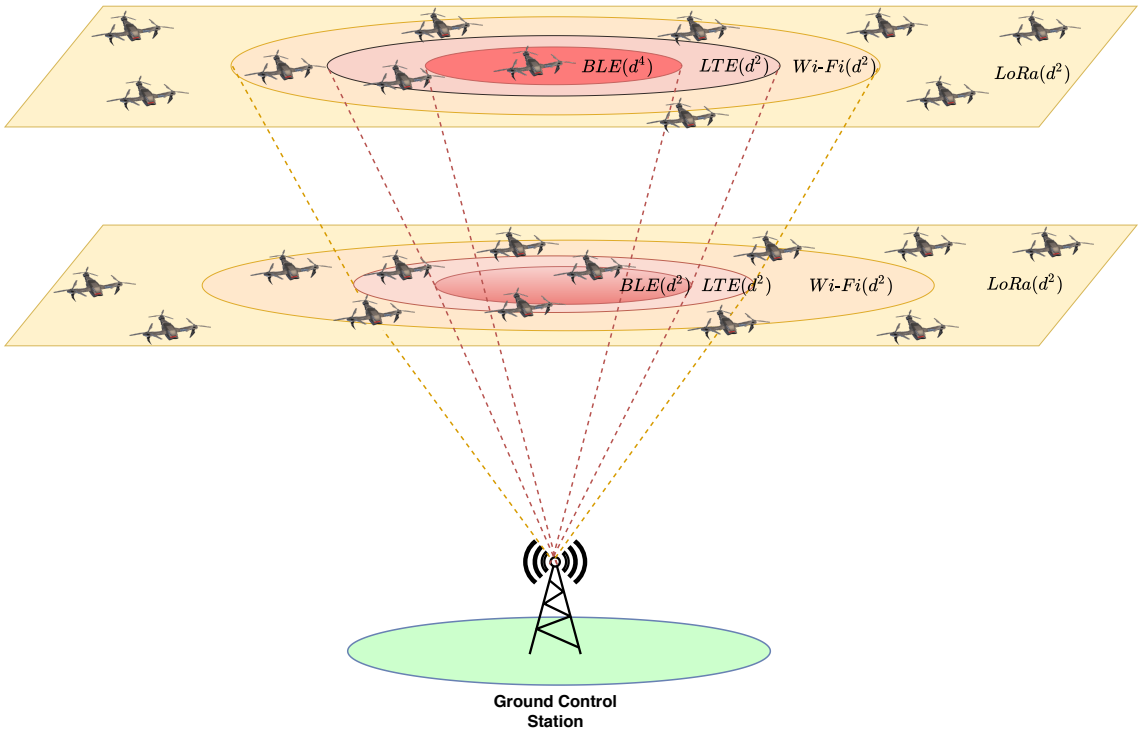


Figure 4.2: System model for the proposed hybrid BLE/LTE/Wi-Fi/LoRa scheme for two layers [15].

UAV-GCS link regardless of the distance. In the event that the radial distance is smaller than r_τ, the energy consumption adheres to the d^2 model. Conversely, if the radial distance exceeds r_τ, the energy consumption conforms to the d^4 model.

4.5 Problem Formulation

The primary goal of this work is to reduce the total energy consumption of the UAV-assisted wireless network. The overall energy consumption of the network is calculated by adding up the energy consumption at each link between the UAV and GCS. The energy expenditure at a UAV-GCS link is determined by the combined energy consumption of transmitting data using a transmit power and transmitting this data over a radial distance d_g. Assuming R as the data rate and P_T as the transmit power, the energy consumed for transmitting k bits of data can be determined as [160]

$$E_{Tr} = \frac{kP_T}{R}. \tag{4.1}$$

As previously stated, the energy consumption is contingent upon the path loss model and the radial distance. Next, the energy consumption for the FS model and FSMP model is derived in the subsequent sections.

4.5.1 FS Model

In the framework of the FS model, a UAV-GCS link selects a communication technology if the radial distance associated with that particular communication technology is smaller than the threshold distance r_τ. The energy consumed for the transmission of data over a radial distance d_g is determined as [161]

$$E_{d,fs} = k\mathcal{E}_e + k\mathcal{E}_{fs}d_g^2, \tag{4.2}$$

where $\mathcal{E}_{fs}$ is the free space power amplification energy and $\mathcal{E}_e$ represents the energy consumed by the electronic circuit for transmitting one bit.

4.5.2 FSMP Model

The FSMP model determines the assignment of a UAV-GCS link to any communication technology, regardless of its radial distance. The calculation of the energy required to transmit k bits of data over a radial distance d_g is derived as [161]

$$E_{d,fsmp} = \begin{cases} k\mathcal{E}_e + k\mathcal{E}_{fs}d_g^2, & \text{when } d_g < r_\tau\,; \\ k\mathcal{E}_e + k\mathcal{E}_{mp}d_g^4, & \text{when } d_g \geq r_\tau\,. \end{cases} \tag{4.3}$$

Here, the symbol $\mathcal{E}_{mp}$ represents the power amplification energy in the multipath fading model, which is calculated as $\mathcal{E}_{mp} = \mathcal{E}_{fs}/r_\tau^2$. For simulation purpose, we consider $\mathcal{E}_{fs} = 10 \times 10^{-12}$ J/bit/m^2 and $\mathcal{E}_e = 25 \times 10^{-9}$ J/bit.

Lastly, in terms of the energy consumption, the overall cost of the UAV-assisted wireless network is given as

$$C = \sum_{i=1}^{N}(E_{Tr,i} + E_{d,i}). \tag{4.4}$$

Thus, the main objective is to reduce the cost of the network in terms of energy consumption, as defined in (4.4).

4.6 Analytical Model for Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network with Energy-Efficient Data Transmission

This section presents the mathematical formulas that have been derived for the determination of network energy consumption in both the FS and FSMP models. For the interested reader, a detailed derivation is provided in Paper G section G.4.

4.6.1 FS Model

The FS model calculates the total energy consumed by the network $E_{N/W}$, as

$$E_{N/W} = \sum_{\tau} P_{pr,\tau} N(E_{Tr,\tau} + E_{d,fs,\tau,avg}) \tag{4.5}$$

where for the communication technology τ, $E_{Tr,\tau}$ can be computed from (4.1). In this context, $E_{d,fs,\tau,avg}$ denotes the mean energy consumption associated with the communication technology τ and it can be obtained as

$$E_{d,fs,\tau,avg} = k\mathcal{E}_e + \frac{k\mathcal{E}_{fs}}{2}(r_a^2 + r_b^2) \tag{4.6}$$

where r_a and r_b are defined as

$$r_a = \begin{cases} 0, & \text{for BLE} \\ r_{BLE,fs}, & \text{for LTE} \\ r_{LTE,fs}, & \text{for Wi-Fi} \\ r_{Wi\text{-}Fi,fs}, & \text{for LoRa} \end{cases} \tag{4.7}$$

and

$$r_b = \begin{cases} r_{BLE,fs}, & \text{for BLE} \\ r_{LTE,fs}, & \text{for LTE} \\ r_{Wi\text{-}Fi,fs}, & \text{for Wi-Fi} \\ r_{LoRa,fs}, & \text{for LoRa} \end{cases} \tag{4.8}$$

The average network delay, denoted as $T_{avg,fs}$, is derived in a similar manner as,

$$T_{avg,fs} = \frac{k}{R} + \frac{2(r_a^2 + r_b^2 + r_a r_b)}{3c(r_a + r_b)} \tag{4.9}$$

where r_a and r_b are as defined previously, R represents the rate of the communication technology, and k denotes the number of bits transmitted. The parameter c represents the velocity of light in vacuum (3×10^8 m/s).

4.6.2 FSMP Model

Unlike the FS model, the FSMP model calculates the total energy consumed by the network $E_{N/W}$ as,

$$E_{N/W} = \sum_{\tau} P_{pr,\tau} N (E_{Tr,\tau} + E_{d,fsmp,\tau,avg}) \tag{4.10}$$

where the term $E_{Tr,\tau}$ is computed by (4.1) for the communication technology τ. Here, the mean energy consumption by the communication technology τ is denoted by $E_{d,fsmp,\tau,avg}$ and it is obtained as,

$$E_{d,fsmp,\tau,avg} = k\mathcal{E}_e + \frac{k}{(r_b^2 - r_a^2)}\left[\frac{k\mathcal{E}_{fs}}{2}(r_c^4 - r_a^4) + \frac{k\mathcal{E}_{mp}}{3}(r_b^6 - r_c^6)\right] \tag{4.11}$$

where r_a, r_b, and r_c are defined as

$$r_a = \begin{cases} 0, & \text{for BLE} \\ r_{BLE,fsmp}, & \text{for LTE} \\ r_{LTE,fsmp}, & \text{for Wi-Fi} \\ r_{Wi\text{-}Fi,fsmp}, & \text{for LoRa} \end{cases} \tag{4.12}$$

$$r_b = \begin{cases} r_{BLE,fsmp}, & \text{for BLE} \\ r_{LTE,fsmp}, & \text{for LTE} \\ r_{Wi\text{-}Fi,fsmp}, & \text{for Wi-Fi} \\ r_{LoRa,fsmp}, & \text{for LoRa} \end{cases} \tag{4.13}$$

and

$$r_c = \begin{cases} r_{BLE,fsmp}, & \text{for BLE} \\ r_{LTE,fsmp}, & \text{for LTE} \\ r_{Wi\text{-}Fi,fsmp}, & \text{for Wi-Fi} \\ r_{LoRa,fsmp}, & \text{for LoRa} \end{cases} \tag{4.14}$$

In a similar manner, $T_{avg,fsmp}$ that represents the average network delay is computed as,

$$T_{avg,fsmp} = \frac{k}{R} + \frac{2(r_a^2 + r_b^2 + r_a r_b)}{3c(r_a + r_b)} \tag{4.15}$$

where r_a, r_b, and c are as previously defined, R is the rate of the communication technology, and k is the number of bits transmitted. Both the hybrid FS and FSMP methods have the same average network delay. This is anticipated as the delay in both scenarios is directly depending upon the variable d_g. Furthermore, this assertion is also substantiated by the simulations results provided in the following sections.

4.7 Proposed Approach I: Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network with Energy-Efficient Data Transmission using FS and FSMP Models

The objective of the proposed scheme is to establish a UAV-assisted wireless network that combines BLE, LTE, Wi-Fi, and LoRa to transmit data in an energy-efficient manner. Two path loss models are considered for forming the network, as explained below.

4.7.1 FS Model

With the FS model, a UAV-GCS link has the capability to choose a communication technology if the radial distance value is less than the predetermined threshold distance r_τ.

The steps for choosing a communication protocol for a UAV-GCS link is outlined in Algorithm 1. According to the data presented in Table 4.1, it can be noted that

Table 4.1: Key parameters for various communication protocols considered in this work [15].

Protocol	$\boldsymbol{P_T}$ **(W)**	$\boldsymbol{R}$ **(Mbps)**	$\boldsymbol{P_T/R}$	$\boldsymbol{r_\tau}$
Wi-Fi [40], [158]	2	10	0.2×10^{-6}	600
LTE [42], [162]	0.1	1	0.1×10^{-6}	400
BLE [163], [7]	0.01	1.36	0.007×10^{-6}	200
LoRa [158], [33], [35]	0.025	0.050	0.5×10^{-6}	1500

Algorithm 1: Algorithm to form the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network by utilizing the FS framework [15].

Input: N, d_g
Output: Network energy consumption and average network delay
if $d_g < r_{BLE}$ **then**
  Compute the energy consumption across a UAV-GCS link with (4.1) and (4.2) for all the communication protocols;
  $E_{\min} = \min\{E_{BLE}, E_{LTE}, E_{Wi\text{-}Fi}, E_{LoRa}\}$;
  Select the communication technology with cost $E_{\min}$;
else if $d_g < r_{LTE}$ **then**
  Compute the energy consumption across a UAV-GCS link with (4.1) and (4.2) for Wi-Fi, LTE, and LoRa;
  $E_{\min} = \min\{E_{LTE}, E_{Wi\text{-}Fi}, E_{LoRa}\}$;
  Select the communication technology with cost $E_{\min}$;
else if $d_g < r_{Wi\text{-}Fi}$ **then**
  Compute the energy consumption across a UAV-GCS link with (4.1) and (4.2) for LoRa and Wi-Fi;
  $E_{\min} = \min\{E_{Wi\text{-}Fi}, E_{LoRa}\}$;
  Select the communication technology with cost $E_{\min}$;
else
  Select the communication protocol as LoRa;
end
Compute the overall energy consumption and average delay of the network;

the threshold distance of BLE is lower than that of LTE. Similarly, the threshold distance of LTE is lower than Wi-Fi which is subsequently lower than LoRa. Initially, if the radial distance of a particular link is lower than the prescribed r_{BLE}, then the link is assigned to any one of the communication technologies based on minimum energy consumption. In cases where the radial distance is below the threshold value r_{LTE}, the link will be allocated to one of the communication technologies, namely LTE, Wi-Fi, or LoRa, based on the principle of minimizing energy consumption. Similarly, if the radial distance is lower than $r_{Wi\text{-}Fi}$, the link is allocated to either Wi-Fi or LoRa based on the same energy criteria. Alternatively, it will be linked to LoRa.

Algorithm 2: Algorithm to form the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network by utilizing the FSMP framework [15].

Input: N, d_g

Output: Network energy consumption and average network delay

1 Compute the energy consumption across a UAV-GCS link with (4.1) and (4.3) for all the communication protocols;

2 $E_{min} = \min\{E_{BLE}, E_{LTE}, E_{Wi\text{-}Fi}, E_{LoRa}\}$;

3 Select the communication technology with cost $E_{\min}$;

4 Compute the overall energy consumption and average delay of the network;

4.7.2 FSMP Model

According to the FSMP model, the assignment of a communication technology is determined by its energy consumption, regardless of its radial distance.

Algorithm 2 outlines the steps to form the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network that enables energy-efficient data transmission. The algorithm works on the principle that the communication technology with lower energy consumption for transmission is assigned a UAV-GCS link.

4.8 Proposed Approach II: Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network with Energy-Efficient Data Transmission using RL Algorithms

This section presents Q-learning and DQN, two reinforcement learning algorithms for forming an energy-efficient data transmission over a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network.

4.8.1 Q-Learning

In this subsection, the proposed Q-learning method for transmitting data in an energy-efficient manner via a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network is presented. Q-learning is a reinforcement learning technique that operates without a model, wherein the agent engages in direct interaction with the environment in order to obtain the optimal policy. Q-learning is value-based and is capable of determining the value function without requiring any prior information of the state transition probabilities [164]. The execution time and storage cost of Q-learning are comparatively lower than those of model-based algorithms, as long as the state and action space are not extensive. The performance and conduct of the algorithm are dependent on the reward, hence appropriate design of the reward function is essential [165], [166]. The objective of this study is to facilitate the transmission of data from UAV to a GCS while minimizing energy usage. An illus-

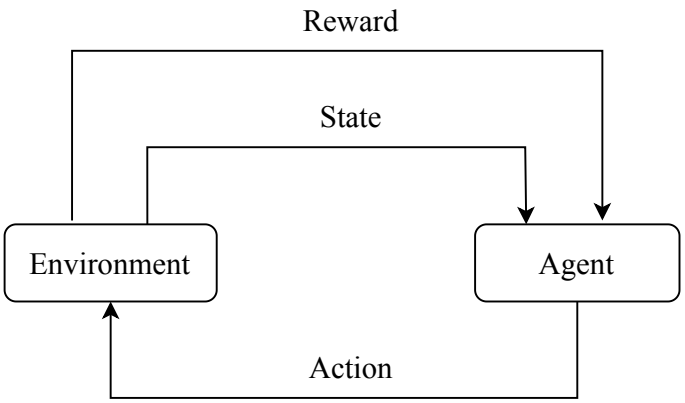


Figure 4.3: Block diagram illustration of Q-learning [15].

Table 4.2: Q-table representing the state-action pairs formulated for Q-learning [15].

States \ Actions	A_1	A_2	$\cdots$	A_M
S_1	$Q(S_1, A_1)$	$Q(S_1, A_2)$	$\cdots$	$Q(S_1, A_M)$
S_2	$Q(S_2, A_1)$	$Q(S_2, A_2)$	$\cdots$	$Q(S_2, A_M)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_N	$Q(S_N, A_1)$	$Q(S_N, A_2)$	$\cdots$	$Q(S_N, A_M)$

tration of the states, actions, and reward for the Q-learning algorithm is provided in Fig. 4.3. In our hybrid system, we define the states, actions, and reward as given below:

States

Consider a collection of N states for the Q-learning algorithm, denoted as $S = \{s_n\}$, where $n = 1, 2, ..., N$. Each state in this work symbolizes the communication link established between the UAV and GCS. Given the presence of N UAVs in the network, the number of UAV-GCS links is N, resulting in a corresponding number of states of N.

Action

An action signifies the agent's selection of a communication technology to allocate to a UAV-GCS link. In our hybrid system, let there be a set of M actions $A = \{a_m\}$ that are associated with each state $s_n \in S$. Hence, in this work, as there are four communication technologies, the number of actions are four given by $M = 4$.

Reward/Penalty

The agent is rewarded or punished by the environment according on their actions. Let γ represent the reward or penalty. In this case, $\gamma = \gamma(s_n, a_m)$ represents the reward or penalty that an agent receives when it is in state s_n and carries out an action a_m. The reward/penalty will be allocated to the agent in our hybrid system subsequent to the selection of the communication technology for the UAV-GCS link. The proposed work aims to achieve energy-efficient data transmission via a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network, and hence γ is defined in relation to the network energy consumption. Thus, γ is calculated as,

$$\gamma = \begin{cases} \dfrac{10^{-}6}{E_{N/W}} & (4.16) \\ \dfrac{1}{E_{N/W}} & (4.17) \\ e^{-E_{N/W}} & (4.18) \\ \dfrac{W_1}{E_{1,N/W}} + \dfrac{W_2}{E_{2,N/W}}, & (4.19) \end{cases}$$

where, $E_{1,N/W}$ denotes the network energy consumption to transmit k-bits of information and $E_{2,N/W}$ represents network energy consumption for transmitting over a distance. Moreover, the weights allocated for $E_{1,N/W}$ and $E_{2,N/W}$ are denoted as W_1 and W_2, respectively. As the reward increases the network formed is more energy-efficient, hence we define γ with an inverse relation to the network energy.

Updating the Q-value

The agent utilizes the reward or penalty to modify the Q-value associated with a state-action pair, as specified in Table 4.2. A reward or penalty is obtained when a UAV-GCS link is associated to a communication protocol. The updated Q-value is given as,

$$Q(s_t, a_t) \leftarrow (1 - \lambda)Q(s_t, a_t) + \lambda(\gamma + \Delta \max_a Q(s_{t+1}, a)), \tag{4.20}$$

where, Δ represents the discount parameter and λ denotes the learning rate of the Q-learning algorithm. In this context, the term $\max_a Q(s_{t+1}, a)$ calculates the maximum future reward and subsequently applies it to the reward associated with the current state.

Algorithm 3: Algorithm to form the hybrid UAV-assisted wireless network by utilizing the Q-learning algorithm [15].

Input: λ, Δ, N UAVs, M communication technologies, exploration-exploitation factor ε, maximum episode count K_{max}

Output: Network energy consumption and average network delay

Assign zeros to the rows and columns of the Q-matrix;
Arrange the UAV-GCS links along the rows of the Q-matrix according to the descending order of distance to the GCS;
Arrange the communication technologies along the columns of the Q-matrix;
Training
for $i = 1$ **to** K_{max} **do**
    Choose the initial state s_0;
    for $t = 1$ **to** N **do**
        Generate random number $u \in (0, 1)$;
        if $u > \varepsilon$ **then**
            Choose the action that corresponds to maximum Q-value from the Q-matrix $a_t = \max_a Q(s_t, a_t)$;
        else
            Choose a communication technology at random as action a_t;
        end
        Compute the energy consumption of the network using (4.1), (4.3), and (4.4);
        Compute γ using (4.16), (4.17), (4.18), and (4.19);
        Modify the Q-value using (4.20);
        if $s_t == s_N$ **then**
            break;
        else
            Change s_t as $s_t = s_{t+1}$ to transition to the next state in the order of decreasing distance to GCS;
        end
    end
end
Validation
Select the maximum Q-value indices for each row in the Q-matrix;
Assign each UAV-GCS link to the communication protocol with maximum Q-value;
Compute the overall energy consumption and average delay of the network;

The Q-learning approach proposed in this work is described by Algorithm 3. The proposed approach involves initializing the Q-matrix by assigning the number of UAV-GCS links to the rows and the communication protocols to the columns. The UAV-GCS links are arranged in a descending order based on their distance from the GCS. Commencing with the initial row of the Q-matrix, select a communica-

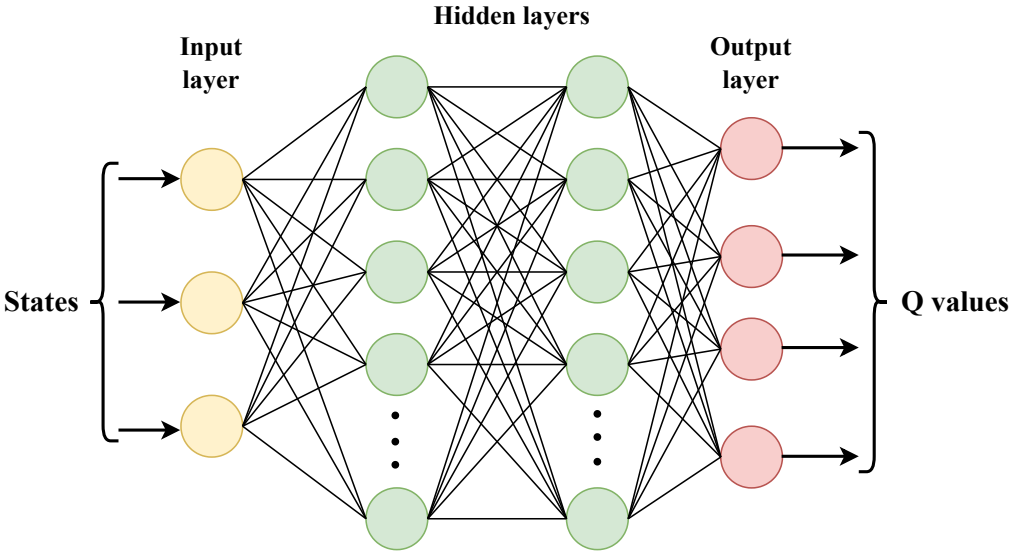


Figure 4.4: Neural network employed in the proposed DQN algorithm [15].

tion protocol at random. Compute the energy consumption of the network for the selected communication protocol. Compute the reward by utilizing (4.16), (4.17), (4.18), and (4.19). Subsequently, the Q-value of the link can be updated by employing (4.20). Afterward, proceed to the next state in accordance with decreasing distance to the GCS. The state transition rule in this work is partially influenced by the work done in [167], wherein the authors employed a similar strategy of transitioning to the next state based on decreasing channel state information. Continue the above steps until the UAV-GCS link that is nearest to the GCS is updated. The aforementioned steps are iterated several times so as to ensure that all communication protocols and links are covered. After the training is completed, establish a connection between the UAV-GCS link and the communication protocol that yields the maximum Q-value. Compute the overall network energy consumption and average delay of the network for achieving energy-efficient data transmission over the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. As the number of state-action pairs increases, the memory requirement for the storage of the Q-values in the Q-matrix also increases. As a result, there is a increased memory overhead that can adversely impact the algorithm performance [167]. In order to address these concerns, the DQN algorithm employs a neural network to approximate the Q-values [168]. The DQN algorithm utilizes the neural network to maintain the relative importance between the Q-values, resulting in comparable outcomes to that of the Q-learning algorithm [169]. Next, we dive into the specifics of the proposed DQN algorithm.

4.8.2 Deep Q-Network (DQN)

In this subsection, we introduce the DQN algorithm to form the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network so as to achieve energy-efficient data transmission.

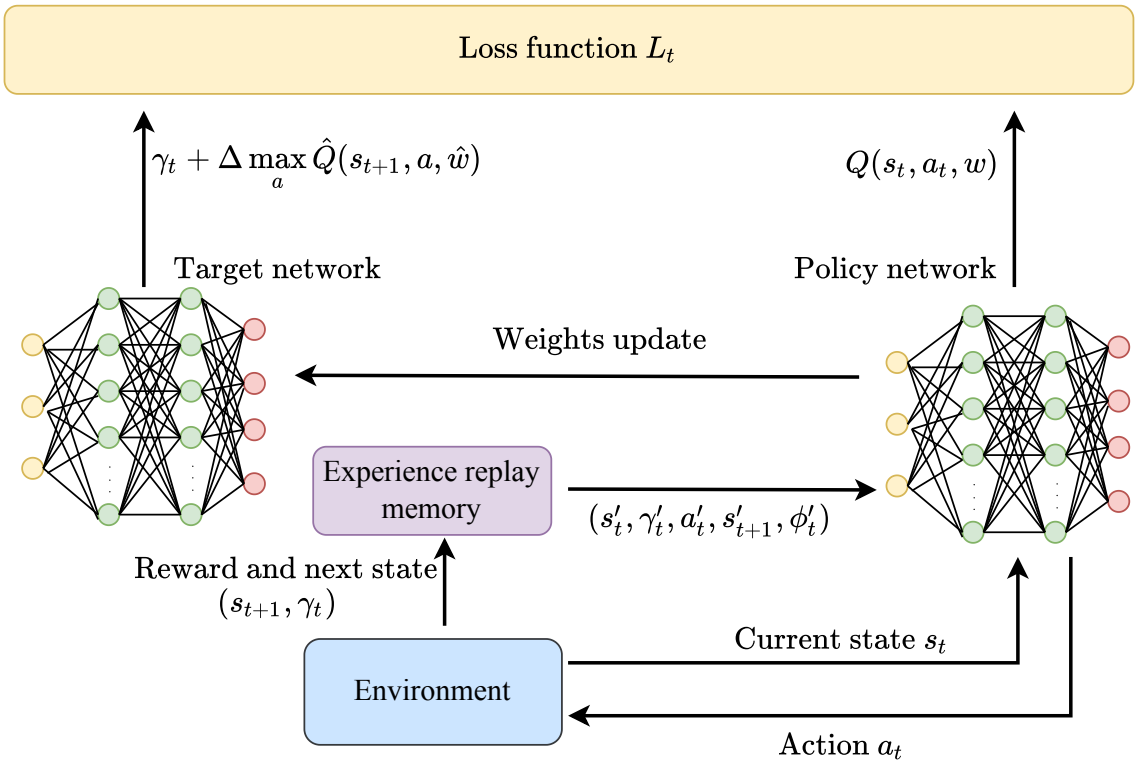


Figure 4.5: Schematic representation of the proposed DQN algorithm [15].

The application of the DQN algorithm to this work is similar to that of Q-learning. The descriptions for states and actions are identical to those defined for Q-learning. Thus, if there are N UAVs in the UAV-assisted wireless network, there will be N UAV-GCS communication links and N number of states. As the number of communication technologies are four, there are correspondingly four actions. The DQN algorithm employs a ε-greedy strategy to choose an action a_t for a given state s_t. This implies that a stochastic action is chosen (exploration) with a probability ε, and the action associated with the highest value $Q(s_t, a_t)$ is chosen with a probability $(1 - \varepsilon)$ (exploitation). Therefore, a tradeoff between exploration and exploitation is provided so as to ensure that the DQN algorithm may effectively train without being trapped in a local optima [167]. For a state s_t and an action a_t, the reward γ_t at step t is computed using (4.17).

The DQN network consist of two distinct neural network models, namely a policy network and a target network. The neural networks receive each UAV-GCS link together with its matching length as input. The policy network is formed by a fully connected neural network with two hidden layers, as depicted in Fig. 4.4. A collection of 256 neurons form the first hidden layer, whereas the number of neurons in the second hidden layer is 128. During the training process, the ReLu activation function is employed to activate all the layers. The policy network is duplicated to form the target network. The architecture of the target network is identical to that of the policy network. The primary function of the target network is to ensure stability during the training process of the DQN algorithm. The policy network's

output layer comprises 4 outputs that are activated linearly. At step t, the output from the DQN algorithm provides the prediction of the state-action values $Q(s_t, a_t)$.

The action a_t for state s_t is chosen according to the ε-greedy policy, as depicted in Fig. 4.5. Selecting action a_t leads to a change in state from s_t to state s_{t+1} and a reward γ_t. The transition between states adheres to the identical distance-based principle employed in Q-learning. The deque is used to store the tuple $K = \{s_t, \gamma_t, a_t, s_{t+1}, \phi_t\}$ in the experience replay memory. In this context, the variable ϕ_t represents the final state. In the subsequent step, a random mini-batch sample $(s'_t, \gamma'_t, a'_t, s'_{t+1}, \phi'_t)$ is chosen for the purpose of training the policy network. The primary objective of employing a random sample is to mitigate the influence of inter-state correlations and enhance the stability of the training procedure. The loss function is minimized during the training process. At any step t, the loss function is derived as [167]

$$L_t = E\left[\left(\gamma_t + \Delta \max_a \hat{Q}(s_{t+1}, a, \hat{w}) - Q(s_t, a_t, w)\right)^2\right] \tag{4.21}$$

where $Q(s_t, a_t, w)$ denotes predicted Q-value from the policy network when trained with weights w, Δ represents the discount factor ($\Delta \in [0, 1]$), and $(\gamma_t + \Delta \max_a \hat{Q}(s_{t+1}, a, \hat{w})$ signifies the output that is obtained from the target network once it is trained with $\hat{w}$ weights. It is important to note that the weights of the target network $\hat{w}$ are replicated from the policy network at each ζ iteration. The Adam optimizer is used to train both the policy network and target network. The policy network is updated by employing the equation provided in [167]

$$Q^*(s_t, a_t) \leftarrow Q(s_t, a_t) + \lambda(\gamma_t + \Delta \max_a \hat{Q}(s_{t+1}, a) - Q(s_t, a_t)) \tag{4.22}$$

where $Q^*(s_t, a_t)$ denotes the updated Q-value and λ represents the learning rate. Algorithm 4 describes the steps for the proposed DQN algorithm.

4.9 Numerical Results

In this section, we initially introduce the simulation configuration that was considered for producing the simulation results. Next, the simulation results for both the FS and FSMP models are shown in order to validate the analytical derivations. Finally, the simulation results are showcased to assess the performance of the proposed RL algorithms in comparison to other models.

4.9.1 Simulation Setup

For simulation, we consider a rectangular area measuring $1500 \times 1500\ m^2$, with the GCS positioned at coordinates $(500, 500, 0)$. There are 500 UAVs deployed in the rectangular area, at different altitudes of 100, 200, 300, and 400 meters. Every UAV is outfitted with four distinct communication technologies, including BLE, LTE, Wi-Fi, and LoRa. The rate at which packets arrive follows a Poisson distribution,

Algorithm 4: Algorithm to form the hybrid UAV-assisted wireless network by utilizing the DQN algorithm [15].

Input: λ, Δ, ζ, ε, N UAVs, M communication technologies, Number of episodes K_{max}

Output: Network energy consumption and average network delay

Initialize the policy network weights w, target network weights $\hat{w}$, and experience replay memory;

Sort the UAV-GCS links (states) in descending order based on their distance from the GCS;

Training

for $i = 1$ **to** K_{max} **do**

  Choose the initial state s_0 (first UAV-GCS link) from the sorted list;

  $\phi_0 = 0$, $j = 0$;

  for $t = 1$ **to** N **do**

    Generate a random number $u \in (0, 1)$;

    if $u > \varepsilon$ **then**

      Choose action $a_t = \max_a Q(s_t, a_t, w)$;

    else

      Choose an action a_t at random;

    end

    Compute the overall energy consumption of the network using (4.1), (4.3), and (4.4);

    Compute the reward γ_t, using (4.17);

    Transition to the next state s_{t+1} in order of decreasing distance to GCS;

    if $s_{t+1} == s_N$ **then**

      $\phi_t = 1$, $j = j + 1$;

    end

    Modify the experience replay memory with $K = (s_t, \gamma_t, a_t, s_{t+1}, \phi_t)$;

    if $\phi_t = 1$ **then**

      break;

    end

    Choose a minibatch sample $(s'_t, \gamma'_t, a'_t, s'_{t+1}, \phi'_t)$ at random, when the experience replay memory is full;

    Estimate the value of $Q(s_t, a_t, w)$ from the policy network;

    From the target network, estimate $(\gamma_t + \Delta \max_a \hat{Q}(s_{t+1}, a, \hat{w})$;

    Using (4.21), calculate the loss;

    Using (4.22), modify the weights w of the policy network;

    if $j == \zeta$ **then**

      Modify the weights $\hat{w}$ of the target network;

    end

  end

  Reduce the value of ε by the defined decay rate;

end

Validation
Retrieve the Q-values from the trained policy network for each UAV;
Assign each UAV-GCS link with the communication technology that exhibits the maximum Q-value;
Compute the overall energy consumption and average delay of the network;

Table 4.3: Simulation parameters employed for the assessment of the performance of the proposed approach [15].

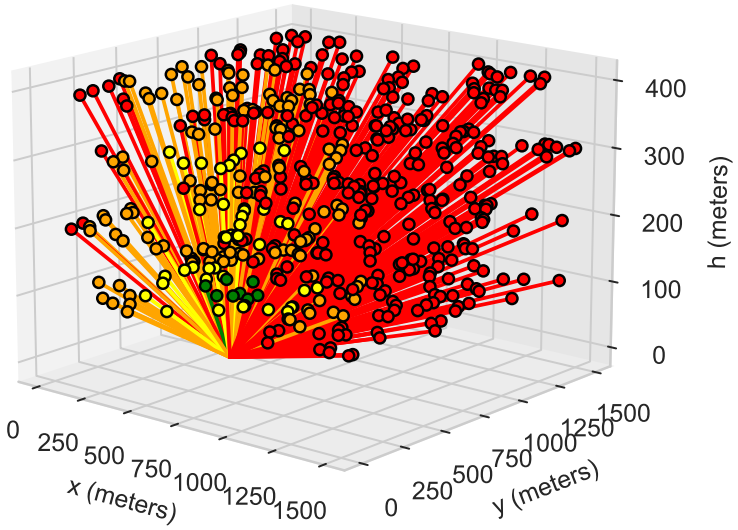
Parameter	**Value**
Number of UAVs, N	500
Location of GCS	$(500, 500, 0)$
Horizontal area	1500×1500
Hovering heights	$\{100, 200, 300, 400\}$
$\mathcal{E}_e$	25 nJ/bit
$\mathcal{E}_{fs}$	10 pJ/bit/m^2
Velocity of UAV, ν	2 m/s
Packet length, k	128 bits
Mean of Poisson distribution, λ	100
Speed of the transmission, c	3×10^8 m/s
Initial energy at a UAV E	10 Joules

with the mean being randomly selected from the set $\{1, 2, \cdots, 100\}$. Each packet consists of 128 bits, resulting in a random distribution of the total number of bits received by each UAV. Table 4.3 provides a detailed list of the parameters employed in the different runs of the simulation. Furthermore, Table 4.1 also includes the parameters for each communication technology.

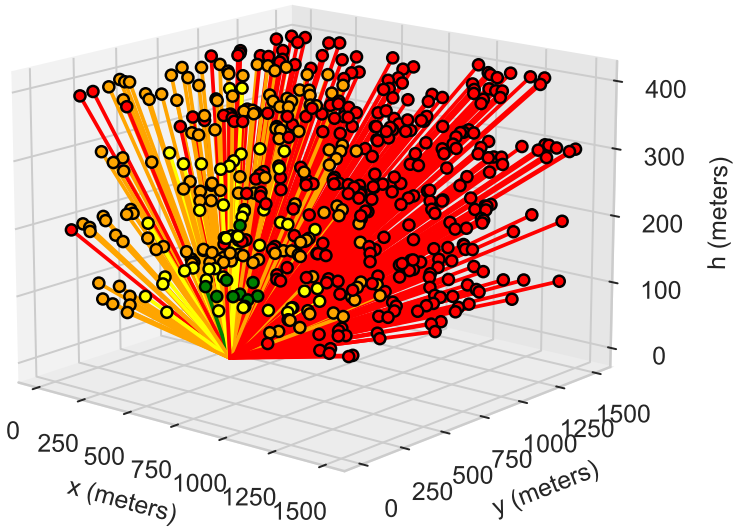
4.9.2 Simulation Results (FS and FSMP)

Based on the simulation parameters described above, we will discuss the obtained numerical results in this subsection.

The topology of the proposed hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network based on the FS energy model is as illustrated in Fig. 4.6a. Fig. 4.6a indicates that a significant proportion of the UAVs employ the LoRa protocol (shown in red) for establishing connections with the GCS. The UAVs situated a medium range employ Wi-Fi (yellow) and LTE (orange) technologies to establish the UAV-GCS connection. UAVs in closer proximity to the GCS are linked through the BLE (green) communication protocol. The BLE communication protocol consumes least amount of energy compared to other protocols when UAVs are in close proximity to the GCS. Therefore, a limited number of UAVs linked to BLE can be observed in close proximity to the GCS as shown in Fig. 4.6a. In the case of UAVs located at distances exceeding the path loss range of BLE, the communication protocol that

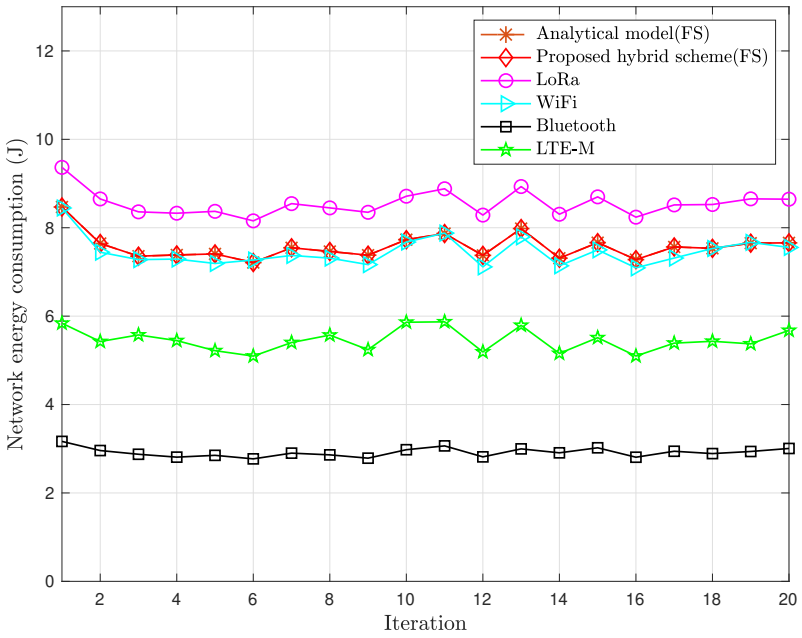


(a)

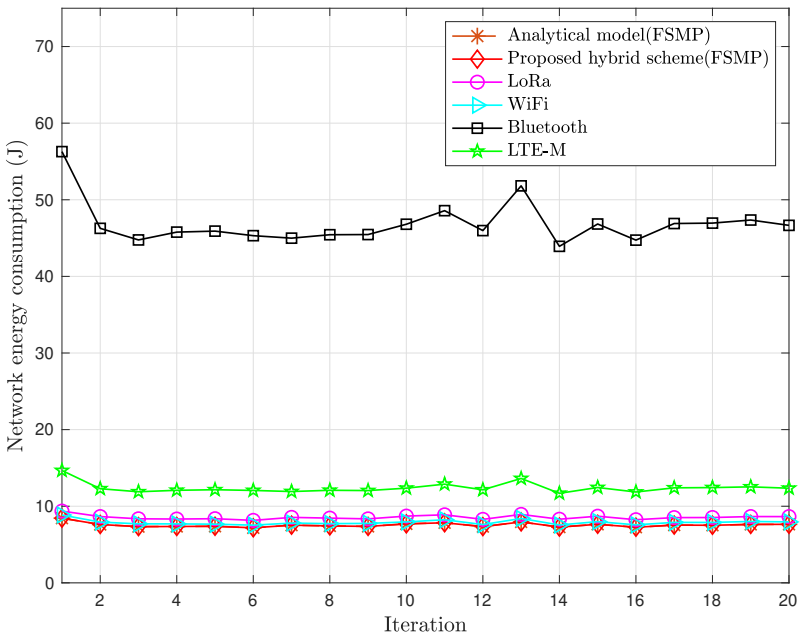


(b)

Figure 4.6: Topological representation of the UAV-assisted wireless network for (a) FS and (b) FSMP configurations. For all topologies, the GCS is situated at coordinates $(500, 500, 0)$. The communication technologies BLE, LTE, Wi-Fi, and LoRa are represented by the colors green, yellow, orange, and red, respectively [15].



(a)



(b)

Figure 4.7: Change in the energy consumption of the network for the models (a) FS and (b) FSMP for all random UAV settings. The curves for the analytical results are plotted using equations (4.5) and (4.10). For all configurations, the GCS is situated at coordinates $(500, 500, 0)$ [15].

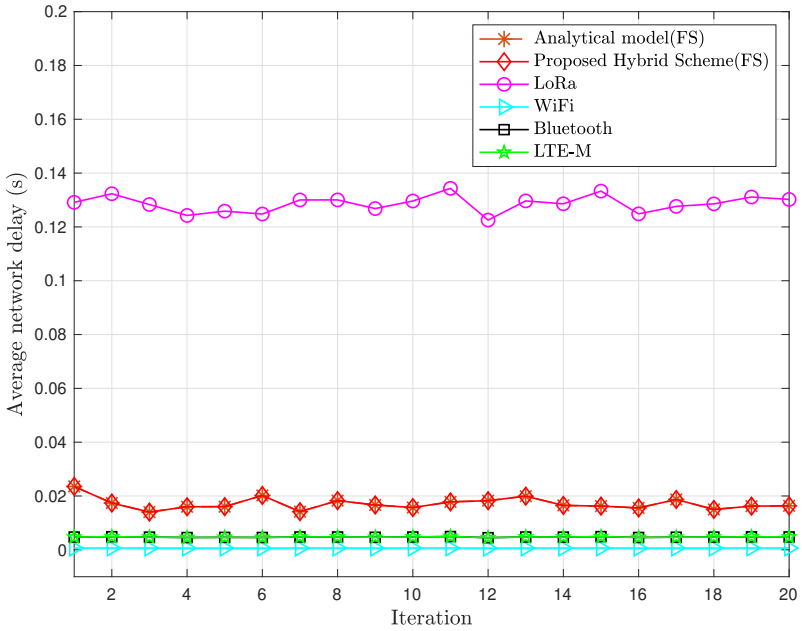
minimizes the energy consumption is selected. In the FSMP energy model, when the proposed scheme is utilized for the UAV-assisted wireless network, the link connections between the UAV and GCS are interchanged, as depicted in Fig. 4.6b. Fig. 4.6b demonstrates that a limited number of UAVs transition their communication protocol from LTE to BLE. This phenomenon can be attributed to the disparity in energy consumption between BLE, which adheres to the d^4 energy model, and LTE,

which adheres to the d^2 energy model. It can be observed that other communication protocols also exhibit similar changes.

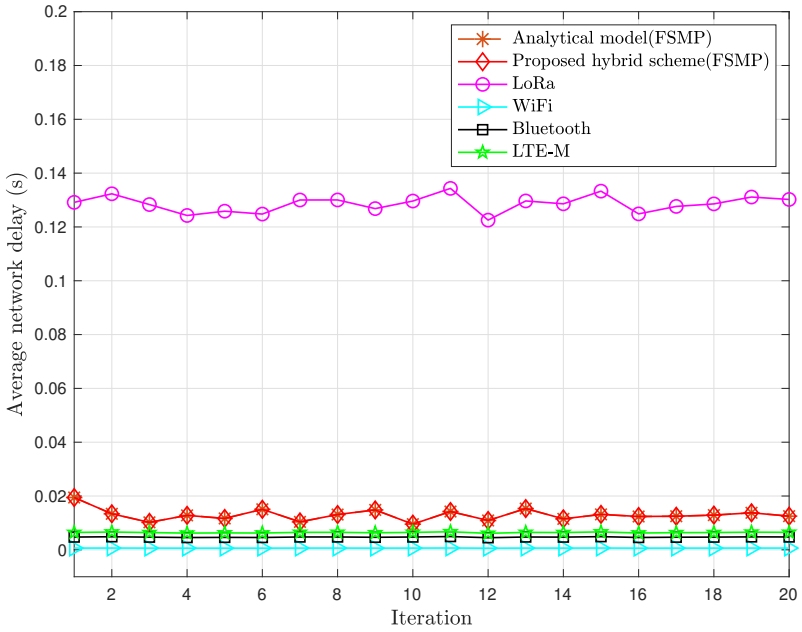
The network energy consumption for the proposed hybrid solution is compared to the standard shortest path routing algorithm with individual communication technology, taking into account the FS path loss model, as shown in Fig. 4.7a. The energy consumption of the proposed hybrid approach for the Wi-Fi protocol is comparable to that of the conventional approach, as depicted in Fig. 4.7a. Nevertheless, the energy consumed is still larger than that consumed by BLE. This can be attributed to the comparatively lower transmit power of BLE in relation to the average transmit power of the proposed hybrid scheme. For individual communication technology, the conventional approach employs the hop-based shortest path algorithm to transmit packets from the UAV to the GCS. The analysis of Fig. 4.7a reveals that the energy consumption obtained from the analytical model aligns as expected with the energy consumption observed in the simulation results. The hybrid approach is also compared to the standard shortest path routing algorithm with individual communication technology in terms of network energy consumption, as shown in Fig. 4.7b. In this case, we utilize the FSMP path loss model. Fig. 4.7b also demonstrates that the energy consumption associated with the proposed hybrid scheme is comparatively lower than that of the conventional approach. Furthermore, it has been shown that the conventional approach employing BLE consumes a greater amount of energy in comparison to alternative methods. The reason for this is that the value of r_τ is significantly low for BLE, leading to a greater number of UAV-GCS links adhering to the d^4 model. Additionally, the changes in the analytical results of the proposed approach is illustrated in Fig. 4.7b. The similarity between the analytical results and the simulation results, as depicted in Fig. 4.7b, serves to validate the accuracy of the analytical derivations.

The plot of the average network delay between the proposed hybrid scheme and the conventional shortest path routing algorithm for individual communication technology is provided in Fig. 4.8a and Fig. 4.8b. From Fig. 4.8a and Fig. 4.8b, it can inferred that there is no substantial variation in the delay for the proposed hybrid approach even when the proposed scheme adheres to both the FS and FSMP energy models. This phenomenon can be ascribed to the delay parameter that is predominantly influenced by the value of the transmission delay. The transmission delay is in turn dependent upon the data rate of the communication protocol. Thus the LoRa protocol has a higher delay, as it has comparatively lower data rate in relation to other communication protocols. Based on the analysis of Fig. 4.7b and Fig. 4.8b, it can be concluded that the proposed hybrid scheme, which follows the FSMP energy model, outperforms the conventional shortest path routing algorithm with individual communication technology in terms of minimizing energy consumption and reducing average network delay.

Fig. 4.9a illustrates the variation in network energy consumption for the hybrid scheme proposed, based on the FS and FSMP energy models. The network energy consumption between the FS and FSMP energy models in the proposed hybrid scheme is found to be similar as evident from Fig. 4.9a. Fig. 4.9b illustrates the



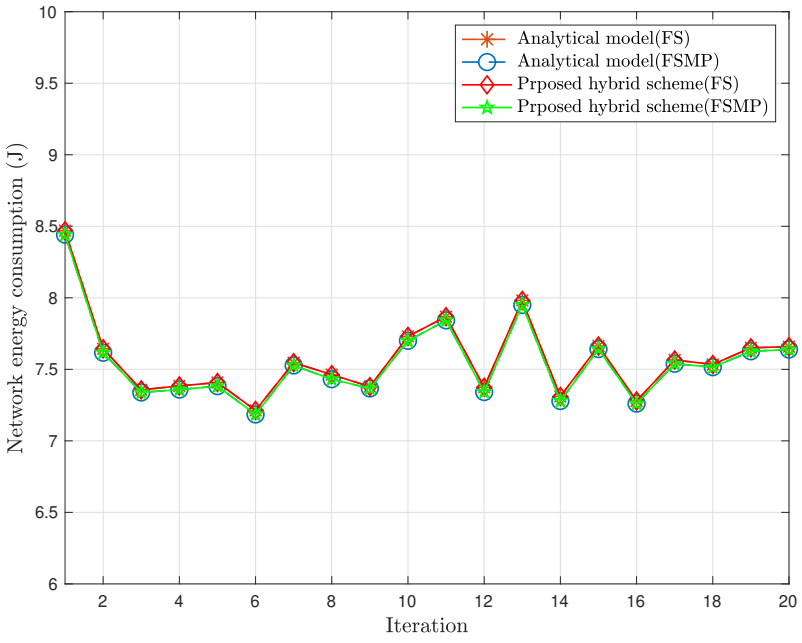
(a)



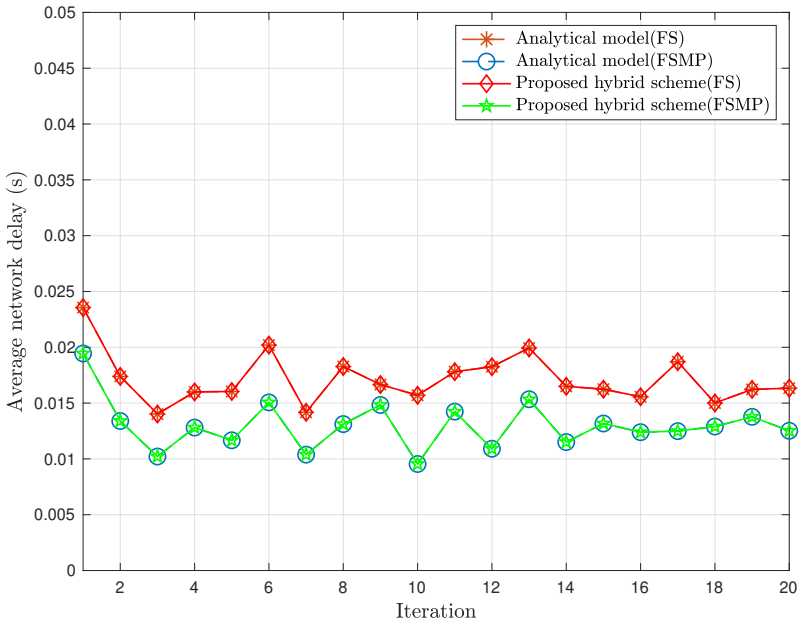
(b)

Figure 4.8: Change in the average delay of the network when based on (a) FS and (b) FSMP models for all random UAV settings [15]. The curves representing the analytical results are plotted by utilizing equations (4.9) and (4.15). In all configurations, the GCS is situated at coordinates $(500, 500, 0)$.

mean network delay for the proposed hybrid method, based on the FS and FSMP energy models. Fig. 4.9b reveals that the FSMP energy model demonstrates a reduced delay in comparison to the FS energy model of the proposed hybrid scheme. This can be better explained through the examination of the distinct interconnections between each communication technology within both schemes. The distribution of UAVs connected to various communication technologies in the proposed hybrid



(a)



(b)

Figure 4.9: Plots representing the change in (a) energy consumption and (b) average delay for the proposed hybrid network [15]. The curves representing the analytical results are plotted by utilizing the equations (4.5), (4.9), (4.10), and (4.15). For all settings, the GCS is situated at coordinates $(500, 500, 0)$.

scheme, based on the FS energy model, is as follows: 14 UAVs are connected to BLE, 153 UAVs are connected to LTE, 279 UAVs are connected to Wi-Fi, and the remaining 54 UAVs are connected to LoRa. The utilization of the FSMP energy model in the proposed hybrid technique results in a modification of this distribution. There is a change in the connection density, with 162 UAVs connected to LTE and 21 UAVs connected to BLE. Furthermore, the number of UAVs linked to LoRa decreases to

Table 4.4: Change in the energy consumption of the network when expressed as a weighted sum of energy components $E_{1,N/W}$ and $E_{2,N/W}$. The variables W_1 and W_2 denote the assigned weights [15].

W_1	W_2	E_{total} (J)
0.1	0.9	0.0204
0.2	0.8	0.0287
0.3	0.7	0.0392
0.4	0.6	0.0512
0.5	0.5	0.0650
0.6	0.4	0.0797
0.7	0.3	0.1002
0.8	0.2	0.1430
0.9	0.1	0.2589

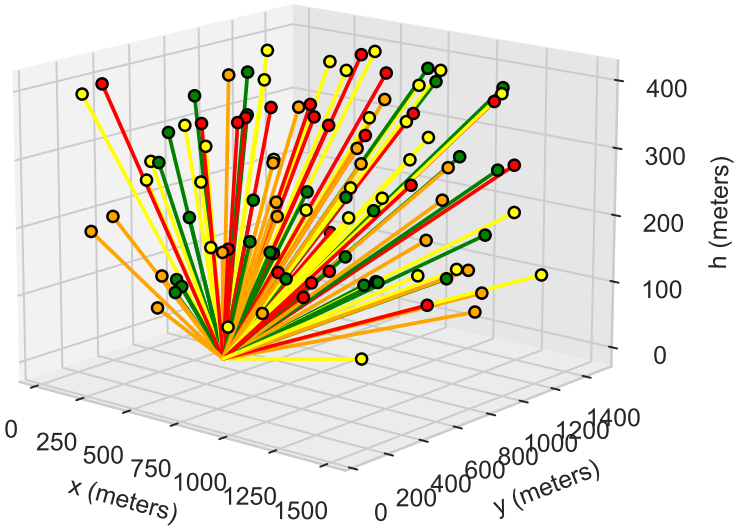
38. This change in the distribution of connections results in a reduction of delay in the proposed hybrid scheme that follows the FSMP energy model. The proposed hybrid scheme, which follows the FSMP energy model, demonstrates a decreased delay due to the increased data rates offered by BLE and LTE in comparison to the LoRa protocol. Thus the proposed hybrid approach, incorporating the FSMP energy model, minimizes the network energy consumption and reduces the mean network latency.

4.9.3 Performance Evaluation of the Proposed RL Algorithms

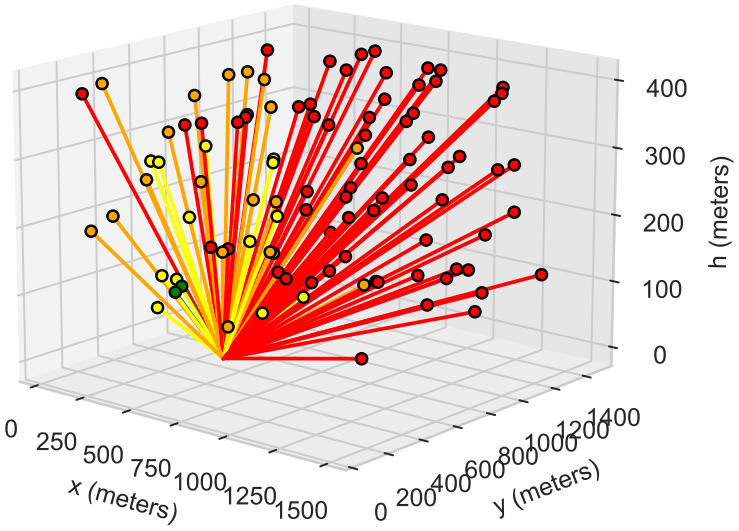
In this section, we provide numerical results that compare the performance of the proposed Q-learning (Algorithm 3) and DQN (Algorithm 4) algorithms with a rule-based algorithm and a hybrid random approach. The hybrid random approach involves randomly selecting UAV-GCS links from BLE, LTE, Wi-Fi, and LoRa. In addition, we demonstrate the efficacy of the proposed algorithms by taking into account the PHY layer parameter, which represents the number of transmissions needed for a packet to be successfully delivered. The simulation configuration resembles the preceding scenario, whereby a surface area measuring $1500 \times 1500\ m^2$ is taken into account, and the GCS is positioned at coordinates $(500, 500, 0)$. We examine a set of 100 UAVs that are placed randomly in the designated area, at different altitudes of 100, 200, 300, and 400 meters. The rest of the simulation parameters remain unchanged as provided in the preceding section.

We performed extensive simulations to analyze the behavior of Q-learning parameters. Primarily, we varied the learning rate and discount with several reward/penalty expressions as defined in (4.16), (4.17), (4.18), and (4.19). It was observed that the Q-learning technique yields an optimal hybrid network with $\lambda = 0.7$ and $\Delta = 0.7$, when the reward is selected from (4.16), (4.17), and (4.18).

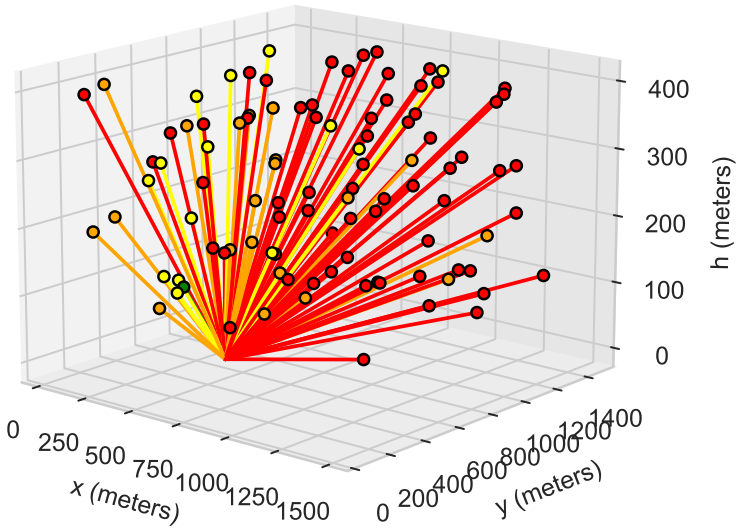
We adjusted the Q-learning parameters λ and Δ within the range of 0.1 to 0.9,


400
300
200
100
0
h (meters)
0
250
500
750
1000
1250
1500
x (meters)
0
200
400
600
800
1000
1200
1400
y (meters)

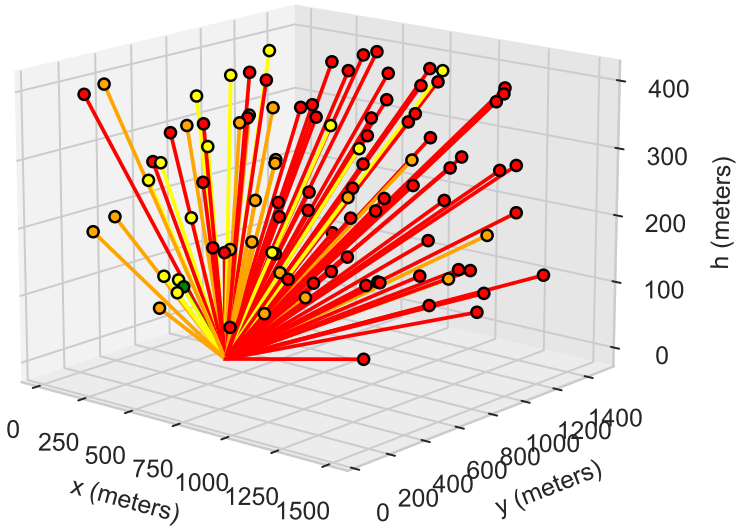
(a)


400
300
200
100
0
h (meters)
0
250
500
750
1000
1250
1500
x (meters)
0
200
400
600
800
1000
1200
1400
y (meters)

(b)



(c)



(d)

Figure 4.10: Topological representation of the hybrid UAV-assisted wireless network that is formed by utilizing (a) random hybrid scheme, (b) rule-based algorithm, (c) proposed Q-learning algorithm, and (d) proposed DQN algorithm. For all topologies, the GCS is situated at coordinates $(500, 500, 0)$. The communication technologies BLE, LTE, Wi-Fi, and LoRa are represented by the colors green, yellow, orange, and red, respectively [15].

Table 4.5: Various parameters used for the DQN algorithm [15].

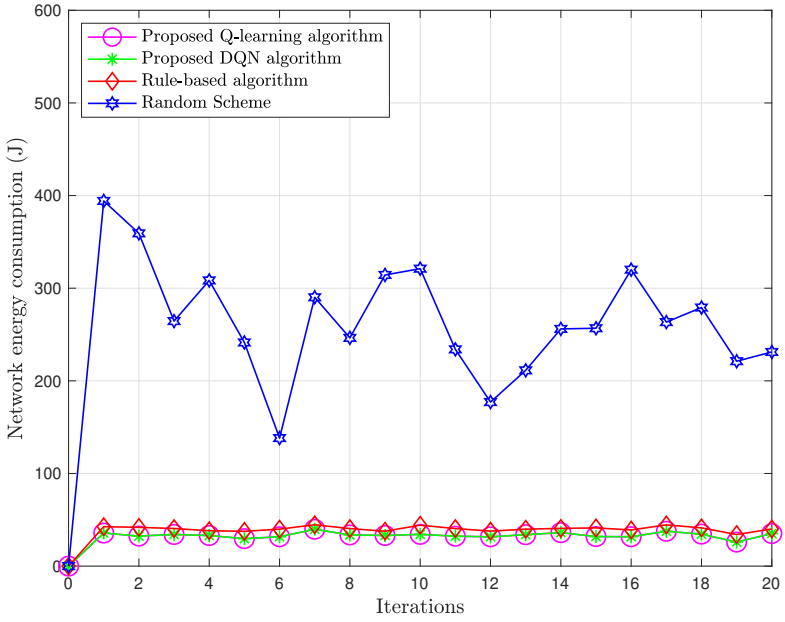
Parameter	**Value**
Learning rate (neural network)	0.01
Discount factor	0.4
Batch size	64
Minibatch size	64
Epsilon	1 to 0.001
Replay memory size	50000
Number of episodes	6400
Hidden layer count	2
Neuron count in first layer	256
Neuron count in second layer	128

with increments of 0.1, for the reward specified in (4.16), (4.17), and (4.18). It was found that when λ falls in the range of 0.4 to 0.9 for all Δ, the Q-learning algorithm formed the hybrid network with minimum energy consumption. The network energy consumption for various weights when the reward is selected as (4.19) is presented in Table 4.4. Here, the weights are selected from the range of $\{0.1, 0.2, \cdots, 0.9\}$. The optimal values for the proposed Q-learning method are $\lambda = 0.7$ and $\Delta = 0.6$. According to Table 4.4, it can be observed that the energy consumption is reduced when $W_1 = 0.1$ and $W_2 = 0.9$. This implies that $E_{2,N/W}$ has a greater influence compared to $E_{1,N/W}$ in achieving the hybrid network with the minimum energy consumption. The reason for this is that in the case of a UAV-GCS link containing k bits of data, the energy required for transmission, denoted as $E_{1,N/W}$, remains constant. However, the energy consumed for propagation, denoted as $E_{2,N/W}$, changes in accordance with the radial distance of the link. The training of the DQN algorithm is performed using identical simulation configuration as that of Q-learning. The policy and target networks consist of fully-connected neural networks, each comprising two hidden layers. The hidden layers include 256 and 128 neurons. The exploration-exploitation factor ε is varied within the range 1 to 0.001. Table 4.5 contains additional parameters pertaining to the DQN algorithm. The optimal performance was attained by setting the learning rate to 0.01 and the discount factor to 0.4. The DQN algorithm achieved convergence after a total of 6400 episodes, leading to the establishment of a hybrid UAV-assisted wireless network that consumes minimal network energy.

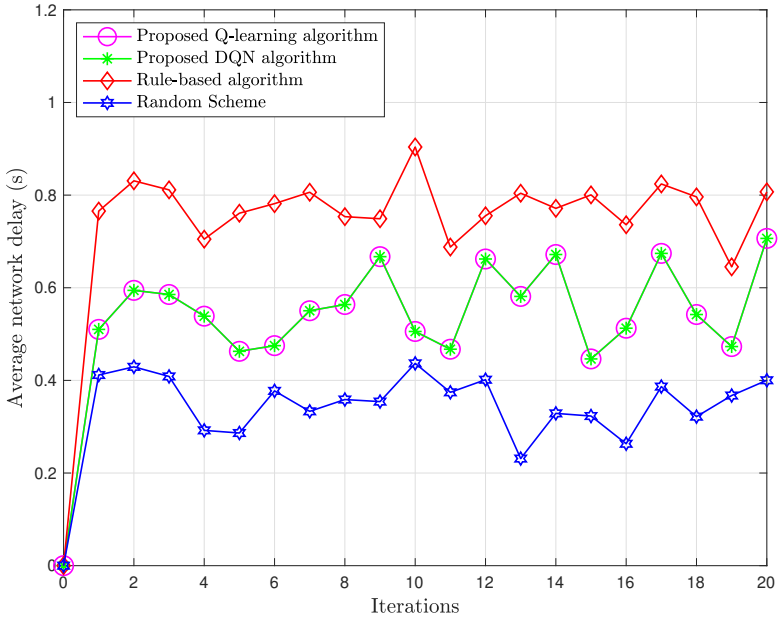
The topology of the hybrid random network is illustrated in Fig. 4.10a. In Fig. 4.10a, BLE is depicted by green lines, LTE by yellow lines, Wi-Fi by orange lines, and LoRa by red lines. The topological representation also illustrates that the connections are uniformly spread over the four protocols, regardless of any specific criteria, as a result of the random distribution. Fig. 4.10b illustrates the UAV-assisted wireless network that was developed using the rule-based algorithm. Based on Fig. 4.10b, it is evident that UAVs located at a greater radial distance from the GCS are linked to the LoRa protocol, and those in closer proximity to the GCS are

linked to BLE. The hybrid network formed with the proposed Q-learning and DQN algorithms is depicted in Fig. 4.10c and Fig. 4.10d respectively. From Fig. 4.10c and 4.10d, it is evident that the UAV-GCS link is connected to a communication technology that has minimum energy consumption with the required number of transmissions.

We evaluate the effectiveness of the proposed algorithms with the random hybrid scheme and the rule-based algorithm [170]. This performance evaluation is based on network energy consumption and average network delay as depicted in Fig. 4.11a and Fig. 4.11b, respectively. The results generated by the Q-learning method are derived from the reward defined in (4.17). The Q-learning and DQN algorithms showcase superior performance compared to the rule-based algorithm in terms of energy consumption and average delay, as depicted in Fig. 4.11a and Fig. 4.11b, respectively. This is because of the inherent learning nature of RL algorithms. During the training or exploration phase of RL algorithms, the environment characteristics are acquired, allowing for an accurate estimation of the network's energy consumption. Nevertheless, the rule-based algorithm determines the communication technology by considering the distance, regardless of the features of the PHY layer. The unknown PHY layer characteristics contribute to an increased frequency of retransmissions necessary for the successful transmission of bits, consequently leading to an increased energy consumption and delay. The proposed RL algorithms thus demonstrate a superior advantage over other schemes in terms of learning and integrating additional parameters that can impact the communication link. The random hybrid scheme exhibits a higher energy consumption in comparison to both the proposed and rule-based methods. As evident from Fig. 4.10a and Fig. 4.10c, this is because of the increased number of UAVs connected with BLE for the hybrid random network configuration as compared to the RL-based hybrid network. The energy consumption thus increases due to the fact that the majority of these links have a geographical distance exceeding r, resulting in an energy consumption of d^4. As compared to other communication technologies, BLE provides high data rate, hence the transmission delay is less, resulting in a decrease in the average network delay, as depicted in Fig. 4.11b. To provide further understanding on the performance of all schemes, we present explicit values derived from the simulation. During the 10^{th} iteration, the hybrid random network consumes 1160 Joules of energy, while the rule-based approach consumes 190 Joules of energy. Nevertheless, the proposed algorithms surpass all the other schemes by achieving a lower network energy consumption of approximately 151 Joules, as depicted in Fig. 4.11a. The rule-based algorithm demonstrates an average network delay of approximately 0.71 seconds, while the random hybrid scheme exhibits an average network delay of approximately 0.344 seconds at the 10^{th} iteration. Additionally, the proposed RL algorithms demonstrate an average network delay of 0.5 seconds at the 10^{th} iteration, as depicted in Fig. 4.11b. The DQN and Q-learning algorithm exhibit similar performance as evident from Fig. 4.11a and Fig. 4.11b. This is because DQN fundamentally adheres to the same mathematical concepts as Q-learning. In contrast to Q-learning, the DQN approach use a neural network to acquire information from



(a)



(b)

Figure 4.11: Performance evaluation of the rule-based algorithm, random hybrid scheme, proposed Q-learning algorithm, and proposed DQN algorithm. The performance is compared for random UAV settings with respect to the (a) energy consumption of the network and (b) average network delay. For all random configurations, the GCS is situated at coordinates $(500, 500, 0)$ [15].

the input and generate an estimated Q-value. As the amount of data increases, the process of storing and retrieving Q-values through Q-learning might result in a

decline in performance. In this case, DQN surpasses Q-learning in terms of reduced memory usage and computational efficiency. However, as compared to Q-learning, the DQN algorithm necessitates longer duration for training and fine-tuning. For instance, the duration needed to train Q-learning demands 20 minutes whereas DQN necessitates around 6 hours.

4.10 Summary

In this chapter, we have presented a framework of a hybrid BLE/Wi-Fi/LTE/LoRa switching scheme for UAV-assisted wireless networks. In the proposed framework, every UAV transitions to the communication protocol that exhibits reduced energy consumption while transmitting the available data. The proposed hybrid network can be formed by implementing two approaches. In the first method, the proposed hybrid network is formed by employing the FS and FSMP path loss models. Subsequently, we have put forth two RL algorithms, namely Q-learning and DQN, in order to form the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. The proposed RL algorithms accept a random network as an input and learn from it. Based on the learning from the input, the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network is then attained by assigning the optimal communication technology to a link. In addition, we have put forth a theoretical framework that addresses the energy-efficient transmission of data over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless networks. This framework is applicable to both FS and FSMP path loss models. In addition, we have obtained and validated the analytical expressions for the energy consumption and the average delay of the network through extensive analysis. In conclusion, it has been demonstrated that the proposed RL algorithms exhibit superior performance in terms of average network delay and network energy consumption in comparison to rule-based algorithms and random hybrid schemes. In the future, we intend to integrate the dataset acquired from physical layer parameters in order to assess the efficacy of the proposed algorithms. Furthermore, a potential avenue for research is examining the impact of introducing intelligent reflecting surfaces in the establishment of the hybrid network.

Chapter 5

Concluding Remarks

5.1 Conclusion

To conclude, this dissertation is a compilation of various publications related to the sensing and communication aspects of UAVs/drones. The dissertation provides a glimpse into some of the challenges encountered in a multi-UAV system. Novel cost-effective methods using state-of-the-art hardware technologies and sophisticated machine learning algorithms are proposed to combat and mitigate these issues. Through rigorous experimentation and testing, the proposed approaches are validated with the current state-of-the-art.

As mentioned above, the thesis is organized into two broad areas, sensing and communication. Chapter 3 discusses the sensing aspects of UAVs. In this chapter, we propose two cost-effective approaches to detect and estimate UAV count in the vicinity. The proposed approaches utilize thermal and acoustic sensors enhanced with lightweight machine learning algorithms. Additionally, to improve UAV localization, a novel approach utilizing rotating mmWave FMCW radar and cutting-edge computer vision algorithms is also proposed. In addition to the dynamics, the proposed approach also provides an accurate classification of the multiple target objects in the scene.

Chapter 4 provides a theoretical framework for obtaining a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network with energy-efficient data transmission. Analytical expressions for calculating the network energy consumption and average network delay are also derived. The chapter also proposes two approaches to form the hybrid network, one using the FS and FSMP path loss models and the other utilizing RL algorithms, namely Q-learning and DQN. Through extensive experiments, the results for the energy-efficient data transmission over a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network are verified and validated.

Thus, in conclusion, the research presented in this thesis underscores the significance of enhancing UAV sensing and communication with novel and innovative techniques. The techniques proposed in this thesis provide robust, cost-effective, and lightweight solutions that can be deployed on UAVs and edge computing devices. Overall, we believe this research has provided meaningful contributions to the scientific community and encourages the need for further exploration and innovation.

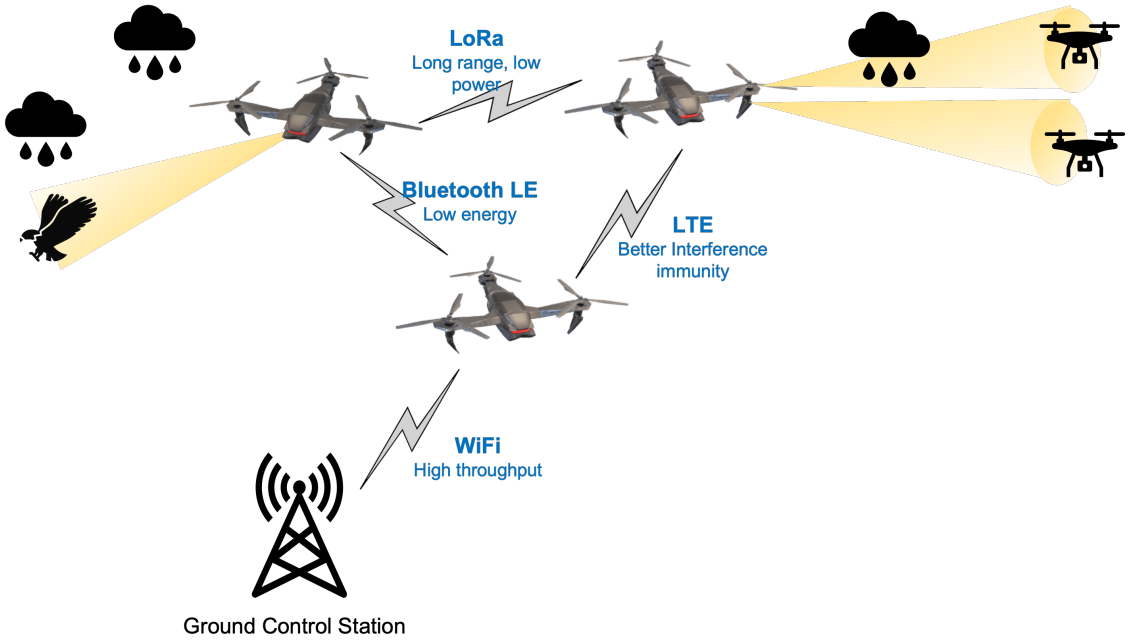


Figure 5.1: Research Summary

5.2 Future Research

In addressing some of the challenges presented in this thesis, it is evident that the proposed solutions can be further improved to increase performance and provide enhanced capabilities. Utilizing an omnidirectional polar patterned antenna for the acoustic-based approach is one step. For the rotating radar approach, exploring and investigating ways to incorporate the radar setup into maneuvering UAVs will significantly improve their detection and collision-avoidance performance. The thermal-based approach can be further enhanced by introducing regression to scale the detection performance to an even larger number of targets. Exploring the effect of introducing intelligent reflecting surfaces for the hybrid switching scheme will provide insight into maximizing the sum rate. Additionally, incorporating novel methods to optimize the machine learning models can also individually improve the proposed approaches.

However, to arrive at a multi-UAV autonomous system with enhanced sensing and communication capabilities, further research is required. Thus, taking a holistic view of the envisioned research project, as shown in Fig. 5.1, some future research directions are outlined below:

- As mentioned in the previous sections, the fusion of various sensor modalities is a potential future research direction. However, the fusion also entails further challenges like an increase in computational complexity, latency, power consumption, type of suitable modalities, etc. Thus a fair amount of exploration and experimentation is required to obtain a lightweight sensor fusion algorithm tuned to UAV applications.

- Tracking and path planning of multiple targets is also another direction of

research. Enabling smart tracking using current sensor setups and lightweight machine-learning algorithms can drastically improve system performance.

- Currently, the hybrid communication switching scheme is catered to serve the energy-efficient data transmission criteria. Incorporating additional parameter criteria to serve more practical scenarios is a venue for further research.

- The practical implementation of the hybrid switching scheme on UAVs using hardware modules like Pycom demands further investigation. In this scenario, the effect of the module weight, overall power consumption, interfacing with the flight controller, switching latency and formats, etc., requires much-needed effort.

We hope that this thesis has provided much-needed insight into the various challenges in UAV sensing and communication. We believe that our efforts will pave the way for additional research that can ultimately culminate in a truly autonomous multi-UAV system that can cater to the applications of the future.

Bibliography

[1] Drone market outlook in 2021. [Online]. Available: https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts.

[2] Panagiotis Katsigiannis et al. An autonomous multi-sensor UAV system for reduced-input precision agriculture applications. In *Proc. Mediterranean Conference on Control and Automation (MED)*, pages 60–64, Athens, Greece, Jun. 2016.

[3] Ario Yudo Husodo et al. Intruder drone localization based on 2D image and area expansion principle for supporting military defence system. In *Proc. IEEE ComNetSat*, pages 35–40, Makassar, Indonesia, 2019.

[4] Parikshit Maini and P. B. Sujit. On cooperation between a fuel constrained UAV and a refueling UGV for large scale mapping applications. In *Proc. ICUAS*, pages 1370–1377, Denver, CO, USA, 2015.

[5] Cornelius Holness et al. Remote sensing archeological sites through unmanned aerial vehicle (UAV) imaging. In *Proc. IEEE IGARSS*, pages 6695–6698, Beijing, China, 2016.

[6] Amazon Prime Air. [Online]. Available: https://www.amazon.com/b?node=8037720011.

[7] A. N. Wilson, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Embedded sensors, communication technologies, computing platforms and machine learning for UAVs: A review. *IEEE Sensors Journal*, 22(3):1807–1826, 2022.

[8] Sky Magic - Drone Light Shows. [Online]. Available: https://skymagic.show.

[9] Waleed Ejaz, Arslan Ahmed, Aliza Mushtaq, and Mohamed Ibnkahla. Energy-efficient task scheduling and physiological assessment in disaster management using UAV-assisted networks. *Computer Communications*, 155:150–157, Apr. 2020.

[10] A. N. Wilson et al. Recent advances in thermal imaging and its applications using machine learning: A review. *IEEE Sensors Journal*, 23(4):3395–3407, 2023.

[11] Wilson A N, Ajit Jha, Abhinav Kumar, and Linga Reddy Cenkeramaddi. Estimation of UAV count using thermal imaging and lightweight CNN. In *2023 11th International Conference on Control, Mechatronics and Automation (ICCMA)*, pages 92–96, 2023.

[12] Wilson AN, Ajit Jha, Abhinav Kumar, and Linga Reddy Cenkeramaddi. Estimation of number of unmanned aerial vehicles in a scene utilizing acoustic signatures and machine learning. *The Journal of the Acoustical Society of America*, 154(1):533–546, 2023.

[13] A. N. Wilson, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Multitarget angle of arrival estimation using rotating mmWave FMCW radar and yolov3. *IEEE Sensors Journal*, 23(3):3173–3182, 2023.

[14] Wilson A N et al. Hybrid BLE/LTE/Wi-Fi/LoRa switching scheme for UAV-assisted wireless networks. In *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 78–83, 2021.

[15] Wilson Ayyanthole Nelson et al. RL-based energy-efficient data transmission over Hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. *IEEE/ACM Transactions on Networking*, pages 1–16, 2023.

[16] Texas Instruments - mmWave IWR1843 - Datasheet. [Online]. Available: https://www.ti.com/product/IWR1843.

[17] Texas Instruments - mmWave IWR1443 - Datasheet. [Online]. Available: https://www.ti.com/product/IWR1443.

[18] Bram van Berlo et al. Millimeter wave sensing: A review of application pipelines and building blocks. *IEEE Sensors J.*, 21(9):10332–10368, Feb. 2021.

[19] Texas Instruments - mmWave radar sensors: What is mmWave. [Online]. Available: https://www.ti.com/sensors/mmwave-radar/what-is-mmwave.html.

[20] Texas Instruments mmWave Demo Visualizer. [Online]. Available: https://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/4.4.0/.

[21] Samiur Rahman and Duncan A. Robertson. Radar micro-Doppler signatures of drones and birds at K-band and W-band. *Scientific Reports*, 8(1):17396, November 2018.

[22] Martins Ezuma et al. Micro-UAV detection with a low-grazing angle millimeter wave radar. In *Proc. IEEE RWS*, pages 1–4, Orlando, FL, USA, 2019.

[23] Prabhat Kumar Rai et al. Localization and activity classification of unmanned aerial vehicle using mmWave FMCW radars. *IEEE Sensors J.*, 21(14):16043–16053, Apr. 2021.

[24] Siddharth Gupta et al. Target classification by mmWave FMCW radars using machine learning on range-angle images. *IEEE Sensors Journal*, 21(18):19993–20001, Sept. 2021.

[25] Linga Reddy Cenkeramaddi et al. A novel angle estimation for mmWave FMCW radars using machine learning. *IEEE Sensors Journal*, 21(8):9833–9843, Apr. 2021.

[26] Prabhat Kumar Rai et al. Angle and height estimation technique for aerial vehicles using mmWave FMCW radar. In *Proc. COMSNETS*, pages 104–108, Bangalore, India, 2021.

[27] FLIR Boson Thermal Imaging Core - Datasheet. [Online]. Available: https://flir.netx.net/file/asset/15754/original/attachment.

[28] FLIR Boson Compact LWIR - Datasheet. [Online]. Available: https://flir.netx.net/file/asset/12673/original/attachment.

[29] Hakan Kayan et al. Heat leakage detection and surveiallance using aerial thermography drone. In *Proc. SIU*, pages 1–4, Izmir, Turkey, 2018.

[30] Morris William Díaz and Juan José Cáceres. A novel application of drones: thermal diagnosis of electrical and telecommunications infrastructure. In *Proc. IEEE CONCAPAN*, pages 1–6, San Salvador, El Salvador, 2018.

[31] Frederik S. Leira, Tor Arne Johansen, and Thor I. Fossen. Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera. In *Proc. IEEE AeroConf*, pages 1–10, Big Sky, MT, USA, 2015.

[32] TELEDYNE FLIR. TELEDYNE FLIR Lepton. [Online]. Available: https://www.flir.eu/products/lepton/?model=500-0771-01&vertical=microcam&segment=oem.

[33] Semtech SX1276 - Datasheet. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rbr/6EfVZUorrpoKFfvaF_Fkpgp5kzjiNyiAbqcpqh9qSjE.

[34] LoRa - Homepage. [Online]. Available: https://lora-alliance.org/.

[35] LoRa Modulation Basics - Semtech. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001OJk/yDEcfAkD9qEz6oG3PJryoHKas3UMsMDa3TFqz1UQOkM.

[36] Bluetooth - What is Bluetooth LE? [Online]. Available: https://web.archive.org/web/20170310111443/https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy.

[37] Bluetooth, Bluetooth Technology. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/.

[38] Praveen Kumar Reddy Maddikunta et al. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sensors Journal*, pages 1–1, early access, 06 Jan. 2021.

[39] Pablo Ramon Soria et al. Bluetooth network for micro-UAVs for communication network and embedded range only localization. In *Proc. ICUAS*, pages 747–752, Miami, FL, USA, 2017.

[40] Evgeny Khorov et al. A Tutorial on IEEE 802.11ax high efficiency WLANs. *IEEE Commun. Surveys Tuts.*, 21(1):197–216, Sep. 2019.

[41] Antonio Guillen-Perez et al. WiFi networks on drones. In *Proc. ITU WT)*, pages 1–8, Bangkok, Thailand, 2016.

[42] 3GPP Standards for the IoT. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1805-iot_r14.

[43] TensorFlow Lite. [Online]. Available: https://www.tensorflow.org/lite.

[44] Raspberry Pi 4 Model B - Datasheet. [Online]. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf.

[45] Nvidia Jetson Nano - Technical Specifications. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[46] Nvidia Jetson Module - Comparison Table. [Online]. Available: https://developer.nvidia.com/embedded/jetson-modules.

[47] Nvidia Jetson AGX Xavier - Documentation. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit.

[48] Nvidia Jetson AGX Xavier - Technical Specifications. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-xavier.

[49] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv*, 2018.

[50] Joseph Redmon. Darknet: Open source neural networks in C. [Online]. Available: http://pjreddie.com/darknet/.

[51] Kaiming He et al. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, Nevada, USA, 2016.

[52] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing.

[54] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[55] Mark Sandler et al. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[56] Christian Szegedy et al. Rethinking the Inception architecture for computer vision. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[57] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proc. IEEE CVPR*, pages 8697–8710, 2018.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[59] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[60] ImageNet. [Online]. Available: https://www.image-net.org.

[61] Yoon Young Kim and Eung-Hun Kim. Effectiveness of the continuous wavelet transform in the analysis of some dispersive elastic waves. *The Journal of the Acoustical Society of America*, 110(1):86–94, 2001.

[62] L. Durak and O. Arikan. Short-time Fourier transform: Two fundamental properties and an optimal implementation. *IEEE Transactions on Signal Processing*, 51(5):1231–1242, 2003.

[63] Continuous Wavelet Transform (CWT. [Online]. Available: https://www.mathworks.com/help/wavelet/gs/continuous-wavelet-transform-and-scale-based-analysis.html.

[64] Ye Wang, Yueru Chen, Jongmoo Choi, and C-C Jay Kuo. Towards visible and thermal drone monitoring with convolutional neural networks. *APSIPA Transactions on Signal and Information Processing*, 8:e5, 2019.

[65] Adrian Carrio, Yucong Lin, Srikanth Saripalli, and Pascual Campoy. Obstacle detection system for small UAVs using ADS-B and thermal imaging. *Journal of intelligent & robotic systems*, 88:583–595, 2017.

[66] Petar Andraši, Tomislav Radišić, Mario Muštra, and Jurica Ivošević. Night-time detection of UAVs using thermal infrared camera. *Transportation Research Procedia*, 28:183–190, 2017.

[67] Naoual El-Djouher Mebtouche and Nadia Baha. Robust UAV detection based on saliency cues and magnified features on thermal images. *Multimedia Tools and Applications*, 82(13):20039–20058, 2023.

[68] Lars Sommer and Arne Schumann. Deep learning-based drone detection in infrared imagery with limited training data. In *Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies IV*, volume 11542, page 1154204. SPIE, 2020.

[69] Alexander Sedunov, Alexander Sutin, and Hady Salloum. Application of cross-correlation methods for passive acoustic unmannded aierial vehicle detection and tracking. *The Journal of the Acoustical Society of America*, 140(4):3119–3119, 2016.

[70] Alexander Sedunov et al. Stevens drone detection acoustic system and experiments in acoustics UAV tracking. In *Proc. IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7, 2019.

[71] Vladimir Kartashov et al. Use of acoustic signature for detection, recognition and direction finding of small unmanned aerial vehicles. In *Proc. IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, pages 1–4, 2020.

[72] Torea Blanchard, J-H Thomas, and Kosai Raoof. Acoustic localization and tracking of a multi-rotor unmanned aerial vehicle using an array with few microphones. *The Journal of the Acoustical Society of America*, 148(3):1456–1467, 2020.

[73] Giuseppe Ciaburro, Gino Iannace, and Amelia Trematerra. Research for the presence of unmanned aerial vehicle inside closed environments with acoustic measurements. *Buildings*, 10(5), 2020.

[74] Jian Fang, Anthony Finn, Ron Wyber, and Russell SA Brinkworth. Acoustic detection of unmanned aerial vehicles using biologically inspired vision processing. *The Journal of the Acoustical Society of America*, 151(2):968–981, 2022.

[75] Michael J Bianco et al. Machine learning in acoustics: Theory and applications. *The Journal of the Acoustical Society of America*, 146(5):3590–3628, 2019.

[76] Aro Ramamonjy, Eric Bavu, Alexandre Garcia, and Sébastien Hengy. A distributed network of compact microphone arrays for drone detection and tracking. *The Journal of the Acoustical Society of America*, 141(5):3651–3651, 2017.

[77] Bilal Taha and Abdulhadi Shoufan. Machine learning-based drone detection and classification: State-of-the-art in research. *IEEE Access*, 7:138669–138682, 2019.

[78] Muhammad Zohaib Anwar, Zeeshan Kaleem, and Abbas Jamalipour. Machine learning inspired sound-based amateur drone detection for public safety applications. *IEEE Transactions on Vehicular Technology*, 68(3):2526–2534, 2019.

[79] Valentin V Gravirov, Ruslan A Zhostkov, and Dmitriy A Presnov. Acoustic fields of unmanned aerial vehicles in the tasks of passive detection. *The Journal of the Acoustical Society of America*, 149(4):A35–A35, 2021.

[80] Bowon Yang et al. UAV detection system with multiple acoustic nodes using machine learning models. In *Proc. IEEE International Conference on Robotic Computing (IRC)*, pages 493–498, 2019.

[81] Andrea Bernardini, Federica Mangiatordi, Emiliano Pallotti, and Licia Capodiferro. Drone detection by acoustic signature identification. *Electronic Imaging*, 2017(10):60–64, 2017.

[82] Juhyun Kim et al. Real-time UAV sound detection and analysis system. In *IEEE Sensors Applications Symposium (SAS)*, pages 1–5, 2017.

[83] Xuejun Yue et al. Software defined radio and wireless acoustic networking for amateur drone surveillance. *IEEE Communications Magazine*, 56(4):90–97, 2018.

[84] Wenshuai Wang, Kuangang Fan, Qinghua Ouyang, and Ye Yuan. Acoustic UAV detection method based on blind source separation framework. *Applied Acoustics*, 200:109057, 2022.

[85] Michael W Berry, Azlinah Mohamed, and Bee Wah Yap. *Supervised and unsupervised learning for data science*. Springer, 2019.

[86] Sungho Jeon et al. Empirical study of drone sound detection in real-life environment with deep neural networks. In *Proc. European Signal Processing Conference (EUSIPCO)*, pages 1858–1862, 2017.

[87] Yoojeong Seo, Beomhui Jang, and Sungbin Im. Drone detection using convolutional neural networks with acoustic STFT features. In *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018.

[88] Luke Russell, Rafik Goubran, and Felix Kwamena. Emerging urban challenge: RPAS/UAVs in cities. In *Proc. International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 546–553, 2019.

[89] Pietro Casabianca and Yu Zhang. Acoustic-based UAV detection using late fusion of deep neural networks. *Drones*, 5(3), 2021.

[90] Sara Al-Emadi, Abdulla Al-Ali, Amr Mohammad, and Abdulaziz Al-Ali. Audio based drone detection and identification using deep learning. In *Proc. International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 459–464, 2019.

[91] Tianyuan Yang et al. An adaptive radar signal processor for UAVs detection with super-resolution capabilities. *IEEE Sensors Journal*, 21(18):20778–20787, 2021.

[92] A. Aubry, Vincenzo Carotenuto, Antonio Maio, and G. Foglia. Exploiting multiple a priori spectral models for adaptive radar detection. *Radar, Sonar & Navigation, IET*, 8:695–707, 08 2014.

[93] Augusto Aubry, Vincenzo Carotenuto, Antonio De Maio, and Mark A. Govoni. Multi-snapshot spectrum sensing for cognitive radar via block-sparsity exploitation. *IEEE Transactions on Signal Processing*, 67(6):1396–1406, 2019.

[94] A. Aubry, Antonio Maio, Luca Pallotta, and Alfonso Farina. Median matrices and their application to radar training data selection. *Radar, Sonar & Navigation, IET*, 8:265–274, 04 2014.

[95] Paolo Braca et al. Multi-class random matrix filtering for adaptive learning. *IEEE Transactions on Signal Processing*, 68:359–373, 2020.

[96] A. Aubry, A. De Maio, and V. Carotenuto. Optimality claims for the FML covariance estimator with respect to two matrix norms. *IEEE Transactions on Aerospace and Electronic Systems*, 49(3):2055–2057, 2013.

[97] Wenqiang Wei et al. DOA estimation of distributed mmWave radar system via fast iterative adaptive approach. In *Proc. International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 414–418, Xi'an, China, 2021.

[98] Muhammad Z. Ikram, Murtaza Ali, and Dan Wang. Joint antenna-array calibration and direction of arrival estimation for automotive radars. In *Proc. IEEE Radar Conference (RadarConf)*, pages 1–5, Philadelphia, PA, USA, 2016.

[99] Daegun Oh and Jong-Hun Lee. Low-complexity range-azimuth FMCW radar sensor using joint angle and delay estimation without SVD and EVD. *IEEE Sensors Journal*, 15(9):4799–4811, Sept. 2015.

[100] Sangdong Kim and Kyun-Kyung Lee. Low-complexity joint extrapolation-MUSIC-based 2D parameter estimator for vital FMCW radar. *IEEE Sensors Journal*, 19(6):2205–2216, March 2019.

[101] Shota Nishiura and Hiroshi Yamamoto. Large-term sensing system for agriculture utilizing UAV and wireless power transfer. In *Proc. International Conference on Information Networking (ICOIN)*, pages 609–614, Jeju Island, South Korea, Jan. 2021.

[102] Vicente Mayor, Rafael Estepa, Antonio Estepa, and German Madinabeitia. Deploying a reliable UAV-aided communication service in disaster areas. *Wireless Communications and Mobile Computing*, 2019:e7521513, April 2019.

[103] Shie-Yuan Wang, Jui-En Chang, Hsin Fan, and Yi-Hsiu Sun. Performance comparisons of NB-IoT, LTE Cat-M1, Sigfox, and LoRa moving at high speeds in the air. In *Proc. IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, Rennes, France, Jul. 2020.

[104] Yeduri Sreenivasa Reddy et al. Optimisation of indoor hybrid PLC/VLC/RF communication systems. *IET Communications*, 14(1):117–126, Jan. 2020.

[105] Mohamed Kashef, Mohamed Abdallah, and Naofal Al-Dhahir. Transmit power optimization for a hybrid PLC/VLC/RF communication system. *IEEE Transactions on Green Communications and Networking*, 2(1):234–245, Nov. 2018.

[106] Eyuel D. Ayele, Kallol Das, Nirvana Meratnia, and Paul J.M. Havinga. Leveraging BLE and LoRa in IoT network for wildlife monitoring system (WMS). In *Proc. IEEE World Forum on Internet of Things (WF-IoT)*, pages 342–348, Singapore, 2018.

[107] Muhammad Asghar Khan, Ijaz Mansoor Qureshi, and Fahimullah Khanzada. A hybrid communication scheme for efficient and low-cost deployment of future flying ad-hoc network (FANET). *Drones*, 3(1), Feb. 2019.

[108] Mengjie Yi et al. Deep reinforcement learning for fresh data collection in UAV-assisted IoT networks. In *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 716–721, Toronto, ON, Canada, 2020.

[109] Shu Fu et al. Towards energy-efficient UAV-assisted wireless networks using an artificial intelligence approach. *IEEE Wireless Communications*, pages 1–11, May 2022.

[110] Sabarish Krishna Moorthy, Maxwell Mcmanus, and Zhangyu Guan. ESN reinforcement learning for spectrum and flight control in THz-enabled drone networks. *IEEE/ACM Transactions on Networking*, 30(2):782–795, Nov. 2022.

[111] Yirga Yayeh Munaye, Rong-Terng Juang, Hsin-Piao Lin, and Getaneh Berie Tarekegn. Resource allocation for multi-UAV assisted IoT networks: A deep reinforcement learning approach. In *Proc. International Conference on Pervasive Artificial Intelligence (ICPAI)*, pages 15–22, Taipei, Taiwan, 2020.

[112] Xijian Zhong, Yan Guo, Ning Li, and Yancheng Chen. Joint optimization of relay deployment, channel allocation, and relay assignment for UAVs-aided D2D networks. *IEEE/ACM Transactions on Networking*, 28(2):804–817, Feb. 2020.

[113] Matthew Ritchie, Francesco Fioranelli, and Hervé Borrion. Micro UAV crime prevention: Can we help Princess Leia? *Crime Prevention in the 21st Century: Insightful Approaches for Crime Prevention Initiatives*, pages 359–376, 2017.

[114] Linga Reddy Cenkeramaddi et al. A survey on sensors for autonomous systems. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1182–1187, 2020.

[115] Ryze Robotics. Tello EDU. [Online]. Available: https://www.ryzerobotics.com/tello-edu.

[116] DJI. DJI Camera Drones. [Online]. Available: https://www.dji.com/no/camera-drones.

[117] SYMA X30 Foldable Drone. [Online]. Available: https://www.symatoys.com/goodshow/x30-syma-x30-foldable-drone.html.

[118] Picamera. Picamera Documentation. [Online]. Available: https://picamera.readthedocs.io/en/release-1.13/.

[119] Ubuntu. Ubuntu 18.04. [Online]. Available: https://releases.ubuntu.com/18.04/.

[120] ROS Wiki. ROS Melodic. [Online]. Available: http://wiki.ros.org/melodic.

[121] Wei Nie et al. UAV detection and identification based on WiFi signal and RF fingerprint. *IEEE Sensors Journal*, 21(12):13540–13550, 2021.

[122] Jesse Callanan et al. Ergonomic impact of multi-rotor unmanned aerial vehicle noise in warehouse environments. *Journal of Intelligent & Robotic Systems*, 100:1309–1323, 2020.

[123] Jesse Callanan et al. Large-aperture experimental characterization of the acoustic field generated by a hovering unmanned aerial vehicle. *The Journal of the Acoustical Society of America*, 150(3):2046–2057, 2021.

[124] Rayhaan Iqbal et al. Efficient training of transfer mapping in physics-infused machine learning models of UAV acoustic field. In *AIAA SCITECH 2022 Forum*, page 0384, 2022.

[125] DJI Mavic 2. [Online]. Available: https://www.dji.com/no/mavic-2-enterprise.

[126] DJI Mini 2. [Online]. Available: https://www.dji.com/no/mini-2.

[127] DJI Mini SE. [Online]. Available: https://www.dji.com/no/mini-se.

[128] DJI Mini 3 Pro. [Online]. Available: https://www.dji.com/no/mini-3-pro.

[129] DJI Tello EDU. [Online]. Available: https://store.dji.com/no/product/tello.

[130] Shure MV7. [Online]. Available: https://www.shure.com/en-MEA/products/microphones/mv7?variant=MV7-K.

[131] Shure MV7 User Guide. [Online]. Available: https://pubs.shure.com/guide/MV7/en-US.

[132] Aveen Dayal et al. Lightweight deep convolutional neural network for background sound classification in speech signals. *The Journal of the Acoustical Society of America*, 151(4):2773–2786, 2022.

[133] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proc. International Conference on Machine Learning (ICML)*, pages 111–118, 2010.

[134] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning (ICML)*, pages 448–456. PMLR, 2015.

[135] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[136] Tesla V100 GPU. [Online]. https://www.nvidia.com/en-us/data-center/v100/.

[137] Keras. [Online]. Available: https://keras.io.

[138] Scikit-Learn: Cross-validation. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html.

[139] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proc. AAAI conference on artificial intelligence*, volume 31, 2017.

[140] Andrew Howard et al. Searching for MobileNetV3. In *Proc. IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[141] Hojun Son and James Weiland. Semantic segmentation optimized for low compute embedded devices. *IEEE Access*, 10:96514–96525, 2022.

[142] Model Profiler (TensorFlow). [Online]. Available: https://pypi.org/project/model-profiler/.

[143] Torchstat (PyTorch). [Online]. https://pypi.org/project/torchstat/.

[144] GPU Delegates - TensorFlow Lite. [Online]. Available: https://www.tensorflow.org/lite/performance/delegates.

[145] Shan Ullah and Deok-Hwan Kim. Benchmarking jetson platform for 3D point-cloud and hyper-spectral image classification. In *Proc. IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 477–482, 2020.

[146] The fundamentals of millimeter wave radar sensors. [Online]. Available: https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf.

[147] FMCW radars - Range estimation. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-1-range-estimation.

[148] FMCW radars - Angle estimation. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-5-angle-estimation.

[149] Texas Instruments AWR2243. [Online]. Available: https://www.ti.com/document-viewer/AWR2243/datasheet.

[150] Texas Instruments AWR1843. [Online]. Available: https://www.ti.com/document-viewer/AWR1843/datasheet/device-overview-x3342#x3342.

[151] MATLAB. [Online]. Available: https://www.mathworks.com/products/matlab.html.

[152] LabelImg software. [Online]. Available: https://github.com/tzutalin/labelImg.

[153] DG Oh, YH Ju, and JH Lee. Subspace-based auto-paired range and DOA estimation of dual-channel FMCW radar without joint diagonalisation. *Electronics letters*, 50(18):1320–1322, Aug. 2014.

[154] Sangdong Kim, Daegun Oh, and Jonghun Lee. Joint DFT-ESPRIT estimation for TOA and DOA in vehicle FMCW radars. *IEEE Antennas and Wireless Propagation Letters*, 14:1710–1713, Apr. 2015.

[155] Daegun Oh, Yeonghwan Ju, Hyunsoo Nam, and Jong-Hun Lee. Dual smoothing DOA estimation of two-channel FMCW radar. *IEEE Transactions on Aerospace and Electronic Systems*, 52(2):904–917, Apr. 2016.

[156] Wen-Hsien Fang and Li-Der Fang. Joint angle and range estimation with signal clustering in FMCW radar. *IEEE Sensors Journal*, 20(4):1882–1892, Oct. 2019.

[157] Nian Xia, Hsiao-Hwa Chen, and Chu-Sing Yang. Emerging technologies for machine-type communication networks. *IEEE Network*, 34(1):214–222, Feb. 2020.

[158] Pycom OEM module. [Online]. Available: https://pycom.io/product/l04-oem-module/.

[159] Zhenhui Yuan et al. Ultra-reliable IoT communications with UAVs: A swarm use case. *IEEE Communications Magazine*, 56(12):90–96, Dec. 2018.

[160] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[161] S. M. Mahdi H. Daneshvar, Pardis Alikhah Ahari Mohajer, and Sayyed Majid Mazinani. Energy-efficient routing in WSN: A centralized cluster-based approach via grey wolf optimizer. *IEEE Access*, 7:170019–170031, Nov. 2019.

[162] 3GPP Release 13 Overview. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1628-rel13.

[163] Bluetooth Wireless Technology. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/.

[164] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*. MIT press Cambridge, 1998.

[165] Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[166] Kai-Chu Tsai et al. Tensor-based reinforcement learning for network routing. *IEEE Journal of Selected Topics in Signal Processing*, 15(3):617–629, 2021.

[167] Fan Jiang, Zesheng Gu, Changyin Sun, and Rongxin Ma. Dynamic user pairing and power allocation for NOMA with deep reinforcement learning. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2021.

[168] Yejian Zhao et al. Dynamic jobshop scheduling algorithm based on deep Q network. *IEEE Access*, 9:122995–123011, 2021.

[169] Sifat Rezwan and Wooyeol Choi. Priority-based joint resource allocation with deep Q-learning for heterogeneous NOMA systems. *IEEE Access*, 9:41468–41481, 2021.

[170] H Chtourou and Mohamed Haouari. A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. *Computers & industrial engineering*, 55(1):183–194, 2008.

Appendix A

PAPER A

Title: Embedded Sensors, Communication Technologies, Computing Platforms and Machine Learning for UAVs: A Review

Authors: **A. N. Wilson**, A. Kumar, A. Jha, L. R. Cenkeramaddi

Journal: IEEE Sensors Journal

Embedded Sensors, Communication Technologies, Computing Platforms and Machine Learning for UAVs: A Review

A. N. Wilson, A. Kumar, A. Jha, L. R. Cenkeramaddi

Abstract: Unmanned aerial vehicles (UAVs) are increasingly becoming popular due to their use in many commercial and military applications, and their affordability. The UAVs are equipped with various sensors, hardware platforms and software technologies which enable them to support the diverse application portfolio. Sensors include vision-based sensors such as RGB-D cameras, thermal cameras, light detection and ranging (LiDAR), mmWave radars, ultrasonic sensors, and an inertial measurement unit (IMU) which enable UAVs for autonomous navigation, obstacle detection, collision avoidance, object tracking and aerial inspection. To enable smooth operation, UAVs utilize a number of communication technologies such as wireless fidelity (Wi-Fi), long range (LoRa), long-term evolution for machine-type communication (LTE-M), etc., along with various machine learning algorithms. However, each of these different technologies come with their own set of advantages and challenges. Hence, it is essential to have an overview of the different type of sensors, computing and communication modules and algorithms used for UAVs. This paper provides a comprehensive review on the state-of-the-art embedded sensors, communication technologies, computing platforms and machine learning techniques used in autonomous UAVs. The key performance metrics along with operating principles and a detailed comparative study of the various technologies are also studied and presented. The information gathered in this paper aims to serve as a practical reference guide for designing smart sensing applications, low-latency and energy efficient communication strategies, power efficient computing modules and machine learning algorithms for autonomous UAVs. Finally, some of the open issues and challenges for future research and development are also discussed.

A.1 Introduction

Recent advances in sensor miniaturization, ubiquitous wireless connectivity, en-

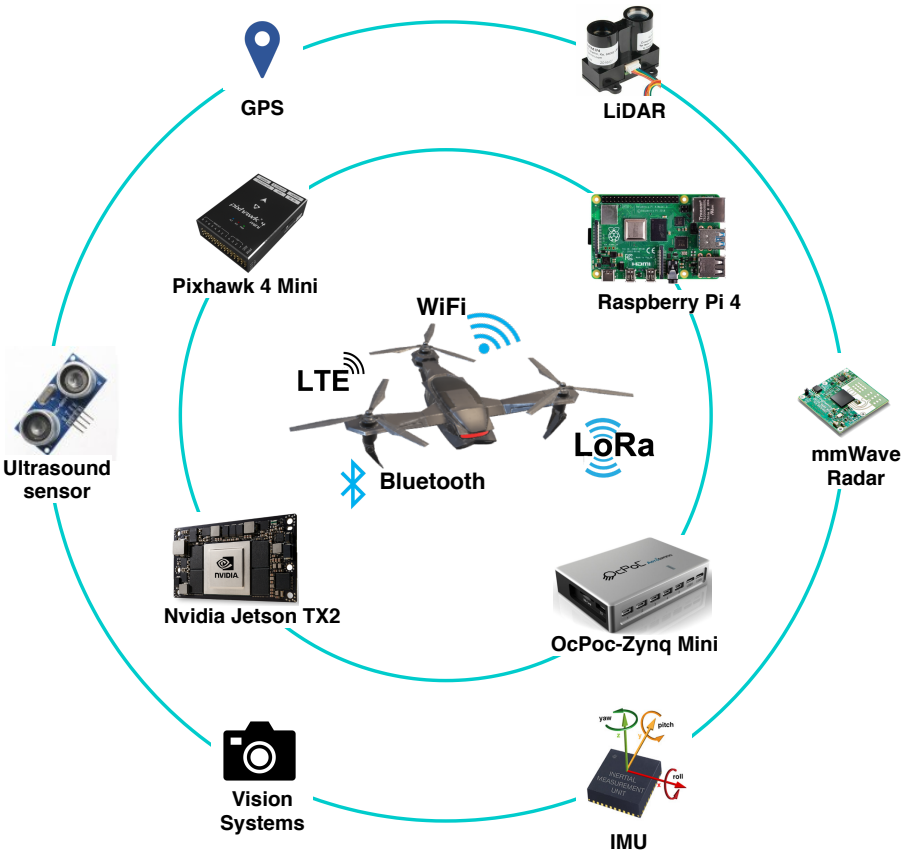


Figure A.1: Diagramatic overview of the survey

hanced processing power and low complexity algorithms have contributed to the growing demand for unmanned aerial vehicle (UAV) or drone-based applications. According to market estimates, the revenue generated from these broad range of UAV applications is expected to exceed 8.5 billion dollars by 2027 [1]. Additionally, UAV-based architectures are explored for the development and implementation of next generation technologies such as 5G, vehicle-to-everything (V2X) communications, etc.

UAVs are originally used in military applications to survey and target enemy territory. However, recent technological advancements have led to the use of UAVs in a wide range of applications spanning multiple industries. Agriculture, disaster management, surveillance, package delivery, and aerial photography are some of the common applications for UAVs. UAVs are used in agriculture to monitor crop health and irrigation. [2]. In defense, they are used for intruder detection and attack [3]. Surveillance applications use UAVs for mapping large areas [4]. UAVs with remote sensing capabilities are useful for scanning large geographical areas for archaeological applications [5]. Recently, UAVs have made their way into the e-commerce industry. Amazon has demonstrated a UAV-based package delivery system called Prime Air [6] that can deliver shipments to customers in remote areas within a specific time.

UAVs provide enhanced aerial inspection, improved line-of-sight (LoS) communication, reliable data acquisition and seamless obstacle free movement which contribute for their wide spread adoption across diverse applications. Furthermore,

their ability to access remote locations and capture images makes them far superior and more flexible than ground-based systems. In addition, factors such as cutting-edge computational resources, readily available components, and low cost have made UAVs the obvious choice for a wide range of application requirements.

To accommodate the wide range of supported applications, UAVs are outfitted with highly sophisticated hardware and software modules. Embedded sensors, communication modules, and computing platforms are among the hardware components, while the software stack supports UAV configuration aspects, control and stabilization algorithms, mission planning, and testing. Additionally, machine learning and deep learning approaches are also utilized to support the various collision avoidance and stabilization algorithms. The hardware and software components are tightly coupled and work in tandem to allow UAVs to fly and perform various operations. As a result, comprehending and appreciating UAV operations requires an understanding of the complex interplay of the underlying technologies.

There are some survey papers in the literature that provide information on the various technologies used in UAVs. In [7], the authors have provided a brief description of the different sensors used in autonomous systems. The principle of operation of each sensor along with their key performance metrics are outlined. Whereas, in [8], a comprehensive review of the diverse computing platforms along with on-board flight controller software is discussed. Information about the different communication technologies used in UAVs together with the open research challenges is presented in [9] and [10]. Recently, the use of machine learning algorithms for UAV and ground control station (GCS) applications have gained prominence [11], [12]. In [13], the use of UAVs for smart agriculture is explored. The authors attempted to describe the various agricultural sensors used on board UAVs, as well as potential future research directions and challenges. Table B.2 provides a brief comparison of existing surveys with the current survey.

As can be seen, the preceding studies focus on specific aspects of UAVs such as applications, components, and software and thus fail to provide a high level practical overview of the system in general. There is a need for a coherent and concise review of UAV literature that can serve as a practical guide for the novice learner, given the abundance of literature on the subject. As a result, the primary goal of this paper is to provide a practical perspective of the UAV system and equip the reader with the tools and techniques needed to deploy a UAV system especially focusing on sensors aspects. The following are the main contributions of this survey article:

- Overview of the UAV system describing the various components and their interactions.

- Description of various embedded sensors used in UAVs outlining their operating principle, along with key performance metrics and limitations.

- Communication technologies used in UAVs to transfer information among UAVs and GCS modules.

Table A.1: Comparison between existing surveys and this survey

Year	Reference	Focus areas	ES	CT	CP	ML
2018	[8]	UAV flight controller hardware and software	✗	✗	✓	✗
2019	[10]	UAV wireless communication, cellular-connected UAVs	✗	✓	✗	✗
2019	[11]	Machine learning for cellular-connected UAVs	✗	✓	✗	✓
2020	[9]	UAV communication technologies	✗	✓	✗	✗
2020	[7]	Embedded sensors for autonomous systems	✓	✗	✗	✗
2021	[13]	Communication and sensor technologies for UAVs focussed on agriculture applications	✓	✓	✗	✗
2021	[12]	Deep learning for UAVs	✗	✗	✗	✓
2021	**This work**	Embedded sensors, computing platforms, communication technology, machine learning with focus on sensing and communication for UAVs	✓	✓	✓	✓

ES - Embedded Sensors; **CT** - Communication Technology; **CP** - Computing Platform; **ML** - Machine Learning.

- Computing platforms that can be equipped on UAVs with emphasis on computational resources and easy integration of hardware and software.
- Machine learning algorithms that are primarily focused on sensor fusion and communication for UAVs.
- Finally, potential future research directions are presented.

The remainder of this paper is organized as shown in Fig. A.2. Section II provides an overview of the UAV system. Section III summarizes the various embedded on-board sensors in UAVs. Section IV focuses on the different on-board communication modules. Section V discusses the computing platforms (hardware and software) that are used in tandem with UAVs. Section VI investigates the various machine learning algorithms employed in UAVs. Section VII explores the relevant potential future directions and finally, section VIII concludes the paper.

A.2 UAV System Overview

In order to better understand and appreciate the content of this paper, it is necessary to first learn about the various components/subsystems used in UAVs. A more

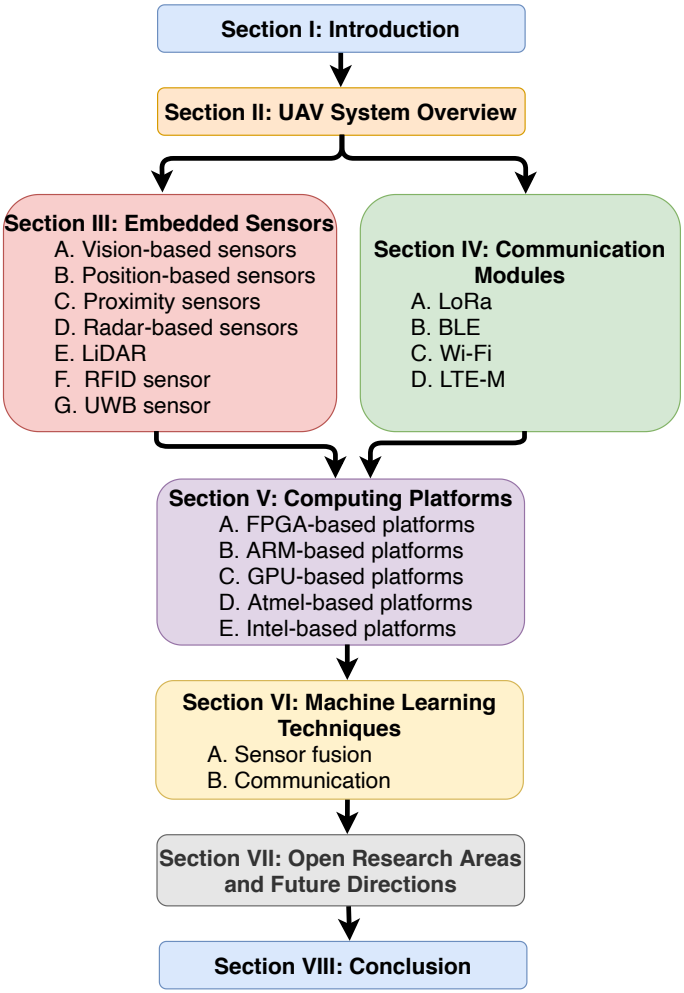


Figure A.2: Structure of the article.

comprehensive approach would be to represent the UAV system abstractly, as shown in Fig. A.3. This view of the UAV system is built by taking into account the flexibility and portability provided by various UAV components in order to cater to a variety of applications.

The ability of UAVs to sense and perceive their surroundings is enabled by various hardware sensor modules. Each sensor is unique in terms of its operation, form factor, cost, performance and output information. Based on application requirements, sensors with varying form factor and performance can be used on UAVs. Agricultural applications rely heavily on temperature and vision-based sensors to monitor crop health, whereas surveillance and remote sensing primarily use camera and light detection and ranging (LiDAR)-based modules.

UAVs are equipped with communication modules that allow data to be transmitted between UAVs and the GCS. Each communication module operates at the specified frequency, bandwidth, power, and coverage. Depending on the application, a suitable communication technology must be chosen, taking into account its capabilities and characteristics. Wireless fidelity (Wi-Fi) modules are used to transfer information among UAVs in applications that require a high data rate, whereas long range (LoRa) modules are used in search and rescue operations that require information transfer over a long distance.

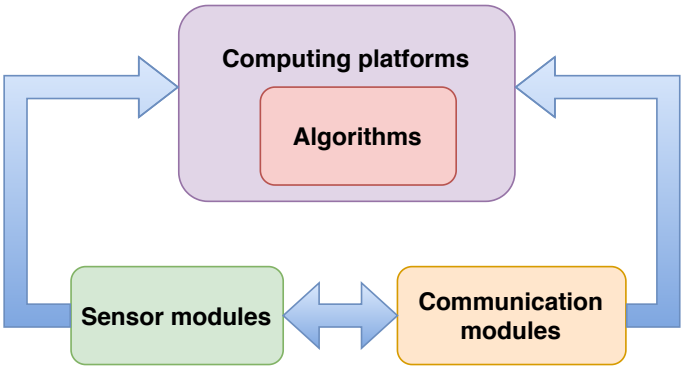


Figure A.3: UAV system overview.

Various sensor and communication modules are integrated on UAVs using computing platforms. Computing platforms are hardware processing units that can connect and act as a medium for processing data from various sensor and communication modules. The type of computing platform to be used on UAVs is determined by the application's requirement specifications. Processing power, interfacing options for connecting with hardware sensor and communication modules, form factor, weight, and other factors are considered when selecting a computing unit for UAVs.

Additionally, the computing platforms host a variety of algorithms that are critical to ensuring the safe and secure operation of UAVs. The algorithms are designed to cater to various application scenarios while also being low in complexity and power consumption. Sensor fusion techniques, interference mitigation schemes, control and stabilization strategies, and so on are examples of these algorithms. Several machine learning and deep learning algorithms have recently been shown to improve performance in various aspects of UAV operations. This review will concentrate on some of the most recently developed machine learning algorithms that can be used in a variety of UAV applications.

Decoupling the UAV system into the building blocks depicted in Fig. A.3 allows for selective analysis and learning of the various aspects of UAVs. This paper's content focuses on each of these blocks separately, providing useful and necessary information from a practical standpoint.

A.3 Embedded Sensors

Embedded sensors are critical in many UAV functions, including autonomous operations, collision avoidance, tracking, communication, and so on as shown in Fig. A.4. These sensors have been classified into vision-based sensors, position-based sensors, proximity sensors, radar, and LiDAR sensors depending upon the type of information they produce. This section provides a high-level overview of these sensors, as well as information on their operating principles and key performance metrics. Other considerations include cost, output format, and power consumption.

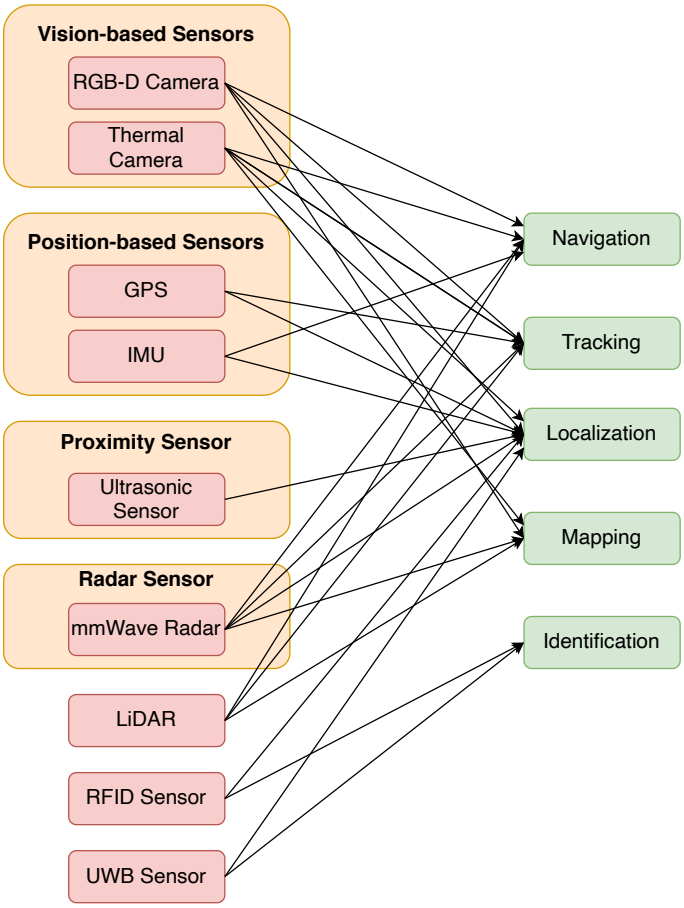


Figure A.4: UAV embedded sensor applications.

A.3.1 Vision-based Sensors

Vision-based sensors generate an image of the captured scene, providing a visual perception of the environment. The generated image is then processed and used with various image processing and computer vision algorithms to ensure and enable a variety of UAV operations and services. The RGB-D cameras and thermal cameras are two popular vision-based technologies, and they are summarized in the following paragraphs.

A.3.1.1 RGB-D Camera

The most common sensors used on UAVs are RGB-D cameras. These sensors provide perception of the surroundings in the form of RGB images. Because RGB images are closely related to human visual perception, it is simple to make sense of the information obtained. Nonetheless, the images can be fed into image processing and computer vision algorithms to perform target detection, localization, and tracking. Furthermore, depth information can be used to calculate the spatial distance of targets from the camera, which is useful for collision avoidance operations. The resolution of RGB cameras and other considerations such as the camera's frame rate, shutter type, and aperture determine their image quality [14], [15].

RGB-D cameras on board UAVs are used for object detection, collision avoidance,

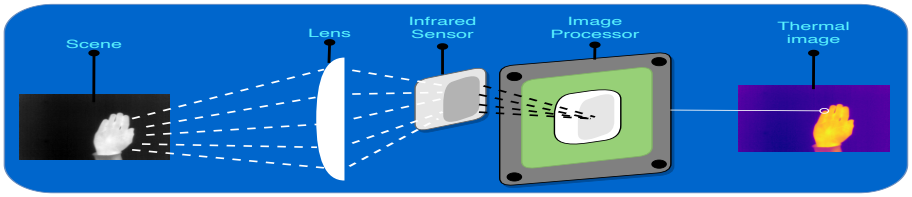


Figure A.5: Working principle of thermal camera.

and tracking. This can in seen in [16] where the authors have used a single RGB-D camera to implement a collision avoidance system which when integrated with a bin-occupancy filter can be used for tracking. Another setup described in [17] uses RGB-D camera along with the inertial measurement unit (IMU) data to provide simultaneous localization and mapping (SLAM).

A.3.1.2 Thermal Camera

Thermal cameras mounted on UAVs can aid in search and rescue operations, disaster management, and surveillance applications. These cameras can work in low-light and robust weather conditions. Thermal cameras use special sensors which can capture the infrared radiations falling on them. The captured radiation information is then processed to generate a temperature profile which is used to improve detection and classification performance. Fig. A.5 depicts the working principle of thermal cameras. When choosing thermal cameras, some of the key parameters to consider are image resolution, range, refresh rate, and lens focal length [18], [19].

When integrated with UAVs, thermal imaging has a wide range of applications. The authors of [20] used UAV equipped with thermal cameras to capture thermal images, which were then processed to detect heat leaks in buildings. In [21], thermal camera mounted UAVs are used to autonomously monitor and detect wildlife. UAVs equipped with thermal cameras are also used to detect, identify, and track objects in the ocean [22].

A.3.2 Position-based Sensors

Position-based sensors detect movement and can also provide relative position information in relation to a known reference point. These sensors are used in UAVs to pinpoint the precise location of UAVs in a given area. Furthermore, these sensors can provide odometric information about UAVs, which helps to determine the orientation of the UAV. Some of the relevant positional sensors used in UAVs are the global positioning system (GPS) and IMUs, which will be discussed in the following subsections.

A.3.2.1 GPS

The GPS is a global radio navigation system that is used in a variety of applications that utilize positional information. GPS works on the trilateration principle [23] using a system of atleast 24 active satellites and GCSs as shown in Fig. A.6.

Figure A.6: Working principle of GPS.

GPS modules provide accurate position and time information for UAV-based applications [24]. Special GPS modules such as the real-time kinematic (RTK) GPS [25] provide high update frequency and are able to withstand the UAVs' high velocity and maneuverability. But despite their benefits, GPS systems consume a significant amount of power due to the constant synchronization and locking of the GPS signal.

UAVs equipped with GPS modules provide precise positional and temporal information that is used for localization, stabilization, tracking, and navigation. GPS-enabled UAVs are used in [26] for accurate landmine detection. The authors of [27] use a high precision RTK GPS to determine the ground control target locations. In [28], the authors use GPS and camera data along with the hierarchical A* algorithms to determine the best flight path for UAVs. Further, GPS enabled UAVs find applications in precision agriculture to monitor crop health, map agricultural areas, and for cropdusting [29].

A.3.2.2 IMU

IMUs are electronic devices that measure inertial quantities such as acceleration, angular motion, and orientation of an object. Accelerometers, gyroscopes, and magnetometers are common components of IMUs for UAVs. IMUs in UAVs work in tandem with GPS modules to form the inertial navigation system (INS), which is responsible for UAV localization, stabilization, and tracking.

Because IMUs are critical components for maintaining stable flight control and guidance for UAVs, understanding some of their key performance metrics is critical. The quality of the IMU is determined by the performance of its internal components, such as accelerometers, gyroscopes, and magnetometers. The range provided by the accelerometer and gyroscope is one of the most important parameters to consider. Another key parameter is the bias instability property which captures the accumulated sensor bias and unknown drift error in the IMUs over time. Aside from the aforementioned factors, bandwidth, output data rate, temperature sensitivity, dimensions, number of axes and weight are other IMU parameters to consider [30],

Table A.2: Key Performance Metrics for Various On-board Sensors

Sensor	Key metrics	Output format	Interfacing options	Power (W)	Vendor	Cost (USD)	Limitations	References
RGB-D camera	RGB resolution: up to 1920 x 1280 Maximum range: 10 m Depth FoV: 87° x 58° RGB FoV: 69° x 42°	RGB and depth image	SPI, USB, MIPI	0.35 – 3.5	Intel RealSense	200 – 600	Rain, fog, mist, ambient lighting, etc.	[14], [15], [16], [17],
Thermal camera	Thermal resolution: 336 x 256 / 640 x 512 Thermal accuracy: ±5° C Spectral range: 7.5˜13.5 μm	Thermal image	UART, USB, I2C, SPI, SDIO	0.5 – 1.55	FLIR	1500–8000	Fails to distinguish objects when they are at the same temperature	[18], [19], [20], [21], [22]
GPS	Velocity accuracy: 0.1 m/s Horizontal position accuracy (RTK): ≈ 2.5 m Sensitivity (navigation): −160 dBm Time to first fix: 28 s	NMEA, UBX, RTCM	UART, USB, SPI, DDC	-	u-blox, EMLID	-	Increased power consumption, cannot penetrate through solid walls or structures.	[24], [25], [26], [27], [28], [29]
IMU	Gyro bias instability: $\approx 0.05°/hr$ Accelerometer bias instability: $\approx 15\ \mu g$ Data rate: 1 to 1000 Hz	Digital output for gyroscope, accelerometer and magnetometer	RS-422, I2C	5	KVH, InvenSense	-	Drift in the values over time leading to incorrect measurement values	[30], [31], [32], [33], [34]
Ultrasonic sensor	Range: ≈ 10 m Range resolution: ≈ 1 mm Measure angle: 15° Frequency: 30 – 80 kHz	Pulse width, real-time analog voltage envelope, analog voltage output, serial digital output	I2C, RS232, TTL, USB, UART	< 1	XL-MaxSonar, Sparkfun, Marvelmind	10˜60	Wind, acoustic disturbances, etc.	[35], [36], [37], [38], [39], [40], [41], [42], [43], [44]
mmWave FMCW radar	Radial range: ≈ 200 m Range resolution: ≈ 4 cm FoV: 5° – 160°	Raw IF sample (time series data), range profile, velocity profile, angle profile	CAN, CSI-2, I2C, LVDS, QSPI, SPI, UART	1 – 3	Texas Instruments	25˜35	Angle estimation can be error prone	[45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55]
LiDAR	Range: ≈ 100 m Range accuracy: ≈ 3 cm FoV (Horizontal): 360° FoV (Vertical): +15° – −15° LiDAR data points: ≈ 600,000 points/s	3D Point cloud	Ethernet, UDP	8 – 22	Velodyne, Ouster	100˜6000	Rain, fog, mist, etc.	[56], [57], [58], [59], [60]
RFID sensor	Reader antenna ports: 2, 4 Frequency: 860 – 960 MHz Tag memory: 96 bits Tag antenna size: 94 x 24 mm	Digital serial output through USB, Ethernet, etc.	Ethernet, USB	Reader: 0.01 – 1.4	Zebra, Smartrac	-	Affected by metals and liquids, prone to interference and jamming	[61], [62], [63], [64], [65], [66], [67]
UWB sensor	Detection Range: 40 m Frequency: 3.1 – 4.8 GHz Accuracy (LoS): 2.1 cm Max Operating Range (LoS): 300 – 1100 m	Digital serial output through USB, Ethernet, etc.	Ethernet, USB, SPI, CAN, UART, GPIO	2	TDSR, Decawave	-	Co-existence with other technologies, interference issues, etc	[68], [69], [70], [71], [72]

[31], [32].

Attitude determination, localization, and navigation are three of the most important functions of IMUs in UAVs. This is demonstrated in [33], where the authors created a single frequency GPS IMU system to provide real-time information to UAVs to aid in localization, guidance, and navigation. Another application for IMUs is stabilization. In [34], authors developed a method for using IMU data to provide real-time video stabilization of images captured by a fixed camera mounted on a UAV.

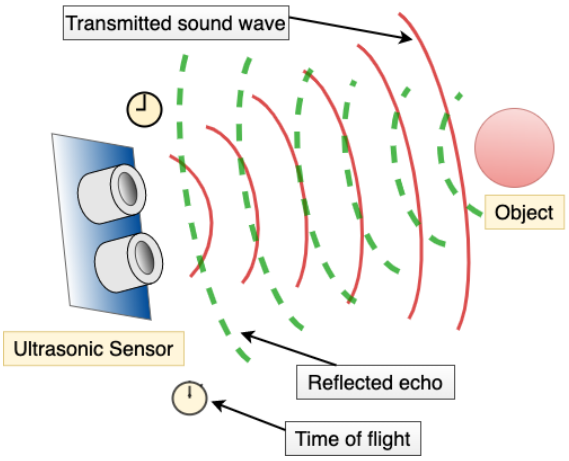


Figure A.7: Operating principle of ultrasonic sensor.

A.3.3 Proximity Sensors

Proximity sensors provide information regarding objects that are placed within a short distance from the sensor. The section that follows discusses ultrasonic sensors, which are one of the most common proximity sensors used in UAVs.

A.3.3.1 Ultrasonic Sensor

Ultrasonic sensors [35], [36] are used in UAVs for target detection, flight navigation, and collision avoidance [37], [38]. They are used in industry for quality control and fault detection [39]. Ultrasonic sensors are widely used in autonomous systems for smart car parking and vehicle detection due to their robust sensing capabilities [40]. Ultrasonic sensors operate by measuring the distance between the sensor and the target object using high-frequency sound waves [41] as shown in Fig. A.7. They are extremely reliable and can detect transparent objects in situations where other vision-based systems may fail. Some of the key performance metrics for ultrasonic sensors can be found in Table A.2.

In addition to providing proper navigation, accurate obstacle detection and timely obstacle avoidance [37], [38], ultrasonic sensors are also used for ensuring safe landing for UAVs. In [42], a sonar-based model is developed which measures the reflected sound waves to determine the suitability of the landing field. If there are obstacles in the landing field that are higher than the UAV legs, the landing is considered unsafe. Precise indoor localization and navigation of UAVs is also achieved by using the ultrasonic system developed by Marvelmind [43]. A network of stationary ultrasonic nodes are placed in an area and connected wirelessly through the license free band. The mobile object carrying another ultrasonic node frequently sends beacons which is captured by these stationary nodes. The propagation delay between the beacons is measured and using trilateration, precise localization is achieved in order of around ± 2 cm. Ivan *et al.* in [44] has demonstrated the use of Marvelmind technology by developing a precise localization system that would enable accurate position estimation for vertical takeoff and landing (VTOL) of UAVs.

Table A.3: Key Performance Metrics for mmWave FMCW Radar

Parameter	Expression
Radial range (d)	$d = f_i c/2S$
Radial velocity (v)	$v = \lambda\Delta\Phi/4\pi T_c$
AoA (θ)	$\theta = \sin^{-1}(\lambda\Delta\Phi/2\pi h)$
Range resolution (d_{res})	$d_{res} = C/2B$
Velocity resolution (v_{res})	$v_{res} = \lambda/2T_f$
AoA resolution (θ_{res})	$\theta_{res} = \lambda/Mh.\cos(\theta)$

f_i - Intermediate frequency; c - Velocity of light; S - Chirp slope; $\Delta\Phi$ - Phase difference between the consecutive chirps; λ - Chirp wavelength; T_c - Time interval between consecutive chirps; h - Spacing between two adjacent receiver antennas; B - RF bandwidth; T_f - Chirp frame time; M - Number of receivers.

In [39], the authors create a prototype ultrasonic inspection system using UAVs that provides information about the structural integrity of an industrial unit.

A.3.4 Radar-based Sensor

Another type of sensor that is proving to be extremely useful for UAVs is radar sensors. Radars have traditionally been used to detect targets by measuring how long it takes an emitted electromagnetic wave to reflect back after striking the target object. Radars were originally used to detect and identify approaching enemy targets in military and defense applications. Their use in self-driving cars has recently demonstrated that they can be used effectively outside of their intended application fields. Radars have since been installed on UAVs to improve their perception and detection capabilities. Due to size and power constraints, the mmWave frequency modulated continuous wave (FMCW) class of radars has shown promise for UAV-based applications. The section that follows provides an overview of these radar sensors, including their advantages, disadvantages, and potential applications.

A.3.4.1 mmWave FMCW Radar

The mmWave FMCW radars [45], [46] are well-known for providing accurate target range and velocity information [47]. Due to their resistance to extreme weather and lighting conditions, mmWave FMCW radars are the obvious choice for UAV-related applications. Furthermore, the combination of their high bandwidth and accurate range and velocity resolution makes them appealing for use in detection and collision avoidance scenarios [48]. mmWave FMCW radars work in the same way that traditional radars do. A frequency modulated continuous chirp waveform is transmitted by the radar and reflected by nearby objects. The reflected chirp is received at the receiver and processed to determine the radial range, velocity, and angle of arrival (AoA) of the target.

As shown in Fig. A.8, A.9 and A.10, the raw IF samples from the mmWave FMCW radar are processed to obtain the range-plot, range-doppler and range-azimuth heatmaps respectively. The plots are directly obtained from the TI mmWave demo visualizer which was used along with the TI AWR1843 mmWave FMCW radar to detect the objects. The range-plot shows that the radar has detected four objects at a radial distance that is less than 3 meters. They are stationary as observed from the range-doppler heatmap, as the higher intensity red colour is situated close to 0 m/s. From the range-azimuth plot, it is inferred that the objects are placed at an angle of 10° from the radar. Some of the key mmWave radar parameters along with their expressions is summarized in Table A.3.

mmWave FMCW radars are used in UAVs for object detection, identification, and tracking due to their excellent sensing capabilities. Detecting and identifying UAVs and birds using the micro-Doppler spectrum obtained from mmWave FMCW radars has been proposed in [49]. Similarly, the work in [50] contributes to the identification of micro-UAVs in low grazing angle scenarios using the mmWave FMCW radar micro-Doppler spectrum. This has broad applications in electronic warfare, where terrain clutter can make detecting micro-UAVs difficult. Additionally, the authors in [51] have used mmWave FMCW radars for localization and activity classification of objects using convolutional neural network (CNN). Similarly, Siddharth *et al.* in [52] has used the range-angle images obtained from mmWave radars along with YOLO and Faster RCNN models to achieve accurate target classification in the range of $87.68\% - 99.7\%$. One of the limitations of mmWave radars is their poor AoA estimation. As seen in [53] and [54], novel machine learning techniques and mechanical rotation of radar in the horizontal direction improves the AoA estimation and field of view (FoV) in both azimuth and vertical directions. This method has the advantage of providing accurate angle estimation while only requiring one transmitter and receiver. In addition, the authors of [54] introduced techniques for estimating the height and angle of UAVs from the GCS using mmWave FMCW radars.

A.3.5 LiDAR

LiDAR is a sophisticated remote sensing technology used to generate 3D maps of the environment. The LiDAR system works by beaming a large number of lasers onto a surface. The beam's wavelength is typically in the optical, infrared, or ultraviolet range. The reflected beam from the object in focus is captured by a laser scanner, and the time of flight between the transmitted and reflected beams is measured and used to calculate the source's distance from the LiDAR. This distance calculation process is repeated several times to produce a complex image map (3D point cloud) of the scanned surface, as shown in Fig. A.11.

One of the most important characteristics of LiDARs is the range over which the laser beam can be focused. The power of the laser beam is restricted to conform to eye safety regulations, which in turn, limits the maximum detectable range of LiDARs. Other factors such as laser type and focal length of lens also impact on

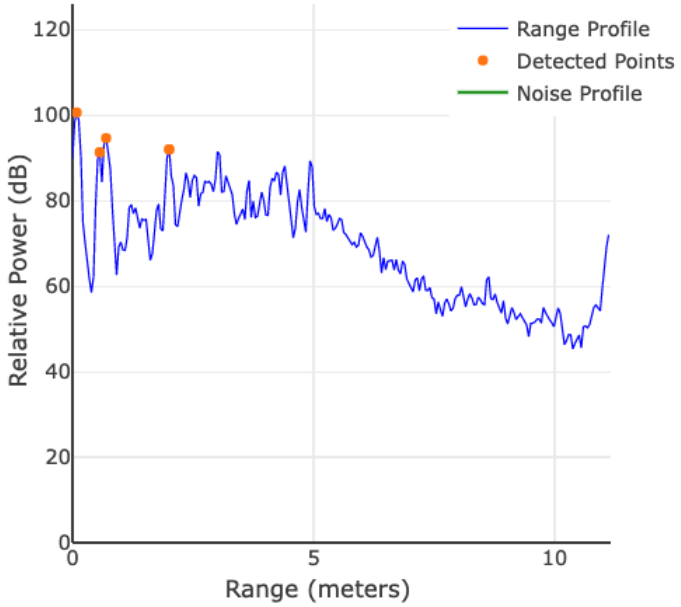


Figure A.8: Range plot.

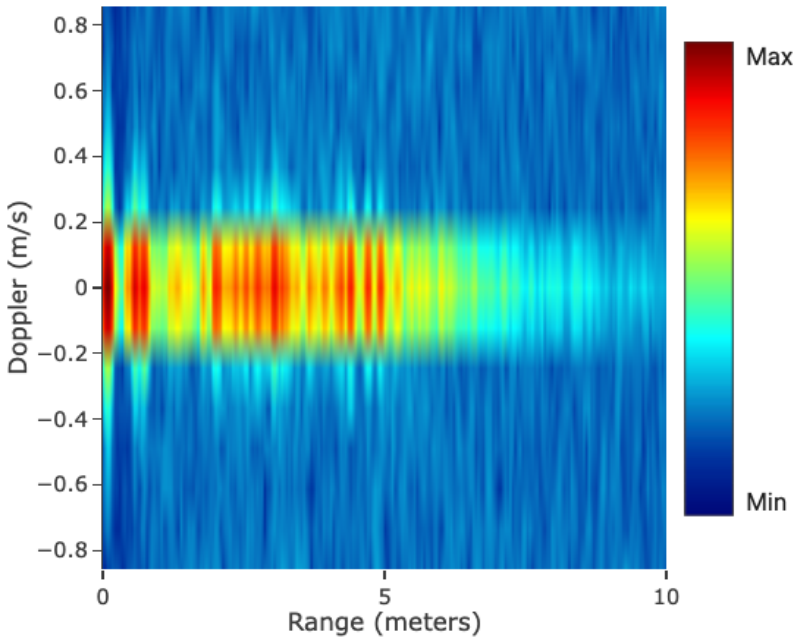


Figure A.9: Range-doppler heatmap.

LiDAR range. Another critical parameter for LiDARs is the density of the LiDAR point cloud which determines the resolution of the images obtained after LiDAR scanning. Other parameters such as range resolution, scan rate, dimension, FoV, and weight are also considered based on application requirements [56], [57].

UAVs equipped with LiDAR modules are used in agriculture, mining, forestry and civil engineering to name a few. A LiDAR-based UAV system is used in [58] for classification of forest vegetation and structure measurements. The authors represent the area's vegetation and topography gradient using a LiDAR-hyperspectral image fusion method. LiDARs are also used to map coastal areas as seen in [59].

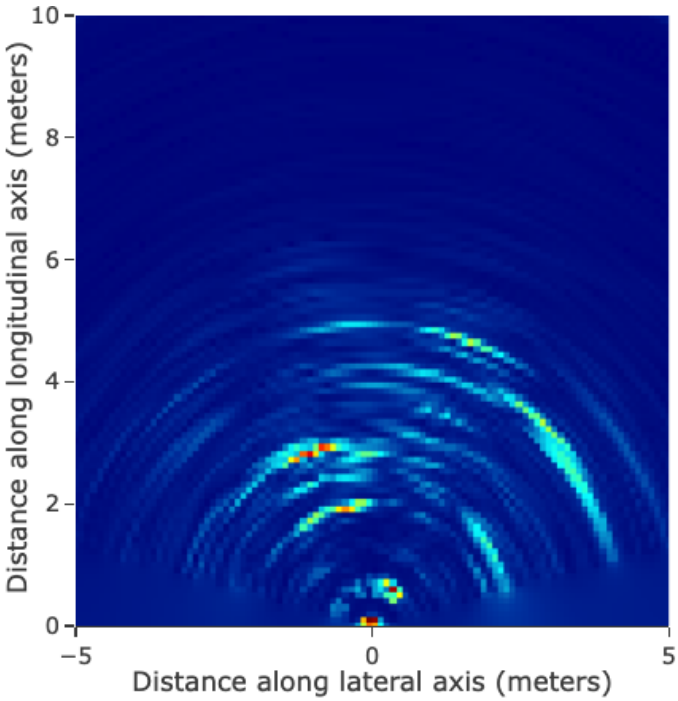


Figure A.10: Range-azimuth heatmap.

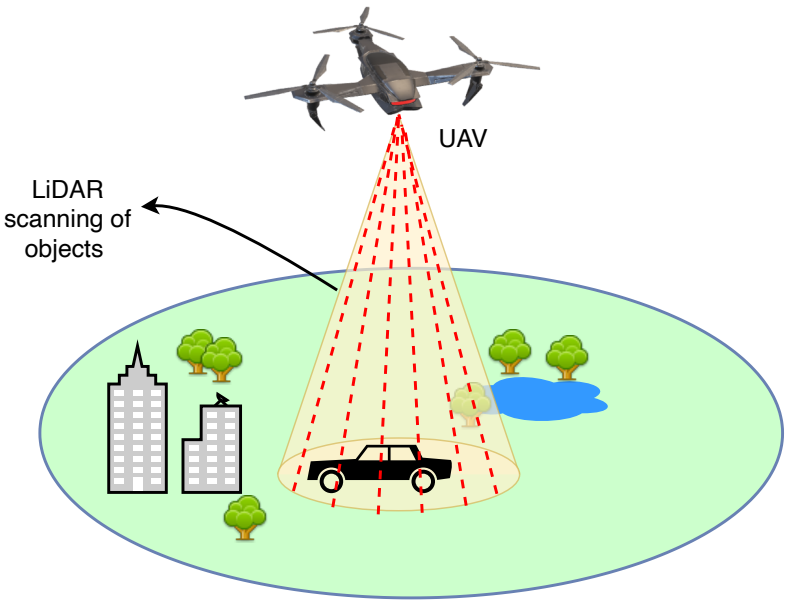


Figure A.11: LiDAR scanning.

LiDAR images provide more information for monitoring shoreline changes. LiDARs are also used to provide navigation capabilities for UAVs in GPS-denied environments for localization and collision avoidance [60].

A.3.6 Radio-Frequency Identification (RFID) Sensor

The RFID is a contactless wireless system that uses radio frequency waves to track and identify objects. The RFID system primarily consists of a RFID reader and RFID tags. RFID tags are placed on objects that require tracking or identification.

The RFID reader transmit radio frequency waves which interact with these attached RFID tags to obtain useful information about the respective object. RFID technology finds applications in a number of diverse areas such as inventory management, supply chain, security, pharmaceutical, transport and airline industry. RFIDs are equipped in UAVs for identifying and tracking objects. Additionally, they are also used for UAV localization [61].

The criteria to determine the best RFID system for a particular UAV application requires the knowledge of key parameters such as operating frequency, power requirements and environmental conditions. RFID systems can operate in low frequency (LF), high frequency (HF), and ultra-high frequency (UHF) ranges. RFID for UAVs primarily uses the UHF (865˘960 MHz) that provides read range of around $4-6$ meters or more depending upon the type of tags used [62]. Depending upon the application use case, the RFID tags employed can be passive or active. The passive tags does not require any external power source to operate whereas the active tags are equipped with a limited battery source to ensure its operation. Environmental factors include the location of the tag, the material on which the tag is attached, temperature, pressure, vibration sensitivity, etc. Other important include communication interface for RFID readers, antennae ports, read accuracy, data rate, cost, etc., [63], [64], [65].

Traditional applications of RFID systems are enhanced using UAV systems. This is evident in RFly [61], where the authors have developed a UAV-based RFID relay for improving the read range of an RFID system. Additionally, they have also developed an RF based localization algorithm for localizing non-line-of-sight (NLoS) objects. Another system in [66], uses passive RFID tags to estimate the 6 degrees-of-freedom (DoF) pose of a controller. The pose is estimated by utilizing a singular value decomposition (SVD)-based approach that uses the position of the tags with respect to the controller. This method provides seamless UAV navigation in indoor environments. In [67], Zhang *et al.* developed a RFID enhanced UAV system that estimates the precise pose of the UAV. The system works by utilizing the phase measurements from the different RFID tags attached to the UAV to determine the 6-DoF pose of the UAV system.

A.3.7 Ultra-Wideband (UWB) Sensor

UWB sensors uses short-range radio frequency pulses to determine location of nearby objects. The operating principle is more or less similar to the RFID technology, where the UWB transmitter sends billions of pulses over a wide frequency spectrum which is captured by the UWB receiver to determine the location of the target. UWB primarily uses two techniques for positioning, the time difference of arival (TDoA) approach and the two way ranging (TWR) method. In TDoA, UWB sensors are placed at multiple locations in an indoor space. The moving UWB tag continuously sends signals which is captured by these different UWB tags. Subsequently, by using multilateration, the location of the moving tag is accurately determined. In TWR, when two devices using UWB come close to each other, they start ranging to

determine their distance. The time taken to transmit and receive the signal is used to calculate the distance between the two objects [68].

UWB sensors provide highly accurate distance measurements (in order of 2 cm) [69], [70]. As the time duration of the pulses gets narrower the accuracy of the measurement is improved. Moreover, UWB is less affected by multipath interference and hence it is preferred in highly crowded environments to achieve reliable positioning. However, as UWB uses a wide range of frequencies, co-existence of UWB with other technologies is a challenge. Additionally, since UWB requires an always ON sensing strategy to capture the transmitted signal, power consumption is more as compared to the RFID sensors.

In UAVs, UWB sensors are primarily used for localization purposes in GPS-denied environments. In [71], the authors propose a target-relative tracking and positioning method using UWB sensors. To estimate the speed of the target, initially the UWB range measurements are fused with other on-board sensor data using an extended kalman filter (EKF). Next, the target orientation is transferred to the quadcopter by using UWB-based communication. The experiment results demonstrate the autonomous capability of the proposed approach to relatively position the quadcopter with respect to the target. Additionally, UWB sensors in UAV can also help in autonomous docking. In [72], the authors have developed UWB-vision approach to facilitate the autonomous landing of quadcopters in GPS-denied environments. The system initially uses a combination of distance and relative displacement measurements to approach the landing zone. Once the landing pad is detected, the measurements from UWB and vision-based systems are fused to provide an accurate landing position.

A.4 Communication Modules

UAVs must constantly update the GCS about their position, battery health, on-board sensor data and other critical information in order to enable and maintain seamless autonomous operations. The requirement for telemetry or sensed data transfer necessitates the use of efficient and reliable communication technology for UAVs. The following section provides insight into some of the characteristics associated with each communication protocol so that the user can make an informed decision about the choice of communication module to equip on board UAVs. Furthermore, Table A.4, highlights the key attributes of various communication protocols used in UAVs.

A.4.1 LoRa

LoRa is a low-power, long-range communication technology that is primarily used in the internet-of-things (IoT) applications [73]. The technology is developed by Semtech corporation. The protocol is designed and implemented in such a way the LoRa physical layer employs a proprietary chirp spread spectrum (CSS) modulation [74], while its MAC layer known as LoRaWAN is open source and maintained by

the LoRa Alliance [75]. LoRa transmits over unlicensed bands at frequencies of 433 MHz, 868 MHz, 915 MHz, and 923 MHz with an approximate maximum range of about 10 to 15 km. One of the main limitations of using LoRa in UAVs is its data rate, which can reach only upto a maximum of 50 kbps [76].

A.4.2 Bluetooth Low Energy (BLE)

BLE or Bluetooth Smart is an enhanced version of the classic Bluetooth technology that is designed for low-power, short-range application demands [77]. The Bluetooth Special Interest Group (SIG) designed and developed the protocol in order to provide low-power solutions for applications such as health care, beacons, and fitness. Furthermore, BLE has a data rate of 1 Mbps and range of about 50 meters, similar to traditional Bluetooth. The protocol supports a variety of topologies such as star, mesh, p2p, and broadcast. BLE is not backward compatible with its predecessor, however it uses the same frequency as traditional Bluetooth, 2.4 GHz to 2.48 GHz. Because of its low power consumption and high data rate, BLE is a promising technology for use in UAVs [13], [78].

A.4.3 Wi-Fi

Wi-Fi is a widely used short-range communication protocol found in laptops, tablets, smartphones, digital televisions, and other devices. Wi-Fi is based on the IEEE 802.11 protocol stack and comes in a variety of versions with varying levels of power consumption, data rate, and bandwidth. Wi-Fi operates in the unlicensed spectrum band of frequencies, specifically 2.4 GHz and 5 GHz [79]. Wi-Fi employs the orthogonal frequency division multiplexing (OFDM) modulation, which is responsible for its high data rate and resistance to interference issues. These properties make Wi-Fi an ideal choice for use in UAVs. Furthermore, because of its modular design, it can be deployed in both infrastructure and ad-hoc modes. In [80], Wi-Fi is set up in an ad-hoc manner to be used with UAV relay networks.

A.4.4 Long-Term Evolution for Machine-Type Communication (LTE-M)

LTE-M is a low-power wide-area communication standard developed by the 3GPP to support machine-to-machine communications and IoT applications. [81]. The protocol provides a high data rate as well as increased bandwidth. It operates within 3GPP specified licensed spectrum band. The increased adoption of LTE-M protocol in UAV applications can be attributed to its design architecture which allow seamless integration with existing cellular infrastructure. Furthermore, its long range, low latency, resistance to interference, and weather conditions have all contributed to its growing popularity

Table A.4: Key Performance Metrics for Communication Technologies

Protocol	Range (km)	Throughput (Mbps)	Power (W)	Frequency (GHz)	Topology
BLE	≈ 0.1	0.125 - 1.36	0.01 - 0.5	2.4	Star, Mesh, Broadcast, P2P
LTE-M	≈ 10	≈ 1	0.1 - 0.2	LTE bands	Star
Wi-Fi	≈ 0.1	$\approx 10^4$	≈ 2	2.4, 5	Star, Mesh
LoRa	≈ 10	0.01 - 0.05	≈ 0.025	0.433, 0.868, 0.915, 0.923	Star

A.5 Computing Platforms

Computing platforms are on-board hardware and software modules that facilitate the integration of various sensor and communication technologies to ensure safe and secure operation of UAVs. The most important component of a UAV computing platform is the flight controller. The flight controller is in charge of interpreting data from on-board sensors in order to facilitate real-time decision making. Flight controllers are also in charge of telemetry, communication with the GCS, power management, and other duties. In addition to flight controllers, the computing platforms also responsible for executing object detection tasks, collision avoidance algorithms, UAV stabilization schemes, and control algorithms. The sections that follow describe the available hardware and software platforms for implementing flight controller systems and other necessary algorithms on UAVs.

A.5.1 Hardware Platforms

UAV hardware platforms are embedded processing units that implement flight controller capabilities. The hardware flight controller systems are in charge of controlling the UAV's altitude and mobility, avoiding collisions, interacting with other sensing and communication modules, controlling and stabilizing the UAV, navigation, and so on. Because of the numerous tasks that these units must perform, the performance of these platforms is primarily determined by the type of processor, speed, and computational memory. When selecting a flight controller module for UAV applications, other factors such as form factor and power consumption are taken into account. Furthermore, these hardware components aid in the execution of sophisticated algorithms required to meet the demands of the UAV applications. The most recent widely used UAV hardware platforms are presented in this section, along with some key metrics such as processor specifications, available memory, form factor, supported operating system, and power consumption.

A.5.1.1 FPGA-based Platforms

***a) OcPoC-Zynq Mini*:** The OcPoC-Zynq Mini [82] is a fully programmable

FPGA+ARM-based system-on-chip (SoC) that is developed by the Aerotenna company. The on-board Artix-7 FPGA enables the seamless integration and interfacing of various sensor and hardware modules. Additionally, the coupled dual-core ARM A9 processor increases the computational capability and input/output (I/O) flexibility. The module comes with on-board IMU with 9-DoF and a high-resolution barometer that is used to measure the atmospheric pressure. The hardware board is also capable of interfacing devices using the following protocols: SPI, CAN, I2C, USB-OTG, and USB-UART. The OcPoC-Zynq Mini is lightweight and has a small form factor, making it an appealing option for use with UAVs [83].

A.5.1.2 ARM-based Platforms

a) Pixhawk 4: Pixhawk 4 [84] is the latest flight controller board designed and developed by Holybro and Auterion [85]. The board is optimized to run the latest PX4 autopilot software stack and comes with advanced features which makes it flexible and reliable for autonomous operations. The board is equipped with a STM32F765 processor which consists of a powerful 32-bit ARM Corex M7 chip. It additionally houses a STM32F100 32-bit ARM Cortex M3 processor dedicated to handle the I/O operations between the board and the various on-board peripherals. The various on-board sensors equipped on the board include gyroscope, accelerometer, magnetometer and barometer. Other peripherals include an on-board u-blox Neo-M8N GPS/GLONASS receiver that aids in the acquisition of better positional information [86]. In order to support real-time operations, the Pixhawk 4 uses the NuttX operating system. Furthermore, the multi-threaded capabilities of NuttX allow for Linux/Unix programming of the flight controller [87].

b) Pixhawk 4 Mini: The Pixhawk 4 Mini [88] flight controller board is intended for use with smaller UAVs commonly used by hobbyists and researchers. With the exception of a dedicated I/O processor unit, the hardware configuration of the Pixhawk 4 Mini is similar to that of the Pixhawk 4 [89]. The Pixhawk 4 Mini also includes the STM32F765 processor, as well as an accelerometer, gyroscope, magnetometer, and barometer. It has a smaller dimensional form factor than the Pixhawk 4, which is detailed in Table A.5. Pixhawk 4 Mini also uses the NuttX operating system for real-time operations.

c) BeagleBone Blue: The BeagleBone Blue [90], [91] is a miniaturized Linux-based system that can be used as a flight controller in UAVs and autonomous vehicles. The Octavo OSD3358 board features an ARM Cortex-A8 processor and 512 MB DDR3 RAM. It is equipped with a NEON floating-point accelerator for performing complex digital signal processing computations. The board can be programmed in Linux using a variety of supported softwares such as MATLAB, Python, ROS and ArduPilot, making it ideal for developers and hobbyists looking to quickly prototype their applications. Bluetooth 4.1, BLE, and Wi-Fi are among the integrated communication modules used to ensure the reliable transmission of sensor and telemetry data. In addition, the board includes built-in sensors such as an IMU, a barometer,

Table A.5: Commonly Used UAV Flight Controller Boards

Platform	Processor	On-board sensors	Available memory	Power (W)	Supported interfaces	Form factor (cm)	Weight (g)	References
OcPoC-Zynq Mini	CPU: ARM-A9 dual-core FPGA: Artix-7	IMU, Barometer	512 MB DDR3	≈ 3	I2C, USB-OTG, USB-UART, SPI, CSI, GSI, CAN	9.2×6.3	25˜35	[82], [83]
Pixhawk 4	CPU: ARM Cortex-M7 IO Processor: ARM Cortex-M3	Accelerometer, Magnetometer, Barometer, GPS	512 KB	-	UART, I2C, SPI, CAN, PWM, R/C	4.4×8.4	≈ 33	[84], [85], [86], [87]
Pixhawk 4 Mini	CPU: ARM Cortex-M7	Accelerometer, Magnetometer, Barometer, GPS	512 KB	-	UART, I2C, SPI, CAN, PWM, R/C	3.8×5.5	≈ 37	[88], [89]
BeagleBone Blue	CPU: ARM Cortex-A8	IMU, Barometer, Thermometer	512 MB DDR3	-	USB 2.0, UART, SPI, I2C, GPIO	17.5×11.2	≈ 36	[90], [91], [92]
Raspberry Pi 4 Model B	CPU: ARM Cortex-A72	-	1/2/4/8 GB DDR4	-	USB 2.0, USB 3.0, UART, CSI, DSI, GPIO	8.5×5.6	≈ 66	[93]
Nvidia Jetson Nano	CPU: ARM Cortex-A57 quad-core GPU: Maxwell 128-core	-	4 GB DDR4	$5-10$	USB 2.0, UART, SPI, I2C, I2S, GPIO	6.96×4.5	≈ 17	[94], [95]
Nvidia Jetson TX2	CPU: Nvidia Denver 2 64-bit dual-core and ARM Cortex-A57 quad-core GPU: Pascal 256-core with 256 CUDA cores	-	8 GB DDR4	7.5−15	USB 3.0, USB 2.0, UART, SPI, I2C, I2S, GPIO, CAN	5.0×8.7	≈ 88	[95], [96], [97], [98]
Nvidia Jetson AGX Xavier	CPU: ARM v8.2 Carmel 64-bit 8-core GPU: Volta 512-core with 64 Tensor cores	-	32 GB DDR4	$10-30$	USB 3.0, UART, SPI, CAN, I2C, I2S, DMIC, DSPK, GPIO	10.0×8.7	≈ 280	[95], [99], [100]
Arduino Mega 2560	CPU: ATmega2560	-	256 KB flash	-	USB 2.0, UART, SPI, I2C, GPIO	10.15×5.33	≈ 37	[101], [102], [103]
Intel UP boards	CPU: Intel Atom x5-z8350 quad-core FPGA: Intel FPGA Altera Max V	-	1/2/4 GB DDR3L	≈ 13	USB 2.0, USB 3.0, UART, GPIO, CSI, I2C, I2S	8.56×5.65	≈ 98	[104], [105], [106]

and a thermometer [92].

e) Raspberry Pi 4 Model B: The Raspberry Pi 4 Model B is the latest version in the Raspberry Pi series with a quad-core ARM Cortex-A72 processor and $2, 4$ or 8 GB of RAM depending upon the requirements. The board comes with Bluetooth 5.0, BLE, Gigabit Ethernet and 2.4/5.0 GHz wireless LAN support for connectivity. The Pi 4 board has a 40 pin GPIO header with support for USB 2.0, USB 3.0, CSI, and DSI ports. As the wireless LAN and Bluetooth are compliance certified, the Raspberry Pi 4 can easily be used in UAVs and autonomous applications [93].

A.5.1.3 GPU-based Platforms

Hardware boards which are integrated with GPUs have recently become popular due to the deployment of deep learning algorithms on embedded platforms. This section lists some of the popular GPU-based platforms which can be used on UAVs for enabling machine learning based operations.

a) Nvidia Jetson Nano: The Nvidia Jetson Nano is a low form factor board intended to be used with small-sized autonomous vehicles. The board consists of a 128-core Maxwell architecture GPU and quad-core ARM Cortex A5 CPU [94]. The Nano comes with 4 GB RAM and supports Gigabit Ethernet, USB 3.0, USB 2.0 Micro-B, GPIO, I2C, I2S, SPI, UART interfaces for connecting with various peripherals. Jetson Nano is reported to provide an AI performance of approximately 472 GFLOPS to accelerate deep learning frameworks [95].

b) Nvidia Jetson TX2: A slightly higher end version of the Jetson Nano, the Nvidia Jetson TX2 gives more performance for computer vision and deep learning applications. The Jetson TX2 board comes with a dual-core 64-bit Nvidia Denver 2 CPU [96]. The GPU features a 256-core Pascal architecture with 256 CUDA cores. The board has 8 GB RAM and supports USB 3.0, USB 2.0, UART, SPI, I2C, I2S, GPIO and CAN interfaces for peripheral connections [97]. The TX2 is reported to provide 1.33 TFLOPs of AI performance as compared to the Jetson Nano [95] and hence the TX2 can be used for slightly higher end and demanding applications. In [98], the authors demonstrate a prototype system that uses the computing capabilities of the Jetson TX2 to run the YOLOv3 algorithm for UAV surveillance in airports.

c) Nvidia Jetson AGX Xavier: The Jetson AGX Xavier board is primarily developed to integrate machine learning and deep learning algorithms for various autonomous applications. The board is based on a 512-core Volta architecture GPU with 64 tensor cores and a 64-bit 8-core Carmel ARM architecture CPU [99]. It has 32 GB RAM memory and is integrated with a dedicated deep learning accelerator, vision accelerator and encoder/decoder units for various computer vision tasks. For interfacing with peripherals, the Jetson AGX Xavier supports RJ45, USB-C, USB 2.0, UART, etc. In terms of performance, the Jetson AGX Xavier is capable of accelerating deep learning algorithms in the order of 32 TFLOPS [100], [95].

A.5.1.4 Atmel-based Platforms

a) Arduino Mega 2560 R3: The Arduino Mega 2560 board [101] is based on the ATmega2560 microcontroller. It has digital I/O and analog input pins for interfacing with peripherals. Out of the 54 digital I/O pins, 15 can be used for PWM output. Additionally, the board also supports UART, I2C and USB connections [102]. As seen in [103], the Arduino Mega 2560 is used to study adaptive control of a quadcopter by using a serial-stage proportional-integral-derivative (PID) controller. The design utilizes the Arduino board to act as the core control board. The board in turn is interfaced with various sensors such as infrared sensor, bluetooth module,

gyroscope and laser ranging sensor.

A.5.1.5 Intel-based Platforms

a) Intel UP board series: The Intel UP boards are Raspberry Pi 2 sized boards that are primarily developed for robotics, UAV, smart home, IoT and digital signage applications [104]. The board is powered by an Intel Atom x5-z8350 quad-core processor along with the Intel FPGA Altera Max V. It can have 1/2/4 GB DDR3L RAM of memory and supports interfacing options such as USB 2.0, USB 3.0, UART, CSI, DSI, and GPIO [105]. Although the board doesn't have dedicated on-board sensors, it is powerful to run Linux, Android and even Windows 10 operating system. The powerful CPU performance can be seen in [106], where the Intel UP boards are used as companion computers to power the TF Mini LiDAR for height estimation.

A.5.2 Software Platforms

In order to interact and maintain proper functionality of the various hardware modules, flight controller software modules are deployed on to UAV computing platforms. The flight controller software stack consists of control and localization algorithms, stabilization techniques, navigation strategies, and other services that enable UAV operations to run smoothly. There are numerous flight controller software packages on the market today. To name a few, selecting the right flight controller software for UAVs would entail taking into account several factors such as UAV application requirements, communication constraints, power consumption, and ease of interacting with other hardware components. The section that follows provides a brief overview of some of the most popular open source flight controller software that is used for commercial and research purposes

A.5.2.1 ArduPilot

ArduPilot is a trustworthy open source flight controller software that can control UAVs like gliders, conventional and VTOL planes, multirotors, and helicopters [107]. ArduPilot which was started by hobbyists in 2009, is now used for commercial and research purposes. It supports a wide range of hardware platforms, including Maverick, Raspberry Pi and Odroid to name a few [108]. The ArduPilot software stack includes control algorithms for UAV localization and navigation. GCS software is also included with ArduPilot, which can help with vehicle configuration, mission planning, and testing. Additionally, the software supports RTK GPS, magnetometers, barometers, airspeed sensors, brushless motors, actuators, and gimbals. ArduPilot is licensed under GPL Version 3. It is free to download and use. Further, it has a rich documentation and source code which can be found in [107] and [109] respectively.

A.5.2.2 PX4

PX4 provides adaptable tools and algorithms to assist UAVs in autonomous navigation. The Dronecode foundation hosts the software, which is distributed under the BSD license [110]. Its modular architecture, configurability, and license permissions make it a viable commercial option. The documentation and source code can be found on GitHub [111].

A.6 Machine Learning Techniques

Unlike static terrestrial systems, UAVs are dynamic and move in three dimensions which makes UAV detection, localization, control, and communication operations challenging. Furthermore, to ensure efficient and reliable UAV operations, a large number of variables such as altitude, speed, and power needs to be optimized. This has put a limit on how far traditional techniques can progress. Recently, machine learning algorithms have shown promise in resolving some of these complex issues. These algorithms are highly scalable and adaptable to a wide range of variables, making them an appealing option for UAV networks. Machine learning algorithms can also retain relevant previous information, which can help with successful UAV operations decision making. Furthermore, because of their inherent data analysis and prediction capabilities, these algorithms are capable of changing the real-time dynamics of UAV networks with ease, which has contributed to their popularity. Control, navigation, detection, collision avoidance, interference management, sensor fusion, computer vision, and communication are just a few of the fields in which the algorithms can be used in UAVs. There is a large body of literature on the use of machine learning in a subset of these fields. A comprehensive analysis and review of the use of machine learning for sensing and communication, on the other hand, is limited. As a result, in this section, the major contributions of machine learning are summarized from a sensing and communication application standpoint, as we discovered that this approach is most beneficial for a novice reader to quickly ramp up on UAV-based machine learning algorithms.

A.6.1 Sensor Fusion

Because a single sensor cannot provide perfect sensing capability, the fusion of sensors with different modalities has demonstrated superior performance. Traditional sensor fusion techniques extract features from each sensor modality separately and then combine them to produce meaningful information. The evolution of deep learning algorithms has added a new dimension to this space by automating feature extraction from input data-sets. This automatic feature extraction in deep learning algorithms has accelerated the rate of sensor fusion to unprecedented levels, allowing us to envision full-scale autonomous operations [112], [113].

Sensor fusion techniques can be divided into three categories. Depending on the stage at which the feature processing and fusion is performed, there are three

types: (1) early fusion, (2) deep fusion, and (3) late fusion. [114]. Early fusion is accomplished by fusing the various sensor modalities directly at the input. This enables greater cross-modal interaction but necessitates extreme caution in terms of data alignment, synchronization, and input data format compatibility. Deep fusion, as opposed to early fusion, operates at an intermediate level, where the features of the various sensor modalities are combined at a halfway point after necessary processing and feature extraction. However, deep fusion is difficult to design and implement. Finally, late fusion employs separate networks to independently process and extract features from each modality. The extracted features are combined towards the end of the processing chain to produce a fused output. Late fusion is simpler to design and can easily handle alignment, synchronization, and data redundancy.

The vast majority of sensor fusion literature is focused on object detection applications. In [115], the RetinaNet architecture is used to fuse 2D camera images and sparse radar data using a novel deep learning framework called CameraRadarFusionNet. The model is intended to intelligently determine the level at which sensor fusion occurs so as to obtain improved 2D object detection performance. The proposed approach shows better performance as compared to the baseline image network by approximately 10%. This can be further improved by feeding the raw radar detections through a noise filter in order to reduce unwanted detections and improve performance. Further, with this approach the improved performance comes with additional latency due to data processing of the radar projections. The authors in [116], use a different approach to improve 3D object detection, a middle fusion center point method. A center point detection network is used to detect the center point of the objects in the RGB images. Based on these center points, a 3D frustum is created to include the radar detections in the image plane. The associated radar detections are then used to generate radar feature maps which can complement the image-based features. The fused feature map can then be used to accurately determine additional object properties such as depth, rotation, and velocity. However, it should be noted that in this approach the frustum is created based on the depth estimated from images. This can lead to include the nearby objects in the generated frustum and hence should be carefully considered.

The combination of LiDAR point clouds and camera images is also used for sensor fusion. On this front, the work by Danfei *et al.* [117], has gained popularity due to its application agnostic nature. The raw point cloud obtained from LiDAR modules is fed into a PointNet architecture, which produces multiple 3D box hypotheses with the input 3D points acting as spatial anchors. The network learns to predict the best hypothesis, which is then combined with CNN-processed image features to detect objects. The approach offers a simple design with no environment and sensor specific assumptions while guaranteeing state-of-the-art performance. However, the algorithm can fail if the number of points are below the recommended threshold. Additionally, issues can also result due to partially visible objects. Another work by Pang *et al.*, [114], uses a low complexity object detection framework to fuse output from camera and LiDAR data using a 2D CNN. The approach used is a late fusion technique that operates on the fused output candidates prior to non-

maximum suppression (NMS). Thus, the semantic and geometric properties of the output can be used to produce more meaningful and accurate results. According to the authors, the results obtained demonstrated high performance in the KITTI benchmark as well as the best performance for long distance object detection. Zhao *et al.* have used the fusion of 3D LiDAR and camera data to perform object detection [118]. The authors employ 3D LiDAR to generate object-region proposals, which are then mapped on to the camera image. The new superimposed image is then used to generate regions-of-interest (ROI) proposals, which are then fed into CNNs to detect objects. The proposed approach has an average processing time of about 66.79 ms making it ideal for real-time operations. However, the proposed approach under performs while trying to detect tiny objects that is beyond 60 m of the LiDAR scanning range. Hence, the future work can employ mmWave radars to improve the detection capability by generating more accurate object-region proposals. LiDAR and camera fusion is also utilized for tracking purposes. In [119], full surround online multi-object tracking (MOT) framework is implemented using a LiDAR and calibrated camera array fusion technique. The tracking problem is formulated using markov decision processes (MDPs) which treats the target appearance/disappearance as state transitions within the MDP. The framework is modular and can support various sensor modalities to improve localization and tracking of objects in 3D. In [120], the authors utilize PointFusion [117] and VoxelFusion for camera and LiDAR fusion by leveraging the VoxelNet architecture. This research focuses on resolving the interfacing problem that exists between highly sparse LiDAR point clouds and region proposal networks (RPN). PointFusion involves mapping of 3D points on to the image plane. VoxelFusion divides the point cloud into equally spaced 3D voxels and then encodes groups of points to each of these voxels based on where they reside. The concatenated output from PointFusion or encoded points from VoxelFusion is then fed into a novel voxel feature encoding layer (VFE) and then used for detection. As a result, this method greatly simplifies feature extraction and bounding box prediction and provides an end-to-end trainable deep network in a single stage. Other LiDAR-based sensor fusion techniques include the TransFuser [121], which investigates the limitations of geometric sensor fusion in dealing with complex scenarios and uncontrolled traffic situations. In this work, the authors create a multimodal transformer to combine the LiDAR birds eye representation with image data. The fusion is based on an attention-based approach that captures the entire global 3D scene with a focus on dynamic objects, greatly improving detection performance. However, this approach under performs in red light conditions and hence requires further improvement. The attention-based approach is also used in [122], where the authors perform 3D object detection of LiDAR bird's eye view (BEV) representations with camera images using a gated feature fusion. Using an auto-calibrated projection mechanism, the 2D camera features are used in the first stage to create a smooth spatial feature map with high correlation with the corresponding LiDAR points. In the second stage, a gated feature fusion network is used to combine these spatial attention maps for camera and LiDAR data fusion based on region. Following this, the camera-LiDAR fusion is achieved using a subse-

quent proposal refinement stage. The proposed method shows significant gain when used with the KITTI and nuScenes datasets. It should be noted that this approach uses a two-stage training method which increases training time. Moreover the data augmentation has to be performed carefully without adding distortion. In [123], a LiDAR and vision-based sensor fusion technique is discussed to achieve reliable object classification with minimal loss. The authors employ an innovative method of upsampling LiDAR point clouds to produce pixel-level depth information, which is then associated with the corresponding RGB data points. For object classification, the fused image is now fed into a CNN. However, it should be noted that the mapping of the upsampled LiDAR feature maps to each of the pixels in the RGB leads to increased processing time and hence can be reduced.

Sensor fusion algorithms help to ensure reliable pose estimation in addition to object detection, tracking, and classification. This is especially beneficial for UAV-based systems, as accurate pose estimation allows for better tracking and collision avoidance operations. One such notable work includes the DenseFusion [124], where the authors fuse the RGB images along with the depth images to implement a reliable 6-DoF pose estimation system. The model employs a heterogeneous architecture in which two input data sources are fused using a novel dense fusion method, after which the pixel-wise embedded features are extracted and used for accurate pose estimation. In addition, an iterative pose refinement algorithm is used to improve the system so that it can handle real-time inference. The proposed approach outperforms the state-of-the-art in terms of better pose estimation, robustness towards occlusions and reduced runtime. In [125], RGB image is combined with depth information in an early fusion to output low-dimensional latent features. This is fed into a deep neural network to perform pixel-wise semantic segmentation for scene understanding. The proposed approach is able to demonstrate increased success rate in both static navigation tasks and dynamic traffic. One of the limitation of the proposed approach is its inability to utilize ego-speed as an input modality. Adding ego-speed can cause the inertia problem which can render the agent unable to restart after it stops for obstacle avoidance. Table A.6 summarizes the above mentioned sensor fusion algorithms which can be tailored to be used with UAV-based applications.

A.6.2 Communication

UAV-based communication systems faces a number of challenges due to the three-dimensional motion, constantly changing channel model, frequently varying orientation and limited energy source associated with UAVs. Moreover, interference issues, cyber-physical attacks, coexistence with existing cellular infrastructure, spectrum sharing, reliable message routing and inclement weather are other challenges for UAV-based communication. As UAV-based communication systems are inherently complex, machine learning algorithms have been found to provide the necessary boost to enable and ensure the various communication requirements for UAVs. These algorithms can accommodate a large number of variables and outperform

Table A.6: Sensor Fusion Techniques

Year	Method	Sensors modalities	Architecture	Fusion level	Application	References
2021	TransFuser	Camera image, LiDAR BEV representation	Transformer with gated recurrent units	Late	3D object detection, Motion forecasting	[121]
2021	Fusion of RGB and depth images	RGB image, Depth image	ResNet architecture	Early	Scene understanding	[125]
2020	CenterFusion	Camera image, Radar point cloud	CNN, Frustum association	Middle	3D object detection	[116]
2020	3D-CVF	Camera image, LiDAR BEV representation	Adaptive gated feature fusion network	-	3D object detection	[122]
2020	CLOC fusion network	Camera image, LiDAR point cloud	2D CNN	Late	3D object detection	[114]
2020	CameraRadar-FusionNet	Camera image, Radar point cloud	RetinaNet, VGG architecture	Early, Late, Halfway	2D object detection	[115]
2020	Fusion of 3D LiDAR and camera data	Camera image, 3D LiDAR data	Region proposal generation, VGG architecture	-	3D object detection	[118]
2019	VoxelFusion	RGB image, LiDAR point cloud	Faster RCNN, VoxelNet	Late	3D object detection	[120]
2019	DenseFusion	RGB image, Depth image	CNN, PointNet architecture	-	6D pose estimation	[124]
2019	Online MOT	Full-surround camera images, LiDAR point clouds	MDP with support vector machines	Early	Multi-object tracking and detection	[119]
2018	PointFusion	Camera image, LiDAR point cloud	CNN, PointNet architecture	Early	3D object detection	[117]
2018	CNN-based fusion of vision and LiDAR	Camera image, LiDAR data	AlexNet	-	Object classification	[123]

traditional algorithms in terms of performance. Furthermore, the algorithms are easily scalable, can be tuned for low complexity, and can predict future states, allowing the UAV system to adapt to changing conditions. Some recent contributions to machine learning algorithms used for UAV-based communication are discussed in the following paragraphs.

In order to ensure reliable communication, it is critical to obtain an accurate channel model estimation. Wang *et al.* in [126] has described an approach where the channel parameters for the air-to-ground (A2G) links between UAV and GCS is predicted using an unsupervised learning algorithm. Received signal strength (RSS) data from mobile users is collected and used to forecast a temporary 3D channel model for the UAV-GCS link. The channel features are then classified using the k-means clustering technique based on LoS/NLoS parameters. The LoS/NLoS parameters are then classified and used to calculate the path loss between the UAV and the GCS. The proposed model is evaluated by comparing simulation results to a conventional statistical channel model and is found to achieve approximately 91.8% accuracy. However, the proposed approach suffers from shadowing effects which can lead to decrease in system performance and hence should be resolved. Another channel modelling approach by William *et al.* in [127] uses generative neural networks to model the channel at millimeter wave frequencies. Firstly, the proposed model predicts whether each link is LoS or NLoS or in outage. This state information is then fed into a variational autoencoder, which generates the delays, AoAs, and so

on for each propagation path. The methodology is tested for UAVs with 28 GHz A2G channels in an urban environment where the training data sets are produced using Wireless Insite [128] ray tracing software. The proposed model is found to effectively capture the scattering effect from nearby buildings. However, in order to completely validate the accuracy of the model, comparison with real-world channel measurements also needs to be studied and verified for different scenarios. In [129], the authors have used an alternate two-stage method to model the millimeter wave channel model for UAVs. The first stage involves developing an effective channel estimation technique to collect millimeter wave channel information so that each UAV can train a local channel model using generative adversarial networks (GAN). In the second stage, a novel distributed GAN framework is developed, allowing each UAV in the UAV network to share channel information with each other while maintaining privacy. According to simulation results, the proposed approach improves average UAV downlink rate by more than 10% when compared to baseline real-time channel estimation schemes. The results also show that sharing more generated channel samples increases the learning rate, but decreases as the total number of UAVs in the network increases. Additionally, in comparison with a perfect CSI scheme, the proposed method yields lower data rate due to the inevitable training error.

Improved interference mitigation schemes are also required for reliable and low-latency communication between multiple UAVs. In [130], the authors used a deep reinforcement learning (DRL)-based echo state network (ESN) to address interference issues in cellular-connected UAVs. Each UAV is designed to reduce its interference from the ground network as well as its energy consumption and wireless latency. The proposed approach models the problem as a dynamic game in which each UAV uses the ESN architecture to determine the best path, transmission power, and cell associations. Furthermore, the computational complexity of the proposed algorithm is reduced by determining an upper and lower bound for each UAV's altitude. The simulation results show that the proposed scheme reduces interference for ground users while also improving wireless latency per UAV. One of the limitations of the proposed approach is the increase in runtime complexity of the algorithm when the performance of the ground UEs is improved. In addition to interference mitigation, UAVs are prone to eavesdropping attacks. The work by [131] provides a method for detecting eavesdropping attacks in a UAV assisted wireless network using unsupervised learning methods. The eavesdropping attack is assumed to occur during the authentication process in this case. In order to detect eavesdropping, the authors build predictive models that use one-class support vector machines (OC-SVM) and the k-means clustering algorithm. To train and test the above algorithms, the authors have also developed novel approaches for generating the data sets under varying channel conditions. The results show that k-means clustering performs better when the eavesdropper has a high SNR, however, in terms of stability, the OC-SVM outperforms the k-means.

There is growing interest in integrating UAVs to provide assistance to the existing cellular infrastructure. On this front, the work done by Lins *et al.* in [132], demonstrates the use of UAVs in search and rescue missions by utilizing the existing

5G infrastructure. The authors show that for search and rescue missions, the system intelligence (SI) unit should handle and optimize decision making, communication-computation tradeoffs, and connection establishment while the edge intelligence (EI) unit should take care of optimization of artificial intelligence based end-user applications. The authors further presents a virtualized testbed to demonstrate the above concepts. The demonstration utilizes various DNN partitioning strategies to evaluate the effects of CPU and memory usage, transmission bit rate, and object detection accuracy. Future research can change the transport infrastructure from fixed to dynamic and provide dynamic system adaptation to the propagation channels for search and rescue operations. Another work by Galkin *et al.* in [133], focuses on establishing reliable 5G cellular connectivity in a UAV-based communication system in presence of interferers. A supervised approach is proposed, followed by neural network training, in which the UAV selects the GCS based on distances to the GCS, channel conditions, received signal power, and interferer location. The authors consider a UAV that is equipped with two sets of antennas: an RF omni-directional antenna and a directional antenna. Based on the received signal power from omni-directional antenna and other information, the neural network is trained to establish the connection with the GCS from the directional antenna while achieving best channel quality. The proposed scheme outperforms other schemes, such as the strongest-signal and closest-neighbor association techniques. The proposed method can be further extended by introducing UAV mobility which can lead to handovers complexities to the UAV association problem.

In [134], the three-dimensional spectrum sharing between UAVs and device-to-device (D2D) communication is studied. It is assumed that UAVs share spatial spectrum in the same licensed bands as D2D networks. A machine learning-based stochastic geometry approach is proposed to optimize the area spectral efficiency (ASE) of UAVs while maintaining the required ASE for D2D networks. For training, a gaussian kernel non-linear regression is used with various input parameters such as D2D density, UAV flight height, UAV spectrum sensing radius, fading factor, and so on. The resulting output provides statistics for an approximated log-normal distribution, which can then be used to derive various insights such as false alarm probability, spatial missed detection probability, and so on. The reported simulation results show that as the spatial spectrum sensing radius decreases, so does the coverage probability of UAVs, but there is an improvement in the ASE of UAVs. The proposed method also aids in determining the best spatial spectrum sensing radius given certain network parameters.

In [135], the power allocation issue of a UAV-assisted visible light communication system using non-orthogonal multiple access (NOMA) is considered. The problem is formulated such that the sum-rate of all users is maximized subject to the constraints on power allocation, quality of service of users and UAV position. The proposed method employs the harris-hawkins optimization (HHO) algorithm in conjunction with a fully connected artificial neural network. The optimization algorithm aids the artificial neural network to avoid the "local minima" trap and hence makes it suitable for real-time applications. According to the numerical simulation

results, the proposed algorithm outperforms conventional optimization schemes and algorithms. One of the future extensions for this work is to extend the approach to fixed-wing UAVs to achieve joint 3D trajectory optimization and power allocation. Currently, the algorithm suffers from a performance loss of about 10%, which should be investigated and improved. To meet the power requirements of UAV swarms, the authors of [136] proposed a distributed federated learning (FL) approach. According to this approach, the UAV swarm is made up of a leader UAV and its follower UAVs. Each follower UAV runs a local FL model based on the data it collects and sends the trained data to the leader UAV. The leader UAV gathers these trained models and combines them into a global FL model. The trained global FL model is now being used to create a UAV swarm power allocation and scheduling algorithm. The proposed algorithm outperforms traditional algorithms in terms of energy consumption and delay for the UAV swarm, according to simulation results. It is to be noted that the convergence of the FL model largely depends upon the UAV antenna angle deviations, where a larger angle deviation variance requires additional communication round for convergence. Beamforming is another area of UAV-based communication where machine learning is used. The authors of [137] propose a machine learning-based beamforming technique to enable low-latency communication in a multi-UAV network. If the channel or signal-to-interference noise ratio (SINR) information is provided, the proposed method employs Q-learning to predict the beamforming coupling coefficients. According to simulation results, the proposed method outperforms conventional rapid beam tracking methods. Furthermore, the new method computes the best digital weights for SINR maximization and thus achieves better performance compared to traditional gradient based schemes for large angle deviation scenarios. A brief summary of the discussed machine learning algorithms for UAV communication can be found in Table B.6.

A.7 Open Research Areas and Future Directions

As seen in the preceding sections, the wide range of sensors, hardware and software solutions for UAVs raises a number of issues and potential future research directions. To enable a complete visual perception of the environment, we need to use sensor fusion algorithms backed by machine learning. Even though the sensor fusion algorithms listed above have excellent accuracy for object detection and classification, their performance and reliability in extreme weather conditions and lighting needs to be evaluated further. There is also a scarcity of data sets which can aid in analyzing and evaluating the sensing performance of UAVs in such harsh weather conditions. Moreover, the algorithms listed in [120] and [121] perform 3D object detection which can significantly increase the computational overhead and hence have to optimized to port to UAV systems. Further research into dynamic sensor fusion techniques, depending on the requirements of the imaging scene, is required. Due to the strict requirements for inference time for autonomous systems, low-complexity sensor fusion algorithms are also essential. Another potential future extension work, as seen

Table A.7: Machine Learning Techniques for UAV-based Communication

Year	Method	Application	Merits	Demerits	References
2021	Distributed GAN	Channel modelling	10% improved average UAV downlink rate compared to baseline	Lower data rate with respect to perfect CSI	[129]
2021	Deep neural networks (SSD-VGG16)	Cellular assistance	Virtualized testbed to demonstrate DNN partitioning for 5G infrastructure	Dynamic system adaptation for propagation channels is not present	[132]
2020	Variational autoencoder	Channel modelling	Accurate channel parameter estimation and scattering effect	Validation with real-world channel measurements is not provided	[127]
2020	Artificial neural networks	Cellular assistance, Interference management	Improved directional beamforming	UAV mobility can introduce handover issues	[133]
2020	Gaussian kernel non-linear regression	Spectrum sharing	Optimal spatial spectrum radius with maximum ASE for UAV networks	Decreased spatial spectrum radius can lead to increased inter-UAV interference	[134]
2020	Artificial neural networks	Power allocation	Real-time applications	10% performance loss	[135]
2020	Distributed federated learning	Scheduling, Power allocation	Better energy consumption and lower delay for UAV swarms	Large angle deviation requires additional computation time	[136]
2020	OC-SVM, K-means	Physical layer security	OC-SVM is less sensitive to change in signal power whereas k-means is more resistant towards high SNR attacks	Verification with real-time experiments is not provided	[131]
2020	Q-learning	Beamforming, Interference mitigation	Enhanced beamforming due to improved coupling coefficient estimation	Noise and interference can distort the received coupling coefficient power leading to performance loss	[137]
2019	DRL ESN	Interference management	Reduces interference to ground users and improves wireless latency	Increased runtime complexity	[130]
2019	K-means	Channel modelling	91.8% accuracy with respect to statistical channel model	Shadowing effects	[126]

in [121], is to improve red light detection performance in the transformer model under various conditions. In [116], the authors provide a novel method of fixed pillar expansion to reduce the height inaccuracies from radar detections. This can help in reducing computational complexity but it can lead to inaccurate estimate of the center point of the object. Employing adaptive pillar size can be one possible future extension. Performing joint detection and tracking using videos and point cloud streams is also an interesting research direction. As some fusion algorithms rely on active sensors such as radars and LiDARs, the interference effects of these sensors must be investigated. A possible research direction in this regard is to develop novel interference mitigation techniques for mmWave FMCW radars [138]. Even though work has been done to improve the AoA and AoA resolution [54], [53], there is still room for advancement in this area for mmWave FMCW radars such as accurate estimation using low-complexity algorithms with reduced inference time.

In terms of communication, when UAVs are outfitted with various communication modules for different applications, obtaining an optimized communication scheme that takes into account various constraints such as energy, latency, and spectrum availability is a potential research direction. Moreover, devising algo-

rithms which perform equally well during UAV motion also requires further investigation. One of the future extensions for [129] can be to improve the distributed GAN model to incorporate UAV mobility and NLoS aspects to ensure robust and reliable communication. For [132], the optimal utilization of available UAV resources to accomodate the various machine learning enhanced communication algorithms is a possible future extension. Furthermore, there is growing interest in the research community in integrating existing cellular networks with UAV networks in order to enable future wireless technologies. Efficient handovers and association in cellular-connected UAVs where the UAVs are in motion provides interesting directions for further research. Apart from the above, an important area of future research is joint sensing and communication for mmWave and UWB sensors. The authors of [139] has provided proof of concept simulations for joint sensing and communication in mmWave radars, yet this area is still nascent and shows huge potential. Expanding joint sensing and communication with enhanced machine learning models can also be explored provided the UAV resource constraints are adequately met. Other potential research directions include efficient and power-optimized routing protocols for UAV wireless networks to cater to different channel conditions and applications.

A.8 Conclusion

We provided a brief overview of the various hardware and software technologies used for UAVs in this survey article. Various on-board sensors, communication components, and computing platforms were discussed, as well as some practical information about the technologies' key metrics. Among on-board sensors, RGB-D cameras provide better visual information related to the static environment whereas radars and LiDARs provide better dynamics of the different objects present. From the perspective of UAV-based communication, LoRa provides long distance coverage, whereas better reliability can be achieved using Wi-Fi for short distances. Pixhawk 4 provides dedicated computational resources and flexibility to operate UAVs. To cater to deep learning applications, the Nvidia boards offer an excellent choice in terms of performance and form factor. There is also discussion of software solutions that aid in various UAV applications. A brief description of the current state-of-the-art machine learning and deep learning algorithms used in UAVs for sensing and communication is also provided. CNNs offer reliable object detection and flexibility to be used with other sensor modalities. For UAV communication, based on the application, a number of machine learning algorithms ranging from deep neural networks to deep reinforcement learning are in use. The information in this article is intended to provide the reader with the most recent sensor, communication and computing technologies for UAVs, as well as research directions for emerging UAV-based applications.

References

[1] Drone market outlook in 2021. [Online]. Available: https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts.

[2] Panagiotis Katsigiannis et al. An autonomous multi-sensor UAV system for reduced-input precision agriculture applications. In *Proc. MED*, pages 60–64, Athens, Greece, 2016.

[3] Ario Yudo Husodo et al. Intruder drone localization based on 2D image and area expansion principle for supporting military defence system. In *Proc. IEEE ComNetSat*, pages 35–40, Makassar, Indonesia, 2019.

[4] Parikshit Maini and P. B. Sujit. On cooperation between a fuel constrained UAV and a refueling UGV for large scale mapping applications. In *Proc. ICUAS*, pages 1370–1377, Denver, CO, USA, 2015.

[5] Cornelius Holness et al. Remote sensing archeological sites through unmanned aerial vehicle (UAV) imaging. In *Proc. IEEE IGARSS*, pages 6695–6698, Beijing, China, 2016.

[6] Amazon Prime Air. [Online]. Available: https://www.amazon.com/b?node=8037720011.

[7] Linga Reddy Cenkeramaddi et al. A survey on sensors for autonomous systems. In *Proc. IEEE ICIEA*, pages 1182–1187, Kristiansand, Norway, 2020.

[8] Emad Samuel Malki Ebeid et al. A survey of open-source UAV flight controllers and flight simulators. *Microprocessors and Microsystems*, 61:11–20, May 2018.

[9] Abhishek Sharma et al. Communication and networking technologies for UAVs: A survey. *J. Netw. Comput. Appl.*, 168:102739, Jul. 2020.

[10] Mohammad Mozaffari et al. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Commun. Surveys Tuts.*, 21(3):2334–2360, Mar. 2019.

[11] Ursula Challita et al. Machine learning for wireless connectivity and security of cellular-connected UAVs. *IEEE Wireless Commun.*, 26(1):28–35, Feb. 2019.

[12] Srishti Srivastava, Sarthak Narayan, and Sparsh Mittal. A survey of deep learning techniques for vehicle detection from UAV images. *Journal of Systems Architecture*, 117:102152, May 2021.

[13] Praveen Kumar Reddy Maddikunta et al. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sensors J.*, 21(16):17608–17619, Jan. 2021.

[14] Intel RealSense D400 series - Datasheet. [Online]. Available: https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf.

[15] Intel RealSense Depth Camera D435. [Online]. Available: https://www.intelrealsense.com/depth-camera-d435/.

[16] Marcin Odelga, Paolo Stegagno, and Heinrich H. Bülthoff. Obstacle detection, tracking and avoidance for a teleoperated UAV. In *Proc. ICRA*, pages 2984–2990, Atlanta, GA, USA, 2016.

[17] Xi Dai et al. Navigation of simultaneous localization and mapping by fusing RGB-D camera and IMU on uav. In *Proc. SAFEPROCESS*, pages 6–11, Xiamen, China, 2019.

[18] FLIR Boson Thermal Imaging Core - Datasheet. [Online]. Available: https://flir.netx.net/file/asset/15754/original/attachment.

[19] FLIR Boson Compact LWIR - Datasheet. [Online]. Available: https://flir.netx.net/file/asset/12673/original/attachment.

[20] Hakan Kayan et al. Heat leakage detection and surveiallance using aerial thermography drone. In *Proc. SIU*, pages 1–4, Izmir, Turkey, 2018.

[21] Morris William Díaz and Juan José Cáceres. A novel application of drones: thermal diagnosis of electrical and telecommunications infrastructure. In *Proc. IEEE CONCAPAN*, pages 1–6, San Salvador, El Salvador, 2018.

[22] Frederik S. Leira, Tor Arne Johansen, and Thor I. Fossen. Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera. In *Proc. IEEE AeroConf*, pages 1–10, Big Sky, MT, USA, 2015.

[23] D.E. Manolakis. Efficient solution and performance analysis of 3-D position estimation by trilateration. *IEEE Trans. Aerosp. Electron. Syst.*, 32(4):1239–1248, Oct. 1996.

[24] LEA-6 u-box 6 GPS - Datasheet. [Online]. Available: https://www.u-blox.com/sites/default/files/products/documents/LEA-6_DataSheet_%28UBX-14044797%29.pdf.

[25] EMLID RTK GNSS - Datasheet. [Online]. Available: https://emlid.com/reach/.

[26] Yuvaraj Ganesh, Ramya Raju, and Rajeshwari Hegde. Surveillance drone for landmine detection. In *Proc. ADCOM*, pages 33–38, Chennai, India, 2015.

[27] Ramin Heidarian Dehkordi et al. Using UAV collected RGB and multispectral images to evaluate winter wheat performance across a site characterized by century-old biochar patches in Belgium. *Remote Sensing*, 12(15), Aug. 2020.

[28] Jeonghoon Kwak and Yunsick Sung. Autonomous UAV flight control for GPS-based navigation. *IEEE Access*, 6:37947–37955, Jul. 2018.

[29] Future farming. [Online]. Available: http://www.fao.org/e-agriculture/news/exploring-agricultural-drones-future-farming-precision-agriculture-mapping-and-spraying.

[30] P-1750 IMU - Datasheet. [Online]. Available: https://kpp-public.s3.amazonaws.com/DS_P-1750_IMU.

[31] Sparkfun - IMU Selection Guide. [Online]. Available: https://www.sparkfun.com/pages/accel_gyro_guide.

[32] IMU 3000 - Datasheet. [Online]. Available: https://www.sparkfun.com/datasheets/Sensors/IMU/ps-imu-3000a-00-01.1.pdf.

[33] Christian Eling, Lasse Klingbeil, and Heiner Kuhlmann. Real-time single-frequency GPS/MEMS-IMU attitude determination of lightweight UAVs. *Sensors*, 15(10):26212–26235, Oct. 2015.

[34] Marcin Odelga, Nicholas Kochanek, and Heinrich H. Bülthoff. Efficient real-time video stabilization for UAVs using only IMU data. In *Proc. RED-UAS*, pages 210–215, Linköping, Sweden, 2017.

[35] HRXL-MaxSonar - WR Series - Datasheet. [Online]. Available: https://www.maxbotix.com/documents/HRXL-MaxSonar-WR_Datasheet.pdf.

[36] Ultrasonic Ranging Module HC-SR04 - Datasheet. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf.

[37] Meng Guanglei and Pan Haibing. The application of ultrasonic sensor in the obstacle avoidance of quad-rotor UAV. In *Proc. IEEE CGNCC*, pages 976–981, Nanjing, China, 2016.

[38] David Gareth Davies et al. Ultrasonic sensor for UAV flight navigation. In *Proc. IWED*, pages 1–7, Moscow, Russia, 2018.

[39] Dayi Zhang et al. Autonomous ultrasonic inspection using unmanned aerial vehicle. In *Proc. IEEE IUS*, pages 1–4, Kobe, Japan, 2018.

[40] Amin Kianpisheh et al. Smart parking system (sps) architecture using ultrasonic detector. *Int. J. Softw. Eng. its Appl.*, 6(3):55–58, 2012.

[41] Texas Instruments - Ultrasonic Sensing Basics. [Online]. Available: https://www.ti.com/lit/an/slaa907c/slaa907c.pdf.

[42] Umberto Papa and Giuseppe Del Core. Design of sonar sensor model for safe landing of an UAV. In *Proc. IEEE MetroAeroSpace*, pages 346–350, Benevento, Italy, 2015.

[43] Marvelmind Indoor Navigation System - Operating manual. [Online]. Available: https://marvelmind.com/pics/marvelmind_navigation_system_manual.pdf.

[44] Ivan Kalinov et al. High-precision UAV localization system for landing on a mobile collaborative robot based on an IR marker pattern recognition. In *Proc. IEEE VTCSpring*, pages 1–6, Kuala Lumpur, Malaysia, 2019.

[45] Texas Instruments - mmWave IWR1843 - Datasheet. [Online]. Available: https://www.ti.com/product/IWR1843.

[46] Texas Instruments - mmWave IWR1443 - Datasheet. [Online]. Available: https://www.ti.com/product/IWR1443.

[47] Bram van Berlo et al. Millimeter wave sensing: A review of application pipelines and building blocks. *IEEE Sensors J.*, 21(9):10332–10368, Feb. 2021.

[48] Texas Instruments - mmWave radar sensors: What is mmWave. [Online]. Available: https://www.ti.com/sensors/mmwave-radar/what-is-mmwave.html.

[49] Samiur Rahman and Duncan A. Robertson. Radar micro-Doppler signatures of drones and birds at K-band and W-band. *Scientific Reports*, 8(1):17396, November 2018.

[50] Martins Ezuma et al. Micro-UAV detection with a low-grazing angle millimeter wave radar. In *Proc. IEEE RWS*, pages 1–4, Orlando, FL, USA, 2019.

[51] Prabhat Kumar Rai et al. Localization and activity classification of unmanned aerial vehicle using mmWave FMCW radars. *IEEE Sensors J.*, 21(14):16043–16053, Apr. 2021.

[52] Siddharth Gupta et al. Target classification by mmWave FMCW radars using machine learning on range-angle images. *IEEE Sensors J.*, 21(18):19993–20001, Jun. 2021.

[53] Linga Reddy Cenkeramaddi et al. A novel angle estimation for mmWave FMCW radars using machine learning. *IEEE Sensors J.*, 21(8):9833–9843, Feb. 2021.

[54] Prabhat Kumar Rai et al. Angle and height estimation technique for aerial vehicles using mmWave FMCW radar. In *Proc. COMSNETS*, pages 104–108, Bangalore, India, 2021.

[55] Texas Instruments - The fundamentals of millimeter wave radar sensors. [Online]. Available: https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf.

[56] Velodyne Alpha Prime - Datasheet. [Online]. Available: https://velodynelidar.com/products/alpha-prime/.

[57] Velodyne Puck LITE - Datasheet. [Online]. Available: https://velodynelidar.com/products/puck-lite/.

[58] Temuulen Sankey et al. UAV lidar and hyperspectral fusion for forest monitoring in the southwestern USA. *Remote Sensing of Environment*, 195:30–43, April, 2017.

[59] Yi-Chun Lin et al. Evaluation of UAV LiDAR for mapping coastal environments. *Remote Sensing*, 11(24), Dec. 2019.

[60] Yazhe Tang et al. Vision-aided multi-UAV autonomous flocking in GPS-denied environment. *IEEE Trans. Ind. Electron.*, 66(1):616–626, Apr. 2019.

[61] Yunfei Ma, Nicholas Selby, and Fadel Adib. Drone relays for battery-free networks. In *Proc. SIGCOMM*, pages 335–347, New York, NY, USA, 2017.

[62] Gao RFID - How to choose an RFID tag. [Online]. Available: https://gaorfid.com/how-to-choose-an-rfid-tag/.

[63] Atlas RFID store - How to select UHF RFID reader. [Online]. Available: https://www.atlasrfidstore.com/rfid-insider/how-to-select-uhf-rfid-reader.

[64] Smartrac DogBone - Datasheet. [Online]. Available: https://rfid.atlasrfidstore.com/hubfs/Tech_Spec_Sheets/Smartrac/ATLAS_Dogbone_R6.pdf.

[65] FX7500 Fixed RFID reader - Datasheet. [Online]. Available: https://www.zebra.com/content/dam/zebra_new_ia/en-us/solutions-verticals/product/RFID/Reader/FX7500%20Fixed%20RFID%20Reader/specsheet/fx7500-spec-sheet-en-us.pdf.

[66] Jian Zhang et al. RFHUI: an RFID based human-unmanned aerial vehicle interaction system in an indoor environment. *Digital Communications and Networks*, 6(1):14–22, May 2020.

[67] J. Zhang et al. Robust rfid based 6-dof localization for unmanned aerial vehicles. *IEEE Access*, 7:77348–77361, Jun. 2019.

[68] Inpixon - Ultra-Wideband Technology. [Online]. Available: https://www.inpixon.com/technology/standards/ultra-wideband.

[69] TDSR - P440 UWB Module. [Online]. Available: https://tdsr-uwb.com/products-p440-uwb-module/.

[70] Decawave - DWM1000 UWB Module. [Online]. Available: https://www.decawave.com/product/dwm1000-module/.

[71] Thien-Minh Nguyen et al. Robust target-relative localization with ultra-wideband ranging and communication. In *Proc. ICRA*, pages 2312–2319, Brisbane, QLD, Australia, 2018.

[72] Nguyen, Thien-Minh et al. Integrated UWB-vision approach for autonomous docking of UAVs in GPS-denied environments. In *Proc. ICRA*, pages 9603–9609, Montreal, QC, Canada, 2019.

[73] What is LoRa? - Semtech. [Online]. Available: https://www.semtech.com/lora/what-is-lora.

[74] LoRa Modulation Basics - Semtech. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001OJk/yDEcfAkD9qEz6oG3PJryoHKas3UMsMDa3TFqz1UQOkM.

[75] LoRa - Homepage. [Online]. Available: https://lora-alliance.org/.

[76] Semtech SX1276 - Datasheet. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rbr/6EfVZUorrpoKFfvaF_Fkpgp5kzjiNyiAbqcpqh9qSjE.

[77] Bluetooth - What is Bluetooth LE? [Online]. Available: https://web.archive.org/web/20170310111443/https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works/low-energy.

[78] Pablo Ramon Soria et al. Bluetooth network for micro-uavs for communication network and embedded range only localization. In *Proc. ICUAS*, pages 747–752, Miami, FL, USA, 2017.

[79] Evgeny Khorov et al. A Tutorial on IEEE 802.11ax high efficiency WLANs. *IEEE Commun. Surveys Tuts.*, 21(1):197–216, Sep. 2019.

[80] Antonio Guillen-Perez et al. WiFi networks on drones. In *Proc. ITU WT)*, pages 1–8, Bangkok, Thailand, 2016.

[81] 3GPP - Standards for the IoT. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1805-iot_r14.

[82] OcPoC-Zynq Mini - Datasheet. [Online]. Available: https://www.sundance.com/wp-content/uploads/docs/AerotennaOcPoCminiDatasheet.pdf.

[83] OcPoC-Zynq Mini Specifications. [Online]. Available: https://aerotenna.readme.io/docs/ocpoc-mini-zynq-specifications.

[84] PixHawk4 - Datasheet. [Online]. Available: http://www.holybro.com/manual/Pixhawk4-DataSheet.pdf.

[85] Holybro Products - PixHawk4. [Online]. Available: http://www.holybro.com/product/pixhawk-4/.

[86] PixHawk4 Documentation. [Online]. Available: https://docs.px4.io/master/en/flight_controller/pixhawk4.html.

[87] NuttX Documentation. [Online]. Available: http://nuttx.apache.org/docs/latest/introduction/about.html.

[88] PixHawk4 Mini - Datasheet. [Online]. Available: http://www.holybro.com/manual/Pixhawk4mini_technical_data_sheet.pdf.

[89] PixHawk4 Mini Documentation. [Online]. Available: https://docs.px4.io/v1.12/en/flight_controller/pixhawk4_mini.html.

[90] BeagleBone Blue - Datasheet. [Online]. Available: https://github.com/beagleboard/beaglebone-blue/blob/master/docs/BeagleBone_Blue_ShortSpec.pdf.

[91] Seeed - BeagleBone Blue. [Online]. Available: https://www.seeedstudio.com/BeagleBone-Blue-p-2809.html.

[92] BeagleBone Blue - Documentation. [Online]. Available: https://beagleboard.org/blue.

[93] Raspberry Pi 4 Model B - Datasheet. [Online]. Available: https://datasheets.raspberrypi.org/rpi4/raspberry-pi-4-product-brief.pdf.

[94] Nvidia Jetson Nano - Technical Specifications. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[95] Nvidia Jetson Module - Comparison Table. [Online]. Available: https://developer.nvidia.com/embedded/jetson-modules.

[96] Nvidia Jetson TX2 Module - Datasheet. [Online]. Available: http://developer.nvidia.com/embedded/dlc/jetson-tx2-series-modules-data-sheet.

[97] Nvidia Jetson TX2 Module. [Online]. Available: https://developer.nvidia.com/embedded/jetson-tx2.

[98] Daniel Tan Wei Xun, Yoke Lin Lim, and Sutthiphong Srigrarom. Drone detection using YOLOv3 with transfer learning on NVIDIA Jetson TX2. In *Proc. ICA-SYMP*, pages 1–6, Bangkok, Thailand, 2021.

[99] Nvidia Jetson AGX Xavier - Documentation. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit.

[100] Nvidia Jetson AGX Xavier - Technical Specifications. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-xavier.

[101] Arduino Mega 2560 Documentation. [Online]. Available: https://docs.arduino.cc/hardware/mega-2560.

[102] Arduino Mega 2560 Wiki. [Online]. Available: https://www.arduino.cc/en/pmwiki.php?n=Main/arduinoBoardMega2560.

[103] Xiaodi Zhang. Adaptive control of quadrotor uav based on arduino. In *Proc. PESA*, pages 1–4, Hong Kong, China, 2020.

[104] Intel UP board - Datasheet. [Online]. Available: https://up-board.org/wp-content/uploads/datasheets/UPDatasheetV8.5.pdf.

[105] Intel UP board series. [Online]. Available: https://up-shop.org/up-board-series.html.

[106] Jorge Peña Queralta et al. UWB-based system for UAV localization in GNSS-denied environments: Characterization and dataset. In *Proc. IEEE IROS*, pages 4521–4528, Las Vegas, NV, USA, 2020.

[107] ArduPilot Homepage. [Online]. Available: https://ardupilot.org/ardupilot/.

[108] ArduPilot - Supported Computers. [Online]. Available: https://ardupilot.org/dev/docs/companion-computers.html.

[109] ArduPilot Source Code - GitHub. [Online]. Available: https://github.com/ArduPilot/.

[110] PX4 Homepage. [Online]. Available: https://px4.io/.

[111] PX4 Source Code - GitHub. [Online]. Available: https://github.com/PX4/PX4-Autopilot.

[112] Ratheesh Ravindran, Michael J. Santora, and Mohsin M. Jamali. Multi-object detection and tracking, based on DNN, for autonomous vehicles: A review. *IEEE Sensors J.*, 21(5):5668–5677, Dec. 2021.

[113] Abdullah S. Alharthi, Syed U. Yunas, and Krikor B. Ozanyan. Deep learning for monitoring of human gait: A review. *IEEE Sensors J.*, 19(21):9575–9591, Jul. 2019.

[114] Su Pang, Daniel Morris, and Hayder Radha. CLOCs: Camera-LiDAR object candidates fusion for 3D object detection. In *Proc. IEEE IROS*, pages 10386–10393, Las Vegas, NV, USA, 2020.

[115] Felix Nobis et al. A deep learning-based radar and camera sensor fusion architecture for object detection. In *Proc. SDF*, 2019.

[116] Ramin Nabati and Hairong Qi. CenterFusion: Center-based radar and camera fusion for 3D object detection. In *Proc. IEEE WACV*, pages 1526–1535, Waikoloa, HI, USA, 2021.

[117] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. PointFusion: Deep sensor fusion for 3D bounding box estimation. In *Proc. CVPR*, pages 244–253, Salt Lake City, UT, USA, 2018.

[118] Xiangmo Zhao et al. Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications. *IEEE Sensors J.*, 20(9):4901–4913, Jan. 2020.

[119] Akshay Rangesh and Mohan Manubhai Trivedi. No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras and lidars. *IEEE Trans. Intell. Veh.*, 4(4):588–599, Aug. 2019.

[120] Vishwanath A. Sindagi, Yin Zhou, and Oncel Tuzel. MVX-Net: Multimodal voxelnet for 3D object detection. In *Proc. ICRA*, pages 7276–7282, Montreal, QC, Canada, 2019.

[121] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proc. CVPR*, Apr. 2021.

[122] Jin Hyeok Yoo et al. 3D-CVF: Generating joint camera and LiDAR features using cross-view spatial feature fusion for 3D object detection. In *Proc. Computer Vision – ECCV 2020*, pages 720–736. Springer International Publishing, 2020.

[123] Hongbo Gao et al. Object classification using CNN-based fusion of vision and LiDAR in autonomous vehicle environment. *IEEE Trans. Ind. Informat.*, 14(9):4224–4231, Apr. 2018.

[124] Chen Wang et al. DenseFusion: 6D object pose estimation by iterative dense fusion. In *Proc. CVPR*, pages 3338–3347, Long Beach, CA, USA, 2019.

[125] Zhiyu Huang et al. Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding. *IEEE Sensors J.*, 21(10):11781–11790, Jun. 2021.

[126] Jing-Ling Wang et al. Machine learning based rapid 3D channel modeling for UAV communication networks. In *Proc. IEEE CCNC*, pages 1–5, Las Vegas, NV, USA, 2019.

[127] William Xia et al. Millimeter wave channel modeling via generative neural networks. In *Proc. IEEE GC Wkshps*, pages 1–6, Taipei, Taiwan, 2020.

[128] Wireless InSite - Remcom. [Online]. Available: https://www.remcom.com.

[129] Qianqian Zhang, Aidin Ferdowsi, and Walid Saad. Distributed generative adversarial networks for mmwave channel modeling in wireless UAV networks. In *Proc. IEEE ICC*, pages 1–6, Montreal, QC, Canada, 2021.

[130] Ursula Challita, Walid Saad, and Christian Bettstetter. Interference management for cellular-connected UAVs: A deep reinforcement learning approach. *IEEE Trans. Wireless Commun.*, 18(4):2125–2140, Mar. 2019.

[131] Tiep M. Hoang, Nghia M. Nguyen, and Trung Q. Duong. Detection of eavesdropping attack in UAV-aided wireless systems: Unsupervised learning with one-class SVM and K-means clustering. *IEEE Wireless Commun. Lett.*, 9(2):139–142, Oct. 2020.

[132] Silvia Lins et al. Artificial intelligence for enhanced mobility and 5G connectivity in UAV-based critical missions. *IEEE Access*, 9:111792–111801, Aug. 2021.

[133] Boris Galkin et al. Intelligent base station association for UAV cellular users: A supervised learning approach. In *Proc. IEEE 5GWF*, pages 383–388, Bangalore, India, 2020.

[134] Bodong Shang et al. 3D spectrum sharing for hybrid D2D and UAV networks. *IEEE Trans. Commun*, 68(9):5375–5389, May 2020.

[135] Quoc-Viet Pham et al. Sum-rate maximization for UAV-assisted visible light communications using NOMA: Swarm intelligence meets machine learning. *IEEE Internet Things J.*, 7(10):10375–10387, Apr. 2020.

[136] Tengchan Zeng et al. Federated learning in the sky: Joint power allocation and scheduling with UAV swarms. In *Proc. IEEE ICC*, pages 1–6, Dublin, Ireland, 2020.

[137] Hsiao-Lan Chiang et al. Multi-UAV mmWave beam tracking using Q-learning and interference mitigation. In *Proc. IEEE ICC Workshops*, pages 1–7, Dublin, Ireland, 2020.

[138] Ashish Basireddy and Hassan Moradi. OFDM waveform design for interference resistant automotive radars. *IEEE Sensors J.*, 21(14):15670–15678, 2021.

[139] Thuy M. Pham et al. Joint communications and sensing experiments using mmWave platforms. In *Proc. IEEE SPAWC*, pages 501–505, Lucca, Italy, 2021.

Appendix B

PAPER B

Title: Recent Advances in Thermal Imaging and its Applications Using Machine Learning: A Review

Authors: **A. N. Wilson**, K. A. Gupta, B. H. Koduru, A. Kumar, A. Jha, L. R. Cenkeramaddi

Journal: IEEE Sensors Journal

Recent Advances in Thermal Imaging and its Applications Using Machine Learning: A Review

A. N. Wilson, K. A. Gupta, B. H. Koduru, A. Kumar, A. Jha, L. R. Cenkeramaddi

Abstract: **Recent advancements in thermal imaging sensor technology have resulted in the use of thermal cameras in a variety of applications, including automotive, industrial, medical, defense and space, agriculture, and other related fields. Thermal imaging, unlike RGB imaging, does not rely on background light, and the technique is non-intrusive while also protecting privacy. This review article focuses on the most recent advancements in thermal imaging technology, key performance parameters, an overview of its applications, and machine learning techniques applied to thermal images for various tasks. The article begins with the most recent advancements in thermal imaging, followed by a classification of thermal cameras and their key specifications, and finally a review of machine learning techniques used on thermal images for various applications. This detailed review article is highly useful for designing thermal imaging-based applications using various machine learning techniques.**

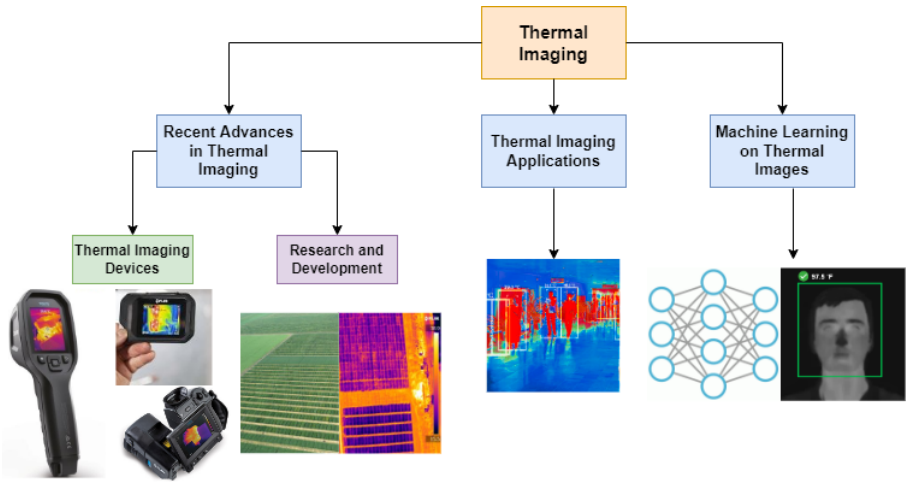


Figure B.1: Diagramatic overview about the recent advances in thermal imaging

Table B.1: Subbands in the infrared spectrum

S No.	IR Band	Wavelength (in nm)
1	Near infrared	700 – 1400
2	Short range wavelength infrared	1400 – 3000
3	Mid range wavelength infrared	3000 – 8000
4	Long range wavelength infrared	8000 – 15000
5	Far infrared	15000 – 1000000

B.1 Introduction

Since 1960, thermal imaging was confined only to military [1] and medical [2] applications, however, with the recent advancements in chip technology and lower cost, thermal imaging has gained widespread popularity. Thermal imaging works by utilizing the radiation in the infrared region of the spectrum, specifically the wavelengths from 3 to 14μm. These wavelengths are subdivided into different subbands as shown in Table B.1. Special devices called thermal imagers utilizes the infrared part of the spectrum to obtain a spatial temperature distribution map of the captured scene [3]. Each pixel in the temperature map depicts the relative temperature of that point in the environment. These temperature maps can be easily used for real-time applications with proper calibration, bias removal, and further processing. [4].

Thermal imaging technology is independent of any external light source because it is based solely on the detection of infrared radiations emitted by objects. As a result, the technology is found to have a faster processing speed than its RGB counterparts [5]. Thermal imaging devices are now being widely used in civilian applications such as fever scanners, insulation detectors, and electrical hotspot detectors due to lower chip costs, improved portability, and flexible designs. Combining thermal cameras with RGB cameras has also gained popularity due to their ability to complement each other's features [4]. In addition to the benefits listed above, thermal imaging provides a non-contact, non-invasive method known as infrared thermography for obtaining useful information about a patient's health and diagnosis, which has widespread medical applications [6].

The high sensitivity of thermal cameras have enabled them to be used in optical applications as well [7]. Other applications enabled by thermal imaging include fire prediction, weather forecasting, and animal monitoring [7]. RGB cameras depend on illumination and reflection from the objects whereas thermal cameras are sensitive to the emitted infrared radiation, even if the object is cold[7]. Because each object's heat signature is unique, thermal cameras have an advantage over standard RGB cameras in distinguishing between similar objects.

Thermal cameras have grown in popularity and use as a result of the benefits listed above. This article focuses on various key technological advancements to provide a glimpse into the most recent developments in this technology. The article primarily highlights the various types of thermal imaging devices/cameras on the

Table B.2: Comparison of previous review articles with our work

Year	Reference	Focus areas	RA	Appl.	CM	Image Pro.	ML Tech.	Comments
2005	[8]	Image processing techniques for active and passive thermography	✗	✗	✗	✓	✗	Restricted to processing techniques
2009	[9]	Status of intra-operative thermal imaging and case report on it's advantages and applications	✓	✓	✗	✗	✗	Restricted to intra-operative thermal imaging
2014	[10]	Uses and applications of thermal Imaging in Agriculture	✓	✓	✗	✗	✗	Restricted to the field of agriculture
2017	[11]	Theory behind thermal imaging and its applications in different fields	✓	✓	✗	✗	✗	Restricted to working and applications of thermal imaging
2020	[12]	Techniques for face Emotion detection using thermal imaging	✓	✓	✗	✗	✓	Restricted to facial emotion detection
2020	[13]	Review of techniques and methodologies on diagnosing breast cancer using thermal imaging	✓	✓	✓	✗	✗	Restricted to applications in breast cancer diagnosis
2021	[14]	Role of thermal sensors and imaging in aerial navigation systems	✓	✓	✓	✗	✓	Restricted to aerial navigation systems
2022	**Our Work**	**Recent advancements in thermal imaging, latest models of thermal cameras available in the market, image processing and machine learning techniques related to thermal imaging**	✓	✓	✓	✓	✓	**Not restricted to particular filed of study or application**

RA - Recent Advancements; **Appl.** - Applications; **CM** - Current Models in market; **Image Pro.** - Image Processing; **ML Tech.** - Machine Learning Techniques

market, thermal camera selection criteria based on application and specifications, recent machine learning techniques for thermal image processing, and potential future research directions. Our survey article differs from others in terms of application focus, recent advancements, a brief overview of camera models, and machine learning techniques. Table B.2 highlights the most important aspects of this article, demonstrating how it differs from previous surveys.

The remainder of this paper is structured as follows: Section B.3 focuses on recent advancements in thermal imaging that includes different thermal camera models and latest research and development in the area. Section B.4 presents different thermal imaging based applications. Section B.5 goes over the recent machine learning techniques used along with thermal imaging. Finally, section B.6 concludes the article.

B.2 Principle of thermal sensing

Thermal imaging is a non-contact and non-destructive method to measure the temperature of an object [15]. Thermal imaging utilizes the infrared radiation (IR) emitted from an object to create a visual temperature profile of the captured scene.

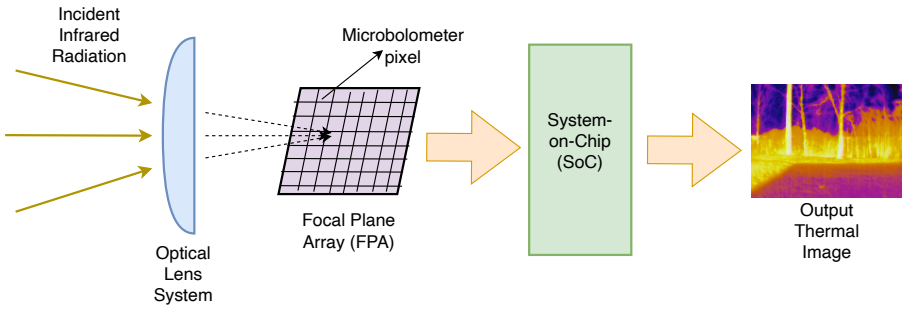


Figure B.2: Working of bolometer-based thermal sensor

As shown in Table B.1, the infrared spectrum is divided into different subbands based on their wavelength. The wavelength determines the intensity of infrared radiation that is emitted. Thermal imaging technology utilizes this energy intensity to generate the temperature map of the captured scene. The amount of thermal radiation emitted by a body primarily depends on the temperature (T) of the body and its emissivity factor (ε). The emissivity factor represents the ratio of energy emitted from a body to that of a perfect black body at the same temperature. The emissivity factor is 1 for a perfect black body and 0 for a perfect white body. Based on the IR energy radiated from a body, the surface temperature T_s of the body can be calculated as follows:

$$W = \left[\frac{2\pi^5 k^4}{15c^2h^3}\right] T^4 = \sigma T_s^4 \tag{B.1}$$

where W represents the energy flux emitted per unit area (Wm^{-2}) of the body, c is the speed of light in vacuum ($3\mathrm{x}10^8\mathrm{ms}^{-1}$), k is the Boltzmann's constant ($1.38\mathrm{x}10^{-23}\mathrm{JK}^{-1}$), σ is Stefan-Boltzmann constant ($5.67\mathrm{x}10^{-8}$ $\mathrm{Wm}^{-2}\mathrm{K}^{-4}$), h is the Planck's constant ($6.63\mathrm{x}10^{-34}$ Js), and T is the temperature of the body in Kelvin.

When (B.1) is applied to real objects, then the surface temperature is computed as,

$$W = \varepsilon\sigma T^4 \tag{B.2}$$

where ε is the object's emissivity. By utilizing W, we can obtain a thermal visualization of the captured scene which is the basis of thermal imaging [15].

The primary component of a thermal imaging system is the thermal detector/sensor. The thermal detector is responsible for mapping the incident infrared radiation to an appropriate temperature value. Based on the operating principle, thermal detectors are classified into three types, pyroelectric, thermoelectric, and bolometer sensors as shown in Table B.3. Pyroelectric sensors are made up of special materials that accumulate the charge on the basis of incident infrared radiation. A temperature change in the captured scene induces a proportional change in the accumulated charge. This change in the accumulated charge is used to obtain the thermal profile of the scene [16]. On the other hand, thermoelectric sensors operate according to the Seebeck effect [17]. Seebeck effect is the phenomenon by which a voltage difference is produced based on the temperature difference between two dissimilar electrical

Table B.3: Popular thermal sensors and key sensing technology

Sl. No.	Model	Type	Material	Manufacturer
1	Lepton 3.5 [19]	Bolometer	Vanadium-oxide (VOx)	FLIR
2	Pyrosens [20]	Pyroelectric	Lithium tantalate	DIAS Infrared
3	InspectionCAM IQ-AAA [21]	Bolometer	Vanadium-oxide (VOx)	Seek Thermal
4	D6T [18]	MEMS-based thermoelectric	-	Omron
5	Evo Thermal 90 [22]	Thermoelectric	-	Terabee

conductors. Microelectromechanical systems (MEMS) based thermoelectric sensors focus the incident infrared radiation onto a thermoelectric sensor. The amount of incident infrared radiation generates an equivalent voltage. The induced voltage is then used to compute the object's temperature using interpolation and look-up table approximations. [18]. Compared to pyroelectric sensors, thermoelectric sensors are reliable and cheap. However, thermoelectric sensors suffer from non-linearity issues due to the non-linear dependence between the output voltage and measured temperature.

Recently, bolometer-based thermal detectors have gained popularity due to their high thermal sensitivity, small size, and high accuracy. A bolometer is a special material whose electrical resistance responds to the amount of infrared radiation incident on it. Commonly used materials for bolometers include vanadium oxide (VOx) and amorphous silicon (a-Si). An example of a bolometer-based thermal sensor is the FLIR Lepton 3.5 [19]. The FLIR Lepton 3.5 uses a VOx-based microbolometer array for thermal imaging. Fig. B.2 shows a simplified block diagram of the operation of a microbolometer-based thermal sensor. Fig. B.2 shows that the optical lens system focuses the incident infrared radiation onto the focal plane array (FPA). Each element on the FPA represents a pixel and each pixel is in turn a VOx microbolometer that responds to the incident flux by producing a temperature change. The temperature change is proportional to the resistance of the microbolometer. The change in resistance is captured by the voltage fluctuations which is fed into a system-on-chip (SoC). The SoC performs the necessary signal processing and outputs the thermal profile of the scene [19].

B.3 Recent Advances in Thermal Imaging

The early thermal camera sensors were designed with a lens filled with gas. They also required refrigeration to function properly. However, due to advancements in semiconductor technology, thermal cameras are now comparable to standard charge-coupled device (CCD) cameras. Furthermore, their improved portability and low cost have made them suitable for use in several applications [23].

Advancements in thermal imaging have paved way for thermal camera sensors that can help in enhancing user interaction with the environment. Thermal imaging-based sensors are used in games to identify the effect of moral decisions based on

the user's facial heat map [24]. Thermal cameras have become portable and easy to integrate such that they are now being used in pocket devices like FLIR C2, FLIR One, Cat S60, and Landguide M4 [4]. Recently dual camera systems with thermal camera integrated along with visual cameras have been developed to provide application-based usage i.e, when surveillance is required, the thermal camera mode is enabled in the dual camera setup. The dual camera setup is used in parking lots to determine car parking history or recently occupied parking spots based on the heat emitted from the engines or surrounding surfaces [4]. However, thermal camera integration with visual camera increases the bandwidth of applications. As thermal cameras cannot detect visual information such as numbers, signs, and words, integration of an optical image provides an additional advantage to the thermal image and thereby enhances it [25]. Through this overlay of optical and thermal images, highly informative and contrast images are obtained, making the detection of hotspots and sources of fire and heat easier. Currently, thermal cameras have become ubiquitous with a wide variety of selection to choose from. Fig. B.3 displays the various types of thermal camera classes. These are categorized with respect to factors such as usage, application, temperature, and range as explained in Table B.4.

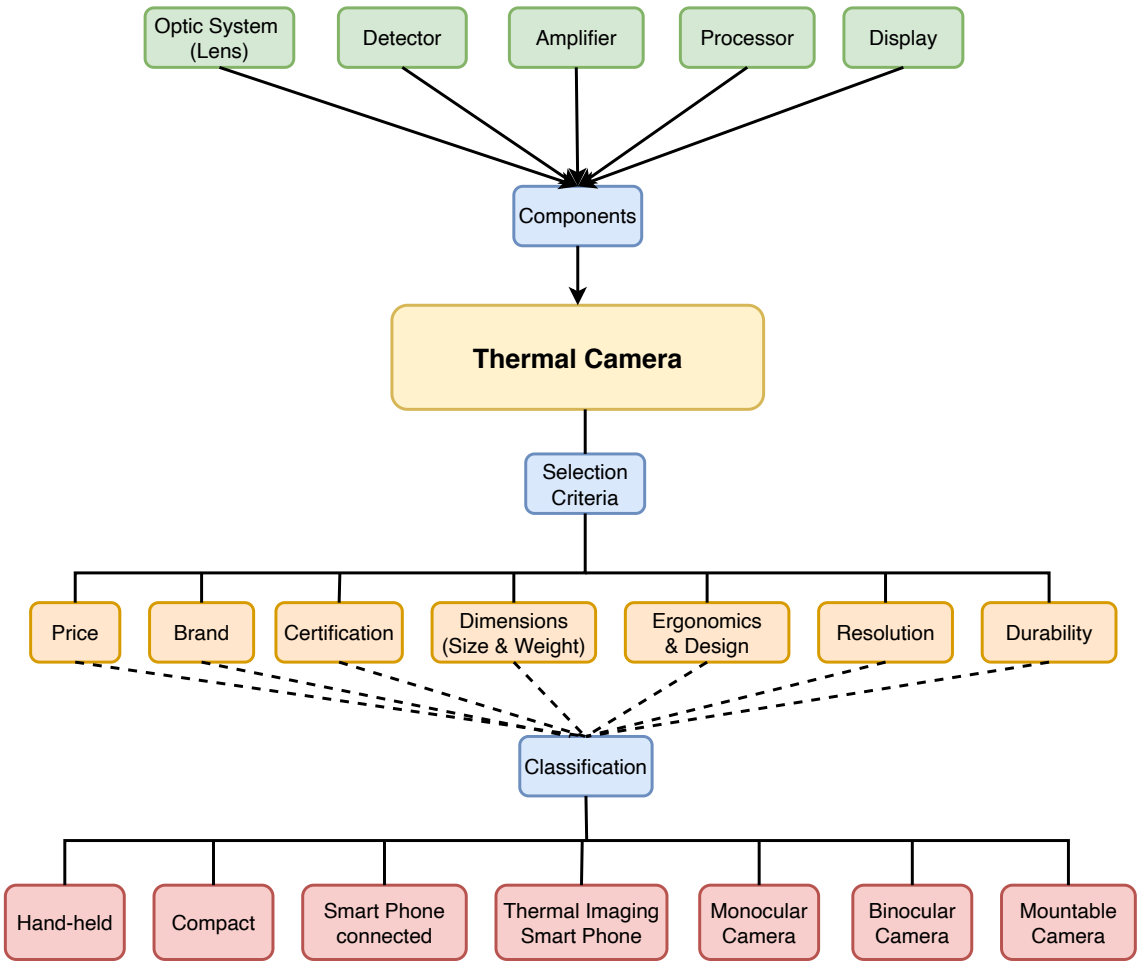


Figure B.3: Components, selection criteria, and classification of thermal cameras

Table B.4: Suitable camera model with respect to the applications

Sl. No.	Temperature	Range	Inspection	Application	Model
1.	Low	Short	Quick and small	Facility maintenance, HVACs pros	E4 through E8
2.	High	Mid and short	Small	Electricians and plant site maintenance	E40 through E60
3.	Low and high	Mid, short, and long	Intensive	Substation surveys and solar farm surveys	T420 to T640

B.3.1 Latest Developments in Thermal Cameras

The current subsection discusses the latest thermal camera models available in the market. Table B.5 shows the different thermal camera models along with some key metrics which can help in deciding the right thermal camera for different applications. Thermal camera models from companies FLIR and MOBOTIX have been discussed here along with their suitability in accordance with various applications.

FLIR thermal imaging cameras are used for predictive maintenance. They are also equally used by electricians and technicians to detect and resolve electrical issues, isolation issues, etc. These cameras are well suited for making long-distance inspections with accurate temperature profiles [35]. Furthermore, the multi-spectral dynamic imaging (MSX) feature in these cameras enables multi spectral dynamic imaging to make the thermal images more refined. The interfaces are also well developed to ensure easy transfer of output data. This feature can be found in Ex, Exx, and the T series versions of FLIR thermal cameras [36].

The E series thermal camera range includes E4, E5, E6, and E8 all of which are highly portable and can be used to detect hidden defects. This allows technicians to take instant action in response to a situation before it becomes too serious [36]. These cameras have thermal, visible and MSX imaging in it. Based on the type of E series model, the IR imaging resolution can be adjusted; E4 (upto 4800 pixels), E5 (up to $10,800$ pixels), E6 (upto 19200 pixels), and E8 (upto $76,800$ pixels). The E40, E50, and E60 models are for frequent and wide angled inspection for onsite technicians and electricians. These cameras also have high wireless connectivity and touchscreen control to do instant analysis of the captured thermal images [36]. FLIR T series is suited for measurements in extreme conditions such as long range or high temperature. It has a rotating optical block and auto rotation feature to correctly aim the target for exact measurement and better view for analysis and capture. T620 and T640 has built-in GPS to add location to the thermal image for better labelling [36]. FLIR A655sc can be used in applications where the thermal camera mount needs to be fixed. For InGaAs detection, FLIR A6200sc thermal camera is suitable. For high-speed mid-wave infrared (MWIR), FLIR X8400sc series shows promise [36].

MOBOTIX thermal cameras are widely used for surveillance applications. M16 Thermal [37] has two adjacent lenses which does thermal overlay with the visual image to pinpoint the location of hotspots like fire-affected regions in an image.

Table B.5: Popular thermal cameras and their specifications

Sl. No.	Model Name (Brand)	Camera Type	Size (mm)	Sensor Resolution (pixels)	Price (USD)	Detecting Temperature Range (°C)	Thermal Sensitivity (°C)	User Interface and Connectivity	References
1	FLIR C5 (Teledyne FLIR)	Compact Pocket Thermal Camera	138 × 84 × 24	160 × 120	855	−20 – 400	0 – 100 : ±3 ; 100 – 400 : ±3%	Touchscreen; FLIR Ignite cloud connectivity (using Wi-Fi)	[26]
2	Ti480 PRO (Fluke)	Hand-held Camera	277 × 122 × 167	640 × 480	–	−20 – 1000	±2 or 2%	Touch screen; Wireless connectivity (Smart Phone, PC); Fluke Connect® app compatible	[27]
3	Compact-PRO XR (Seek Thermal)	Smart Phone Connected Thermal Camera	25.4 × 44.45 × 25.4	320 × 240	599	−40 – 330	< 0.070	Seek Thermal app	[28]
4	Helion 2 XP50 Pro (Pulsar)	Monocular Camera	242 × 75 × 60	640 × 480	4376	–	< 0.025	Built-in WiFi module - connects to Smart Phones using Stream Vision 2 app	[29]
5	CAT S62 PRO (CAT)	Thermal Imaging Smart Phone	158.5 × 76.7 × 11.9	1440×1080	530	−20 – 400	–	5.7" FHD + Display	[30]
6	Merger LRF XP50 (Pulsar)	Binocular Camera	196 × 143 × 76	640 × 480	6486	–	< 0.025	Built-in WiFi module - connects to Smart Phones using Stream Vision 2 app	[31]
7	RSE600 (Fluke)	Mountable Camera	83 x 83 × 165	640 × 480	–	−10 – 1200	±2 or ±2%	SmartView® desktop software	[32]
8	M16B Thermal TR (MOBOTIX)	Mountable Camera	210 × 158 × 207	336 × 252	–	−40 – 170	±0.05	HD wideband audio, Ethernet, RS232 support	[33]
9	S16B DualFlex (MOBOTIX)	Mountable Camera	130 × 115 × 33	336 × 252	–	−40 – 160	±0.05	HD wideband audio, Ethernet, RS232 support	[34]

M16 TR thermal camera [38] is a low-power camera that has an additional thermal radiometry feature that enables the measurement of thermal radiation in the image. S16 DualFlex is a flexible dual thermal camera with one or two weatherproof sensors which can withstand any conditions due to the robust casing around the dual camera sensor setup [25], [39]. S16 TR [40] enables the radiation values to trigger an alarm or activation to alert the user if the temperature values exceed or are lesser than the threshold values calibrated in the sensor. Choosing a thermal camera for a particular application requires careful consideration of a variety of factors as it is a long-term investment. One needs to keep in mind the right supplier to suit the needs, as the functioning of the thermal camera depends largely on its hardware. The different thermal camera selection criteria is as shown in Fig.B.3.

To choose the right camera model based on the application, the following char-

acteristics should be kept in mind [41]:-

- *Camera resolution* - Based on the application it can be decided if a basic resolution model is required or an advanced one. Basic resolution is around 60 x 60 pixels. 320 x 240 pixels offer superior definition and for even more advanced resolution 640 x 480 is suitable.

- *Thermal sensitivity* - Thermal sensitivity provides an indication of the thermal cameras ability to sense minute variations in temperature. Higher the value of sensitivity, more accurately the camera can measure lower temperature differences. Hence, in industrial applications where such conditions of lower temperature differences prevail, a thermal camera with high temperature sensitivity should be selected.

- *Accuracy* - Depending upon the desired accuracy of the temperature readings, a suitable thermal camera model should be selected. Currently, the standard accuracy values are $\pm 2\%$ or $\pm 2°C$. However, in more advanced thermal cameras, the accuracy ranges as $\pm 1\%$ or $\pm 1°C$.

- *Camera features* - Based on the application, having the right set of features for the thermal camera is necessary to ensure smooth operations. In certain applications a dual camera setup of visual and thermal camera is required. In others, thermal fusion must be a necessity. In-built GPS helps to determine the location which can be useful in unmanned aerial vehicle (UAV) applications whereas, in others portability is the prime feature. Thus it can seen that according to preference of various camera features, a suitable thermal camera model should be selected.

- *Software*- Software compatibility with the corresponding hardware is essential to maintain operations. Hence, based on the intense level of inspection, the corresponding software should be selected.

The following are the most important considerations to make when selecting a particular thermal camera [41]:

- *Hardware*- It is advantageous to have a wide range of hardware to meet the needs of any custom application at all stages of development, from basic inspection equipment to advanced high resolution defined thermal image producing cameras.

- *Software*- The software should be compatible with the application and hardware platform, as the software defines how the image will be produced and displayed. Hence, based on the information that should be retrieved from the image, an appropriate software should be chosen.

- *Hardware Interfaces*- It is ideal to have multiple hardware interfaces such as I2C, SPI, and USB so that it can be used with a variety of hard platforms and embedded/edge devices.

B.3.2 Future work in Thermal Imaging

Due to the vast scope in thermal imaging and its utilization in different applications, most of the research in thermal imaging is in the production of sophisticated thermal cameras. These cameras are more application specific and has greater range, sensitivity, and tolerance. Interestingly, thermal imaging is being used in a variety of new applications [42]. Researchers use thermal imaging to detect anxiety and classify it based on the heat map of the face. As stated in medical research, based on the type of anxiety, the bloodflow in the face can get altered. Certain types of anxiety can trigger more bloodflow in the cheeks whereas others can incite low bloodflow in the forehead [23]. Current research is also focussed on Airborne thermography in which high resolution thermal imaging is used to measure crop fields on the basis of temperature, drought tolerance of crops, and efficient water delivery[42].

B.4 Applications of Thermal Imaging

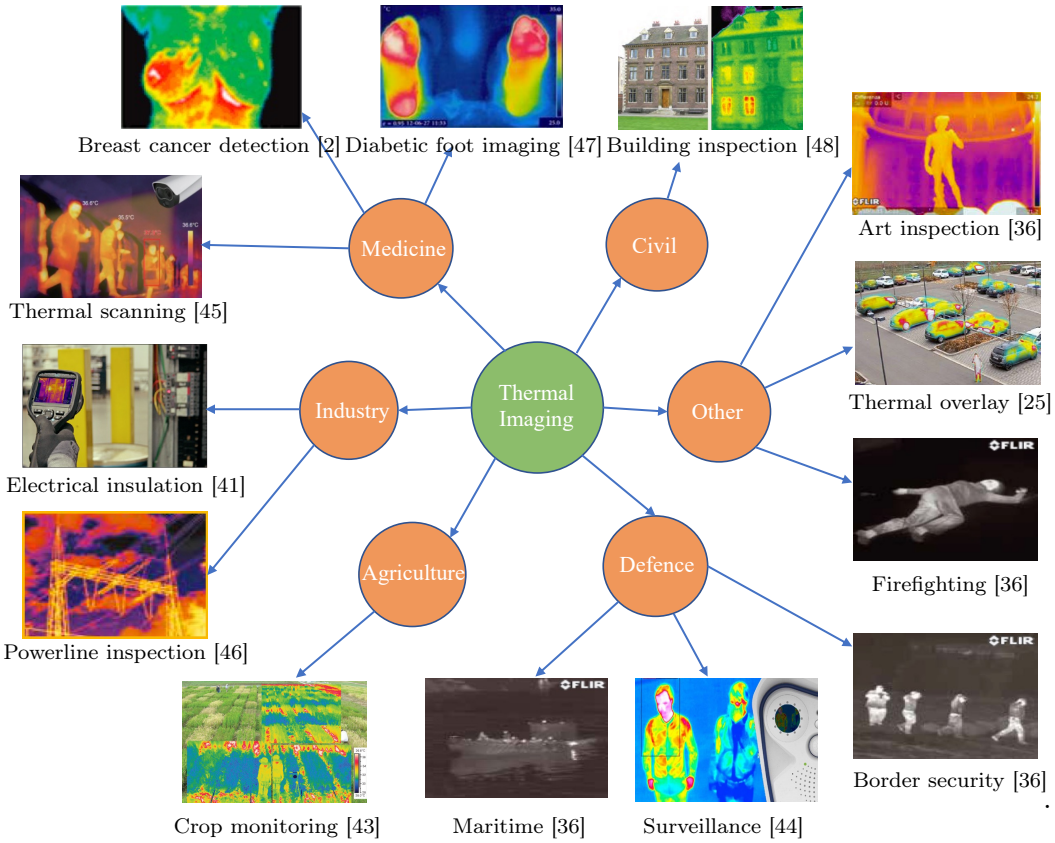


Figure B.4: Applications of thermal imaging

A vast majority of applications involve the use of thermal imagery as shown in Fig B.4. These include detecting cracks in building structures [48], identifying breast cancer [2], surveillance [44], autonomous driving [49], etc. Thermal cameras are favoured primarily because of their ability to obtain useful information through a non-contact non-intrusive manner. High resolution thermal cameras are employed

to detect temperature variations in different parts of the human body which can in turn help in medical diagnosis. The development of a neonatal in intensive care units can be assessed based on the time-dependent thermal variations obtained from thermal imaging. As the method is non-contact and non-invasive it poses reduced risk to neonatals [50]. Thermal imagery also helps to identify irregularities and detect diseases early. Breast cancer identification [51], [52], tumor detection [53], diabetic foot inspection [54], [47], covid-19 screening [55], [45], diabetic eye disease [56], and skin cancer lesions [57] are some areas where thermal imaging has an edge over other traditional methods. Other applications include dental diagnosis where the amount of deposits and activity on the tooth root caries are effectively measured using thermal imaging to make accurate decisions [58].

One of the earliest use of thermal cameras stems from military and defense applications. Thermal cameras can easily identify intruders in the dark proving to be an effective sensing technology in low light environments. Additionally they aid in surveillance [44], detection and tracking of UAVs [59], ship navigation [36], flight landing assistance [60], maintaining border security [61], etc.

Industrial applications for handheld thermal cameras have also gained prominence due to their ease to carry and detect fault and issues. Issues with electrical insulation [41], pipeline rework [46], and power-line inspection [46] are some areas where thermal cameras has helped improve industrial processes. Additionally, they are also used in welding applications to inspect defects [62]. Similarly, in civil and construction, thermal imagery helps to identify air leakages in buildings, defects and cracks in bridge structures [48], etc.

Thermal cameras are used in agriculture for crop monitoring, yield forecasting, and irrigation scheduling [43]. Monitoring field nurseries to detect early diseases in tender seeds using thermal signatures has helped improve yield. In addition to the above applications, thermal cameras also find use in firefighting [63], face de-identification [64], human activity recognition [65], occupancy estimation [66], disaster management [67], thermal overlay [25], etc. Their use in artwork inspection to validate the authenticity and identify defects is also prominent [68]. Thermal fusion along side RGB images have helped in semantic segmentation of urban environments in order to assist autonomous vehicles [69]. Recent advances in semiconductor technology coupled with enhanced computing capabilities and machine learning algorithms have also helped to explore new applications for thermal cameras. Some of these will be covered in the subsequent section.

B.5 Machine Learning Techniques for Thermal Imaging Applications

In recent years, thermal imaging coupled with machine learning techniques has gained traction. Thermal images provide the temperature gradient of the captured scene. Any fault or anomaly in a system or device is associated with a change in its temperature profile. By utilizing state-of-the-art machine learning techniques

along with thermal imaging, these anomalies can be easily detected and inferred in a contactless and non-invasive manner. Moreover, the ability to perform highly scalable operations on large datasets have also added to the popularity in using thermal imaging with machine learning techniques.

In electric power industry, identifying equipment faults early from the temperature distribution of thermal images can help prevent equipment failure, fire hazards and other potential risks. In [70], Ying *et al.* addresses the problem of incorrect detection of equipment parts with different orientation from hand-held thermal camera images. The authors propose a cascaded two-stage spatial transform network (STN) that is fed into faster region-based convolutional neural networks (R-CNN) to identify the required rotation transformation. Training is performed separately for the two stages and then further end-to-end fine tuning is performed to achieve detection with large orientation angles. The proposed approach outperforms current state-of-the-art methods including the oriented you only look once (YOLO) algorithm with a higher mAP value. Another work in [71], utilizes thermal imaging to characterize the condition of a machine. The authors use two convolutional neural networks (CNNs) along with the Zeiler method [72] to obtain useful insights from the thermal image. One of the CNN is used to extract the spatial aspects from the roll bearing element and the other CNN infers the gradation of imbalance using the extracted temporal features. The proposed system is able to obtain 91.6% and 95% accuracy for rotating machinery use cases in machine-fault detection and oil-level prediction applications. Another work by Choudhary *et al.* [73], focuses on detecting bearing faults in induction motors using thermal imaging. The performance of traditional feature extraction methods fail due to insignificant information and string noise from the thermal noise. The authors in this work thus use a two dimensional discrete wavelet transform (2D-DWT) along with principal component analysis (PCA) to resolve this issue. The extracted features were then arranged according to the Mahalanobis distance to select the optimal features. Classification is performed using support vector machines (SVM), linear discriminant analysis (LDA), and complex decision trees (CDT). Reported results show that SVM obtained higher classification accuracy as compared to other techniques. The work by Ogbechie *et al.* [74] uses dynamic bayesian networks for anomaly detection in laser surface heat treatment process. The proposed approach uses a NIT Tachyon 1024 thermal camera to obtain images of the heat treatment process. After the necessary preprocessing and feature selection, the data is trained using two different types of dynamic bayesian networks with a k-fold cross validation. An anomaly score is calculated that is used to identify and detect anomalies in the laser heat process.

Other fault detection and classification areas where thermal images combined with machine learning prove useful are in photovoltaic systems [75]. Photovoltaic systems are vulnerable to various defects such as encapsulant defects, back sheet defects, cracket cell, and faulty interconnections. In this work, the authors have initially performed a texture feature analysis for the different faulty panels using grey-level co-occurrence matrices (GLCMs). The extracted features are then trained using artifical neural networks to classify the faults. The new approach exhibits

93.4% training and 91.7% testing efficiency respectively. It is also reported that the proposed approach outperforms other conventional techniques such as SVMs and k-nearest neighbours (kNNs) by a significant margin.

Machine learning is being extensively used in the medical domain to the minimize the manual decision making which can led to errors. Machine learning models once trained with the thermal images can be used to predict and detect tumors. Early cancer identification using non-invasive techniques with thermal imagery helps reduce fatality rate. In [82], Karthiga and Narasimhan, study various machine learning classification techniques to best extract the features to display these cancer symptoms. Thermal imagery is initially preprocessed using top-hat and bottom-hat transforms. The resulting image structure is segmented using morphological operations to yield various statistical, geometrical, and intensity features. Further processing using GLCM is performed to obtain texture features. The texture features in the spatial domain are classified using various machine learning techniques. The cubic SVM shows the most promising accuracy with 93.3% as compared to other techniques such as kNN, decision tree classifier, and logistic regression. In [79], the authors use thermal camera and heart rate sensors to study the time delay associated with various physiological functions of the body. The thermal camera provides facial images which is processed using a two-layer artificial neural network (ANN) in order to predict the actual change in breathing temperature. Additional adaptive algorithms are also employed with the heart rate measurements to accurately estimate the temperature. Results show that the time delay associated with the drop in heart rate and breathing frequency corroborate with real world measurements obtained from heart rate sensors of cyclists.

Other applications of thermal imagery coupled with machine learning include detection of damaged pavements as studied in [80]. In this work, the authors use a pre-trained EfficientNet B4 fusion architecture to combine thermal and RGB images to detect pavement damages. An argument dataset is also generated by addition of camera noise, non-uniform illumination, and other parameters to replicate real-world pavement damages and scenarios. Experiments carried out with images from individual RGB sensors, thermal cameras, and the fused images show that the fused thermal-RGB image provides better prediction accuracy of about 98.34%. The fused images are capable of providing reliable detections for various cracks such as alligator, longitudinal, and transverse along with pothole categories.

In [81], the authors develop a novel approach to enable semantic segmentation for thermal images by introducing a gated featurewise transform layer to the proposed edge-conditioned convolutional neural network (EC-CNN) architecture. Low resolution thermal images are affected by thermal crossover and imaging noise that makes detecting object boundaries challenging. To overcome this issue, the authors utilize hierarchical edge features obtained by training RGB images. The trained model is then fed to the proposed semantic segmentation network that is based on DeepLabv3 [88]. The reported results show that the proposed method outperforms traditional methods for thermal image semantic segmentation. Additionally, the authors have also provided with a manually annotated thermal image dataset (SODA)

Table B.6: Machine learning techniques for thermal imaging

Year	Method	Application	References	Comments
2016	Bayesian networks	Firefighting	[76]	1. Combination of mean, dissimilarity, correlation, skewness, and standard deviation produced highest performance results. 2. Usage of CNN and discrete wavelet transforms can be explored.
2016	K* algorithm	Occupancy estimation	[77]	1. Entropy-based algorithms exhibited excellent performance. 2. Can improve classification performance using recent occupancy history.
2017	Dynamic bayesian networks	Anomaly detection	[74]	1. Obtained 90% specificity with greater than 80% sensitivity. 2. Can improve process parameter values. Online implementation for timely feedback can also be a potential future work.
2017	Dynamic bayesian networks	Occupancy estimation	[78]	1. Three order reduction in memory requirement and 8x reduction in memory utilization. 13x improvement in execution time. 2. Future extension can include exploring the performance with multiple PIR sensors.
2018	STN, Faster R-CNN	Equipment detection	[70]	1. Accurate identification of rotated equipments. 2. Future extension can be online implementation for live detection.
2018	CNN	Condition monitoring of machines	[71]	1. More than 90% accuracy in real-world applications. 2. Online condition monitoring for offshore wind turbines can be future work.
2018	ANNs	Breathing frequency estimation	[79]	1. Results corroborate with real values obtained from cycling expeditions. 2. Usage of CNNs and parallel algorithms for future work.
2019	ANNs	Identifying faults in photovoltaic systems	[75]	1. Fast detection time along with remote monitoring. Enhanced classification accuracy of 92.8% with ANNs as compared to SVMs and kNNs. 2. Online methods and CNNs can be explored.
2019	Deep RNN based LSTM	Human activity recognition	[65]	1. HAR systems with RNNs offer 96% accuracy as compared to other methods. 2. Performance with bright images, oriented/tilted subjects, etc can also be explored.
2020	EfficientNet B4	Pavement defect detection	[80]	1. Excellent pavement defect detection performance compared to other methods. Additionally, the proposed method outperforms detection as compared to RGB images. 2. Prediction performance is low for road marking, manholes, and shadow as compared to RGB.

2020	Gated featurewise transform with EC-CNN	Thermal semantic segmentation	[81]	1. Proposed method performs better than DeepLabv3 with 4.8% gain. Resonable performance under thermal crossover. 2. Incorporate transfer learning from RGB data and including more annotated thermal images can be future work.
2020	Random forests and ensemble learning	Face recognition and de-identification	[64]	1. Ensemble learning can improve the prediction performance. 2. Fusion techniques can be investigated.
2021	SVM, LDA, CDT	Induction motor fault detection	[73]	1. SVM outperformed LDA and CDT for bearing defect classification. 2. Can increase cases for bearing defects.
2021	Cubic SVM	Breast cancer identification	[82]	1. Cubic SVM classifier produced highest accuracy of 93.9% as compared to other classifiers. 2. The work can be extended to include other views such as right, left, and lateral to improve accuracy.
2021	CNN	Hand gesture classification	[83]	1. Lightweight CNNs with high accuracy of 99.52% 2. Can extend for hand gesture recognition in complex backgrounds.
2021	ResNet-50 with random forest	Pollution identification on MOSA	[84]	1. Proposed approach uses ResNet50 with random forest to achieve mean recognition accuracy of 98.04%. 2. Experiments on real MOSA samples can be performed.
2021	Random forest	Occupancy estimation	[66]	1. Uses blob filtering algorithm to distinguish humans close to each other. Achieves 99% accuracy. 2. Sensor fusion, different deployment configurations, transient heat conditions, CNNs and RNNs can be explored.
2021	Deep convolutional encoder-decoder network with adaptive boost	Occupancy estimation	[85]	1. Privacy-friendly, low-resolution TSA sensing technique with 98.43% classification accuracy. 2. Estimating human-sensor distance using entropy point methods can be potential future work.
2021	ResNet encoder-decoder network with pyramid supervision training	Semantic segmentation in snowy environment	[86]	1. Proposed network achieved 78% mIoU and has become state-of-the-art for snowy environments. 2. Dataset classes can include additional classes such as animals and maintenance holes. Integrate other sensors for future work.
2022	ResNet-50 encoder decoder model with transformer based fusion	Glass segmentation	[87]	1. Proposed RGB-T network outperformed existing methods with 93.8% IoU for images with glass. 2. Can combine with polarization to improve result.

for further research into thermal semantic segmentation. Another interesting work by Dong *et al.* in [87], utilizes the transmission properties of infrared radiation to

segment images containing glass elements. As glass is transparent to visible light, traditional methods for RGB images fail to effectively detect and segment regions containing glass. The architecture in the proposed work is made up of two independent ResNet-50 networks that act as the encoder stage for extracting high-level features from both the RGB and thermal images. These features are then combined using a transformer-based fusion module. The result is then fed into a decoder for obtaining the desired segmentation output. Qualitative and quantative evaluations have shown that the proposed approach outperforms current state-of-the-art techniques and effectively segments glass components in images. However, the approach still requires further work to classify in polarized image conditions.

Occupancy estimation is another potential application based on thermal imagery. In [66], the authors provide a comparative study of various low resolution thermal sensors, GridEye, MLX90640, and Lepton that can be used to provide highly accurate real-time occupancy estimation. The proposed approach involves a unified algorithm pipeline that involves noise reduction, bilinear interpolation, blob filtering to distinguish multiple people close to each other, and connected component analysis to obtain the best possible results. The output from this pipeline is fed into a novel feature vector design that is used in conjunction with classification algorithms to classify target occupants. Classification algorithms include random forests, gaussian naive bayes, kNNs, and SVM. Experiment results have reported that the random forests algorithm exhibits 99% accuracy. In comparison to the above, the authors in [77], use a low-pixel count 4 x 16 thermal detector array to perform occupancy detection. The thermal detector along with RGB camera is mounted on the Raspberry Pi to obtain the images. The RGB camera served as the ground truth occupancy values. Background separation algorithm coupled with a slow-moving exponential weighted moving average (EMWA) accompanied the preprocessing stage before feature extraction. Three features, number of active pixels, number of connected components, and size of the largest connected component were identified to be used with the classification algorithms. The Weka toolchain [89] was used to compare different machine learning classification algorithms. It was found that entropy-based algorithms such K* gave the best performance with an accuracy of 82.56% and a root mean square error of 0.304. Another approach followed by Naser *et al.* in [85], uses an array of thermal sensors at different locations of the room to perform human segmentation and occupancy estimation. In this work, the authors have proposed a deep convolutional encoder-decoder network for human segmentation from the thermal images. Residual thermal signatures are further eliminated during the post processing using connectivity filters. To classify and also to determine occupancy estimation, the output is then fed to an adaptive boosting algorithm. The adaptive boosting approach provides accuracy of 98.43% from vertical and 100% from overhead sensor locations. Another work by Charles *et al.* [78] uses a bayesian machine learning algorithm on a resource constrained ARM Cortex-Mo based ST Nucleo-F070RB board to estimate the room occupancy using a single analogue passive infrared (PIR) sensor. The proposed algorithm uses infinite hidden markov model (iHMM) with laplace components on the raw PIR data for

segmentation. The segmented data is now manipulated to estimate the laplace diversity which provides an indication of the room occupancy. Reported results show that a moderately high-performance microcontroller is able to house the occupancy algorithm while providing real-time performance and reduced power consumption.

Classification of hand gestures for sign language digits can also be performed using thermal imaging as demonstrated in [83]. Daniel *et al.* demonstrates an end-to-end edge computing system based on light weight CNNs that can classify thermal images of different hand gestures. The proposed approach utilizes a 3200 thermal image dataset to train a model that is based on bottleneck layers. The model is deployed on a Raspberry Pi and the developed system achieves 99.52% accuracy. Furthermore, the proposed approach is compared with the Big Transfer (BiT) model to report approximately a 20% improvement in accuracy. For human activity recognition (HAR), the authors in [65] extend the OpenPose approach to extract body joints from human thermal images for activity recognition. The proposed approach utilizes OpenPose and subsequently performs a spatiotemporal feature extraction. Discriminant analysis is performed on the extracted features followed by a deep recurrent neural network (RNN) based long short-term memory (LSTM) to better retain the embedded time-sequential information. The novel approach is reported to outperform other techniques such as hidden markov models (HMM), deep belief networks (DBN), CNN and RNNs. Thermography is also used for face recognition and de-identification [64]. Normal RGB images can easily deceive face recognition systems as they work only on identifying the extracted features. The authors in this work use additional features extracted from thermal images such as feature matrix and feature image along with random forests and ensemble learning to improve prediction accuracy and better facial de-identification. This can help in preventing erroneous face recognition with use of facial images. Another interesting work in [76] uses thermal imagery in firefighting situations to identify local conditions and decide proper navigation course through a fire or smoke prone area. High temperature regions are separated using the Otsu method which is then fed into a bayesian classifier to probabilistically detect multiple classes during real-time implementation. Further, a multi-objective genetic algorithm using resubstitution and cross-validation errors is also used to find the best combination of features to obtain the lowest error and highest performance.

Improving road safety in snowy environments is yet another application in which thermal imagery can be beneficial. The authors in [86], have developed a multimodal RGB-thermal fusion model for semantic segmentation of roads in snow filled environments. The architecture utilizes a convolutional encoder-decoder fusion network where the encoder is based on a fully pre-activated ResNet-50 model [90] to maintain good tradeoff between computation and feature learning. Both the RGB and thermal images are fed into the encoder module that is followed by a atrous spatial pyramid pooling (ASPP) which is used to incorporate image features. The ASPP's output is fused and then fed into a ResNet-34 based decoder module. Additional pyramid supervision training scheme [91] is also employed to improve training accuracy. The proposed method has obtained a 78% mIoU outperforming state-of-

the-art network for snowy environments. Thermal imaging coupled machine learning is also used to identify the severity of pollution on metal oxide surge arrester (MOSA) [84]. The proposed approach utilizes thermal images of MOSA at various pollution levels, which is fed into pretrained 'ResNet50' architecture for feature extraction. The extracted features are then given to various classifiers such as kNN, SVM, naive Bayes and random forest. Random forest gave the best accuracy along with fast inference time. The authors were also able to validate the practicality of the proposed approach by experimenting on 11kV MOSA which also gave accurate results. A brief summary of the various machine learning techniques used for thermal imaging can be found in Table B.6.

B.6 Conclusion and Future work

Technological innovations in the semiconductor industry and along with other advancements has made thermal imaging applications more accessible and prevalent. Novel machine learning algorithms in thermal imaging applications have shown to provide better performance as compared to their traditional counterparts. Thermal fusion has shown to be used in various applications scenarios and is still an active research topic. Combining polarization properties in thermal fusion algorithms containing glass segments is still unexplored. Further, it is shown above that offline algorithms for thermal data provides good performance. However, their online implementations require further research. In object detection and classification, eliminating the thermal retention signatures is also crucial to improve performance. Further, the current algorithms utilize existing implementations for normal RGB with some variations. To obtain better performance and efficient resource utilization, devising or modifying the algorithms to incorporate the thermal characteristics of the scene would be highly beneficial.

In this article, the most recent developments in thermal imaging are reviewed. The key specifications of the most recent thermal imaging devices have been discussed. The use of thermal imaging in a wide range of disciplines and scenarios was discussed. Finally, machine learning techniques for thermal imaging were discussed along with possible future work. This article is useful as a reference guide for designing and implementing thermal imaging-based systems and/or applications.

References

[1] Amanda Berg. *Detection and tracking in thermal infrared imagery.* PhD thesis, Linköping University Electronic Press, 2016.

[2] P Rajmanova, P Nudzikova, and D Vala. Application and technology of thermal imagine camera in medicine. *IFAC-PapersOnLine*, 48(4):492–497, Sept. 2015.

[3] Rafael C Gonzalez, Richard E Woods, et al. Digital image processing second edition. *Beijing: Publishing House of Electronics Industry*, 455, 2002.

[4] Megger - Introduction to thermal imaging. [Online]. Available: https://d20g1hcwzqzdjk.cloudfront.net/sites/www.voltimum.pt/files/fields/attachment_file/introduction-thermography-en_v01.pdf.

[5] Nickolay I Hristov, Margrit Betke, and Thomas H Kunz. Applications of thermal infrared imaging for research in aeroecology. *Integrative and Comparative Biology*, 48(1):50–59, Jul. 2008.

[6] Shan Lin. *Monitoring of Thermal Processes for Medical Applications Using Infrared Thermography.* PhD thesis, 2017.

[7] JCB Marins et al. Applications of infrared thermography in sports. A review. *International Journal of Medicine and Science of Physical Activity and Sport, in press*, 15(60):805–824, 2013.

[8] B. Wiecek. Review on thermal image processing for passive and active thermography. In *Proc. IEEE Engineering in Medicine and Biology*, pages 686–689, Shanghai, China, 2005.

[9] Babak Kateb et al. Infrared thermal imaging: a review of the literature and case report. *NeuroImage*, 47:T154–T162, Mar. 2009.

[10] Roselyne Ishimwe et al. Applications of thermal imaging in agriculture—a review. *Advances in remote Sensing*, 3(03):128, Sep. 2014.

[11] Wang Yongqing et al. The temperature measurement technology of infrared thermal imaging and its applications review. In *Proc. IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, pages 401–406, Yangzhou, China, 2017.

[12] Om M. Rajpurkar et al. A survey on engagement and emotion analysis in theatre using thermal imaging. In *Proc. International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 905–911, Coimbatore, India, 2020.

[13] Aayesha Hakim and RN Awale. Thermal imaging - An emerging modality for breast cancer detection: A comprehensive review. *Journal of Medical systems*, 44(8):1–18, Jul. 2020.

[14] Tran Xuan Bach Nguyen, Kent Rosser, and Javaan Chahl. A review of modern thermal imaging sensor technology and applications for autonomous aerial navigation. *Journal of Imaging*, 7:217, 10 Oct. 2021.

[15] A.A. Gowen et al. Applications of thermal imaging in food quality and safety assessment. *Trends in Food Science & Technology*, 21(4):190–200, Jan 2010.

[16] KEMET QMS Pyroelectric IR Motion Sensor. [Online]. Available: https://content.kemet.com/datasheets/KEM_SE0212_QMS.pdf.

[17] A.W. Van Herwaarden and P.M. Sarro. Thermal sensors based on the seebeck effect. *Sensors and Actuators*, 10(3):321–346, 1986.

[18] Omron D6T MEMS Thermal Sensor - Datasheet. [Online]. Available: https://components.omron.com/us-en/ds_related_pdf/A284-E1.pdf.

[19] FLIR Lepton 3.5 Thermal Camera. [Online]. Available: https://www.flir.eu/products/lepton/?model=500-0771-01.

[20] PYROSENS - Datasheet. [Online]. Available: http://www.dias-infrared.de/pdf/pyrosens_arrays_eng_mail.pdf.

[21] Seek Thermal InspectionCAM - Datasheet. [Online]. Available: https://www.thermal.com/uploads/1/0/1/3/101388544/inspectioncam_-_spec_sheet_v3.pdf.

[22] Evo Thermal 90 - Datasheet. [Online]. Available: https://terabee.b-cdn.net/wp-content/uploads/2020/10/TeraRanger-Evo-Thermal-Specsheet-1.pdf.

[23] Christopher Gobbi. Low cost thermal imaging system for welding applications. Master's thesis, University of Waterloo, 2016.

[24] Gianluca Guglielmo and Michal Klincewicz. The temperature of morality: A behavioral study concerning the effect of moral decisions on facial thermal variations in video games. In *Proc. Foundations of Digital Games (FDG)*, pages 1–4, New York, NY, USA, 2021.

[25] MOBOTIX Thermal solutions. [Online]. Available: https://d347awuzx0kdse.cloudfront.net/vsp/content-file/mx_thermal_technology_en_200406_web.pdf.

[26] Compact Thermal Camera FLIR C5. [Online]. Available: https://www.flir.eu/products/c5/?vertical=condition%20monitoring&segment=solutions.

[27] FLUKE - Fluke Ti480 PRO Infrared Camera. [Online]. Available: https://www.fluke.com/en/product/thermal-cameras/ti480-pro.

[28] SEEK Thermal - CompactPRO XR. [Online]. Available: https://www.thermal.com/compact-series.html.

[29] PULSAR - HELION 2 XP PRO Thermal imaging monoculars. [Online]. Available: https://www.pulsar-nv.com/glo/products/33/thermal-imaging-scopes/helion-version-2/.

[30] CAT - Cat® S62 Pro. [Online]. Available: https://www.catphones.com/en-gb/cat-s62-pro-smartphone/.

[31] PULSAR - MERGER LRF Thermal imaging binoculars. [Online]. Available: https://www.pulsar-nv.com/glo/products/33/thermal-imaging-binoculars/merger/.

[32] FLUKE - Fluke RSE600 mounted infrared camera. [Online]. Available: https://www.fluke.com/en/product/thermal-cameras/rse600.

[33] MOBOTIX M16 Thermal TR - Datasheet. [Online]. Available: https://www.mobotix.com/sites/default/files/2022-08/Mx_TS_M16B_V1.05_EN_20220811.pdf.

[34] MOBOTIX S16 DualFlex - Datasheet. [Online]. Available: https://www.mobotix.com/sites/default/files/2022-08/Mx_TS_S16B_V1.05_EN_20220811.pdf.

[35] FLIR thermal imaging cameras for predictive maintenance. [Online]. Available: https://assets.tequipment.net/assets/1/7/FLIR-Predictive-Maintenance-Brochure-7038_EN.pdf.

[36] FLIR Imaging for building diagnostics. [Online]. Available: https://www.flirmedia.com/MMC/THG/Brochures/T820484/T820484_EN.pdf.

[37] MOBOTIX - MOBOTIX M16 Thermal. [Online]. Available: https://www.mobotix.com/en/products/thermographic-cameras/m16-thermal.

[38] MOBOTIX - MOBOTIX M16 Thermal TR. [Online]. Available: https://www.mobotix.com/en/products/thermographic-cameras/m16-thermal-tr.

[39] MOBOTIX - MOBOTIX S16 Dual Flex. [Online]. Available: https://www.mobotix.com/index.php/en/products/outdoor-cameras/s16-dualflex.

[40] MOBOTIX - MOBOTIX S16 Thermal TR. [Online]. Available: https://www.mobotix.com/index.php/en/products/thermographic-cameras/s16-thermal-tr.

[41] FLIR 12 Things to know. [Online]. Available: https://www.flir.com/landing/instruments/12-things-to-know-before-buying-a-thermal-camera/.

[42] Thermal imaging for Science/R&D. [Online]. Available: https://www.flirmedia.com/MMC/THG/Brochures/T820486/T820486_EN.pdf.

[43] Hamlyn G Jones et al. Thermal infrared imaging of crop canopies for the remote diagnosis and quantification of plant responses to water stress in the field. *Functional Plant Biology*, 36(11):978–989, Nov. 2009.

[44] Surveillance Secure. [Online]. Available: https://surveillancesecure.com/thermal-surveillance-cameras-provide-proactive-warning-with-temperature-imaging-technology/.

[45] Platinum CCTV - Skin temperature measurement. [Online]. Available: https://platinumcctv.com/bf3221-skin-temp-thermal-camera.

[46] FLIR Thermal imaging guidebook for industrial applications. [Online]. Available: https://www.flirmedia.com/MMC/THG/Brochures/T820264/T820264_EN.pdf.

[47] Shazia Shaikh, Nazneen Akhter, and Ramesh Manza. Current trends in the application of thermal imaging in medical condition analysis. *Int. J. Innov. Technol. Explor. Eng*, 8(8):2708–2712, Jun. 2019.

[48] Indrasenan Thusyanthan, Tim Blower, and William Cleverly. Innovative uses of thermal imaging in civil engineering. In *Proceedings of the Institution of Civil Engineers-Civil Engineering*, volume 170, pages 81–87, Apr. 2017.

[49] Ben Miethig et al. Leveraging Thermal Imaging for Autonomous Driving. In *Proc. IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 1–5, Detroit, MI, USA, 2019.

[50] Duygu Savaşci and Murat Ceylan. Thermal image analysis for neonatal intensive care units (First evaluation results). In *Proc. Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, Izmir, Turkey, 2018.

[51] Sebastien Jean Mambou, Petra Maresova, Ondrej Krejcar, Ali Selamat, and Kamil Kuca. Breast cancer detection using infrared thermal imaging and a deep learning model. *Sensors*, 18(9), Aug. 2018.

[52] Volkan Tanrıverdi and Nevzat G. Gençer. Induced current thermal imaging in breast cancer detection. In *Proc. Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, Istanbul, Turkey, 2021.

[53] Shazia Shaikh et al. Segmentation of thermal images using thresholding-based methods for detection of malignant tumours. In *Proc. Intelligent Systems Technologies and Applications*, pages 131–146, Jaipur, India, 2016.

[54] Susan Quinn et al. A thermal imaging solution for early detection of pre-ulcerative diabetic hotspots. In *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1737–1740, Berlin, Germany, 2019.

[55] Rafael Y Brzezinski et al. Automated processing of thermal imaging to detect COVID-19. *Scientific Reports*, 11(1):1–10, Sep. 2021.

[56] D. Selvathi and K. Suganya. Support vector machine based method for automatic detection of diabetic eye disease using thermal images. In *Proc. Innovations in Information and Communication Technology (ICIICT)*, pages 1–6, Chennai, India, 2019.

[57] Sebastián E Godoy et al. Detection theory for accurate and non-invasive skin cancer diagnosis using dynamic thermal imaging. *Biomedical optics express*, 8(4):2301–2323, Apr. 2017.

[58] V Yang et al. Thermal imaging of root caries in vivo. *Journal of dental research*, 99(13):1502–1508, Aug. 2020.

[59] Fredrik Svanström, Cristofer Englund, and Fernando Alonso-Fernandez. Real-time drone detection and tracking with visible, thermal and acoustic sensors. In *Proc. International Conference on Pattern Recognition (ICPR)*, pages 7265–7272, Milan, Italy, 2021.

[60] Tomasz Sosnowski et al. Thermovision system for aircraft landing. *Measurement Automation Monitoring*, 61, 2015.

[61] Dawoud ALshukri et al. Intelligent border security intrusion detection using IoT and embedded systems. In *Proc. MEC International Conference on Big Data and Smart City (ICBDSC)*, pages 1–3, Muscat, Oman, 2019.

[62] Andrea Fernández et al. Embedded vision system for monitoring arc welding with thermal imaging and deep learning. In *Proc. International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–6, Barcelona, Spain, 2020.

[63] Manish Bhattarai and Manel Martinez-Ramon. A deep learning framework for detection of targets in thermal images to improve firefighting. *IEEE Access*, 8:88308–88321, May 2020.

[64] Chih-Hsueh Lin, Zhi-Hao Wang, and Gwo-Jia Jong. A de-identification face recognition using extracted thermal features based on deep learning. *IEEE Sensors Journal*, 20(16):9510–9517, Apr. 2020.

[65] Md. Zia Uddin, Weria Khaksar, and Jim Torresen. A thermal camera-based activity recognition using discriminant skeleton features and RNN. In *Proc. International Conference on Industrial Informatics (INDIN)*, volume 1, pages 777–782, Helsinki, Finland, 2019.

[66] Veena Chidurala and Xinrong Li. Occupancy estimation using thermal imaging sensors and machine learning algorithms. *IEEE Sensors Journal*, 21(6):8627–8638, Jan. 2021.

[67] AV Arunraj et al. An IoT application in disaster management using real-time thermal imaging system. In *AIP Conference Proceedings*, volume 2222, page 030025, Kollam, India, 2020.

[68] Massimo Rippa et al. Active thermography for non-invasive inspection of an artwork on poplar panel: Novel approach using principal component thermography and absolute thermal contrast. *Journal of Nondestructive Evaluation*, 40(1):1–9, Feb. 2021.

[69] Yuxiang Sun, Weixun Zuo, and Ming Liu. RTFNet: RGB-Thermal fusion network for semantic segmentation of urban scenes. *IEEE Robotics and Automation Letters*, 4(3):2576–2583, Mar. 2019.

[70] Ying Lin et al. A cascaded spatial transformer network for oriented equipment detection in thermal images. In *Proc. IEEE Conference on Energy Internet and Energy System Integration (EI2)*, pages 1–5, Beijing, China, 2018.

[71] Olivier Janssens et al. Deep learning for infrared thermal image based machine health monitoring. *IEEE/ASME Transactions on Mechatronics*, 23(1):151–159, Jul. 2018.

[72] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. ECCV*, pages 818–833, Zurich, Switzerland, 2014.

[73] Anurag Choudhary, Deepam Goyal, and Shimi Sudha Letha. Infrared thermography-based fault diagnosis of induction motor bearings using machine learning. *IEEE Sensors Journal*, 21(2):1727–1734, Aug. 2021.

[74] Alberto Ogbechie, Javier Díaz-Rozo, Pedro Larrañaga, and Concha Bielza. Dynamic bayesian network-based anomaly detection for in-process visual inspection of laser surface heat treatment. In Jürgen Beyerer, Oliver Niggemann, and Christian Kühnert, editors, *Machine Learning for Cyber Physical Systems*, pages 17–24, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

[75] V S Bharath Kurukuru et al. Fault classification for photovoltaic modules using thermography and machine learning techniques. In *Proc. International Conference on Computer and Information Sciences (ICCIS)*, pages 1–6, Sakaka, Saudi Arabia, 2019.

[76] Jong-Hwan Kim, Seungsik Jo, and Brian Lattimer. Feature selection for intelligent firefighting robot classification of fire, smoke, and thermal reflections using thermal infrared images. *Journal of Sensors*, 2016, 09 2016.

[77] Ash Tyndall, Rachel Cardell-Oliver, and Adrian Keating. Occupancy estimation using a low-pixel count thermal imager. *IEEE Sensors Journal*, 16(10):3784–3791, Feb. 2016.

[78] Charles Leech, Yordan P. Raykov, Emre Ozer, and Geoff V. Merrett. Real-time room occupancy estimation with bayesian machine learning using a single pir sensor and microcontroller. In *2017 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, 2017.

[79] Aleš Procházka et al. Machine learning in rehabilitation assessment for thermal and heart rate data processing. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(6):1209–1214, Apr. 2018.

[80] Cheng Chen et al. Deep learning-based thermal image analysis for pavement defect detection and classification considering complex pavement conditions. *Remote Sensing*, 14(1), Dec. 2021.

[81] Chenglong Li et al. Segmenting objects in day and night: Edge-conditioned CNN for thermal image semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 32(7):3069–3082, Jul. 2021.

[82] R Karthiga and K Narasimhan. Medical imaging technique using curvelet transform and machine learning for the automated diagnosis of breast cancer from thermal image. *Pattern Analysis and Applications*, 24(3):981–991, Feb. 2021.

[83] Daniel S. Breland et al. Deep learning-based sign language digits recognition from thermal images with edge computing system. *IEEE Sensors Journal*, 21(9):10445–10453, Feb. 2021.

[84] Arup Kumar Das et al. A transfer learning approach to sense the degree of surface pollution for metal oxide surge arrester employing infrared thermal imaging. *IEEE Sensors Journal*, 21(15):16961–16968, May 2021.

[85] Abdallah Naser, Ahmad Lotfi, and Junpei Zhong. Adaptive thermal sensor array placement for human segmentation and occupancy estimation. *IEEE Sensors Journal*, 21(2):1993–2002, Jan. 2021.

[86] Sirawich Vachmanus et al. Multi-modal sensor fusion-based semantic segmentation for snow driving scenarios. *IEEE Sensors Journal*, 21(15):16839–16851, May 2021.

[87] Dong Huo et al. Glass segmentation with RGB-thermal image pairs. *arXiv preprint arXiv:2204.05453*, 2022.

[88] DeeplabV3. [Online]. Available: https://pytorch.org/hub/pytorch_vision_deeplabv3_resnet101/.

[89] Weka. [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/.

[90] Kaiming He et al. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, Nevada, USA, 2016.

[91] Jindong Jiang et al. RedNet: Residual encoder-decoder network for indoor RGB-D semantic segmentation. *arXiv preprint arXiv:1806.01054*, 2018.

Appendix C

PAPER C

Title: Estimation of UAV Count Using Thermal Imaging and Lightweight CNN

Authors: **W. A. Nelson**, A. Jha, A. Kumar, L. R. Cenkeramaddi

Conference: IEEE ICCMA

Estimation of UAV Count Using Thermal Imaging and Lightweight CNN

W. A. Nelson, A. Jha, A. Kumar, L. R. Cenkeramaddi

Abstract: Illegal and improper use of UAVs has damaged public property and challenged the safety and security of the civilian population. Due to their small form factor, UAVs are undetectable using conventional aircraft detecting methods. In this work, we have addressed this issue by utilizing thermal images to detect and estimate the UAV count in a multi-UAV setting. Thermal imaging-based detection provides a number of advantages, including night vision, temperature sensitivity, low visibility, camouflage penetration, and non-invasiveness. It is a non-contact and non-intrusive detecting method that can detect hidden objects or people even in low-visibility environments such as smoke and fog. Experiments were carried out by capturing thermal images of a multi-UAV setting, where an arbitrary number of UAVs are flying in a random manner. Further, a UAV-thermal dataset is also developed so as to facilitate further research. Extensive experiments were carried out and the reported results show that the proposed model accurately estimates the number of UAVs with an accuracy 99.9%**.**

C.1 Introduction

Advancements in unmanned aerial vehicles (UAV) technology have sparked interest in utilizing UAVs for various applications such as package delivery, maintaining law and order, surveying, disaster management, defense, etc [1]. However, in recent times, the illegal and improper use of UAVs has elevated a growing concern to detect, monitor, and track UAV-related activities [2]. In this context, techniques to identify and detect illegal and intruder UAVs are of prime significance. There are a lot of works in the literature which address this issue using various sensors and techniques. Some of the notable ones include using acoustic sensors [3], RGB images [4], radar modules [5], etc. However, in this work, we focus our attention on using thermal imaging for UAV detection. The primary motivation for using a thermal imaging-based solution can be attributed to the improved visibility that thermal sensors offer in extreme light and dark conditions. Subsequently, thermal sensors are invariant to background noise as compared to acoustic sensors [6], [7].

There have been multiple attempts throughout the literature that use thermal imaging for UAV detection. One such notable work is in [8], where the authors develop a UAV detection and tracking system using deep learning techniques. A data set containing UAVs in visual and thermal modes was generated using adversarial data augmentation methodologies. Additionally, extensive experiments were conducted that revealed that the proposed model performed well on real-world UAV images with complex environments despite being trained on synthetic data. In [9] and [10], the authors develop UAV detection systems to detect obstacles using thermal sensors during the night. Particularly, in [9], the proposed method uses RGB camera and ADS-B signals for validating the obtained results. Results also showed that the detection accuracy was 100% for extreme illumination conditions and during the night in all cases. The authors in [11] used faster-RCNN, a saliency map, and a magnifying small objects (MSO) module to develop a UAV detection system. The thermal saliency map helps to extract meaningful features from the thermal images. Additionally, the MSO module enhances the resolution of small objects before feeding the thermal image to the model. The MSO module thereby increases the accuracy of the model. Reported results indicate more than 93% accuracy for the proposed approach. In [12], the authors investigate the effectiveness of training a UAV detection system using limited thermal images. In order to improve the accuracy of the model, the RGB images containing UAVs were preprocessed to retain characteristics found in thermal imagery. The model was hence trained on preprocessed RGB and limited thermal images. It was reported that the accuracy of the trained model improved significantly based on the type of preprocessing performed on the RGB images.

It can be seen from the above paragraph that most of the literature pertaining to UAV detection using thermal imagery lacks sufficient real-world datasets. Synthetic thermal images are generated from RGB images or other sources to mimic thermal image characteristics. Additionally, the literature focuses on the detection of just a single UAV. In real-world intruder and trespassing scenarios, simultaneous detection of multiple UAVs is crucial to ensure safety and security.

Our work aims to bridge this gap by providing a method to accurately estimate the number of UAVs in a multi-UAV scenario. Additionally, this work will act as a reference for future detection systems to embark on the challenge of detecting multiple UAVs simultaneously. The main contributions of this paper are:

- Proposed a simple cost-effective approach to estimate the count of UAVs in multi-UAV setting by utilizing thermal imaging.
- Developed a lightweight machine learning model that can be used on the edge to estimate the UAV count in a multi-UAV scenario.
- Developed a thermal dataset that contains a total of 10 UAVs flying simultaneously in all directions in a random manner.

The remainder of the paper is organized as follows. Section C.2 provides the problem formulation. Section C.3 explains the proposed method that includes the

Table C.1: Experimental parameters.

Parameters	**Details**
Semicircular area	Radius: 5 meters
Measurement duration	5 minutes
UAV model	**Count**
DJI Mavic 2 Enterprise	1
DJI Mini 2	1
DJI Mini SE	2
DJI Mini 3 Pro	1
Tello EDU	4
SYMA X30	1

Table C.2: Thermal camera specifications.

Parameters	**Value**
Thermal sensitivity (mK)	< 50
Resolution (pixels)	160×120
Spectral Range (microns)	$8 - 14$ (nominal)
Frame Rate (Hz)	8.7 (effective)
Horizontal Field of View (°)	57
Lens Type	f/1.1
Size (mm)	$10.5 \times 12.7 \times 7.14$

experiments, measurement setup, dataset details, and proposed convolutional neural network (CNN) architecture. Section C.4 provides the results obtained after using the proposed approach. Finally, Section C.5 concludes the paper by providing a summary and possible future extensions.

C.2 Problem formulation

UAVs are powered with battery sources to conduct their various operations such as flight, landing, and hovering. These operations dissipate energy. Additionally, the rotation of propellers and other aerodynamic moving parts also results in energy dissipation. A part of the energy dissipated is in the form of heat or thermal energy. The dissipated thermal energy is characteristic of each UAV and depends on various factors such as shape, size, internal components, and battery type of the UAV. Hence, these thermal signatures can be utilized to detect and estimate the UAV count in the captured frame. Hence in this work, the primary objective is to estimate the UAV count from a captured frame by utilizing thermal imaging and machine learning techniques.

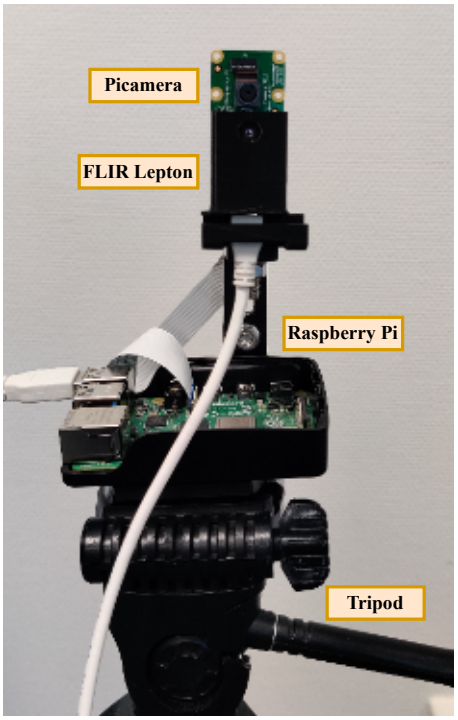


Figure C.1: Experimental setup used for collecting thermal images of UAVs.

C.3 Method

Each UAV has its own characteristic thermal signature. The proposed method exploits these thermal signatures to detect UAVs. The thermal signatures are captured by using a thermal camera, in this case, the FLIR Lepton thermal camera [13]. The captured thermal images are then provided to our custom lightweight CNN model to estimate the UAV count in the captured frame. The precise steps are outlined below:

Step 1: Collect thermal images of various types of UAVs using the FLIR Lepton thermal camera.
Step 2: Preprocess the obtained raw thermal images by resizing, normalizing, grayscale conversion, etc. before feeding to the proposed CNN architecture.
Step 3: Train the processed images with our custom lightweight CNN model to obtain an estimated UAV count in the scene.

C.3.1 Experiment

The experiment was conducted primarily in an indoor lab environment that consisted of a semicircular area of radius 5 meters. Each measurement of the experiment involved capturing thermal images of UAVs flying in the area. The measurements consisted of an arbitrary number of UAVs flying in all directions in a random manner. The measurements were taken in the order of increasing number of UAVs in the scene. Different UAV models such as DJI series [14], Tello EDU series [15], and SYMA series [16] were employed for the measurements. Details regarding the UAV

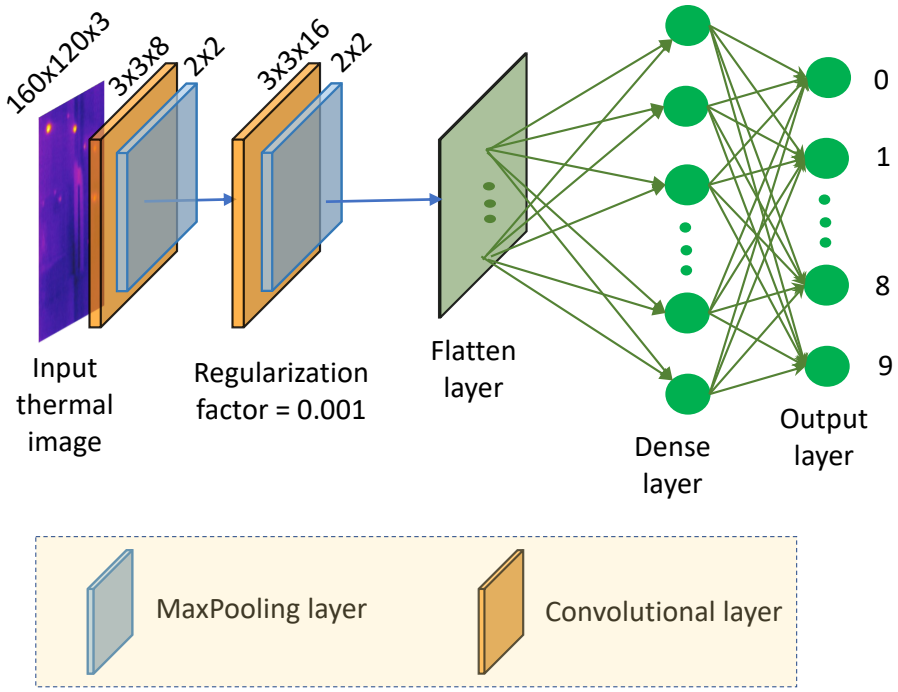


Figure C.2: Proposed CNN architecture for estimating the UAV count from thermal images.

model types can be found in Table C.1. As detecting small-sized UAVs is more challenging, the designed experiments also used Tello EDU UAVs for measurement. All the Tello EDU UAVs were programmed to fly in predefined trajectories. The remaining UAVs however, were operated manually by human control. Additional details regarding the experiment can be found in Table C.1.

The thermal images are captured using the FLIR Lepton 3.5 camera [13]. The FLIR Lepton 3.5 outputs a thermal image that has a resolution of 160×120 pixels with an effective frame rate of 8.7 Hz. The FLIR Lepton offers a nominal spectral range of $8 - 14$ microns with a thermal sensitivity of less than 50 mK. Due to its small size and excellent performance, it serves to be a good candidate to be used in the proposed method. Additional details regarding the thermal camera can be obtained from Table C.2.

The thermal camera is mounted on top of a static tripod as shown in Fig. C.1. The power to the thermal camera is supplied with the help of a USB A connection from the Raspberry Pi Model 4B [17]. A Picamera [18] is also fitted just below the FLIR Lepton. The purpose of the Picamera is to capture RGB images and provide a ground truth for the thermal images. The Picamera is also powered by the Raspberry Pi and operates synchronously with the thermal camera. In addition to powering the two cameras, the Raspberry Pi also serves the purpose of capturing and storing the thermal and RGB images obtained from the FLIR Lepton and Picamera respectively. The Raspberry Pi is installed with Ubuntu 18.04 [19] as the operating system. Ubuntu 18.04 was selected so as to facilitate seamless integration and installation of ROS Melodic[20] and other dependent software packages. ROS Melodic helps to capture data simultaneously from both the FLIR Lepton and Picamera. A portable

Table C.3: Layerwise architecture details of the proposed CNN model

No.	Layer	Output Size	Parameter
1	Input	[(None, 160, 120, 3)]	0
2	Conv2D (1)	[(None, 160, 120, 8)]	224
3	MaxPooling2D (1)	[(None, 80, 60, 8)]	0
4	Conv2D (2)	[(None, 80, 60, 16)]	1168
5	MaxPooling2D (1)	[(None, 40, 30, 16)]	0
6	Flatten	[(None, 19200)]	0
7	Dense	[(None, 10)]	192010

power bank is used to supply power to the Raspberry Pi board.

As mentioned above, each measurement consisted of flying a fixed number of UAVs in all directions. Thermal and RGB images for each of these flight trajectories are captured and stored. A total of 10 measurements are taken where each measurement consisted of 1, 2, 3, etc. up to 10 UAVs being flown in random directions. Each measurement lasted for a total of 5 minutes so as to capture at least 2500 thermal images. Different UAV models were used in these measurements to improve variability. Further, one measurement was conducted in an outdoor environment to introduce noise and additional variability to the data set. This in turn provides a way to understand how the proposed method performs in outdoor scenarios.

C.3.2 Dataset Details

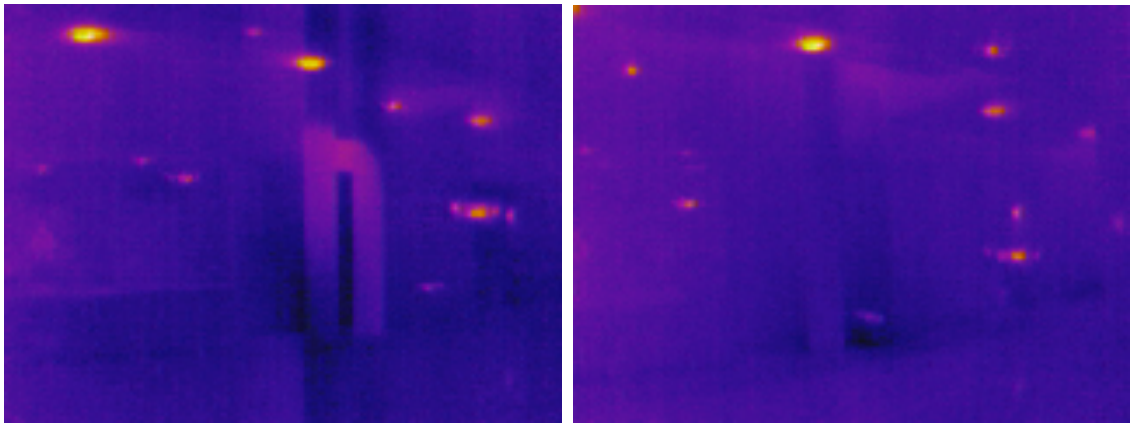
The thermal images captured using the FLIR Lepton camera have an image resolution of $n_{row} \times n_{col} = 160 \times 120$ pixels. Using the setup described in the above section, we have created a data set $D_j = \{y_j^i\}_{i=1}^{N=1000}$ of thermal images where y represents each thermal image in a set of 1000 of images from the class j. Here class j denotes the images that have j UAVs present in them. For example, class 2 denotes thermal images with 2 UAVs. A subset from this data set is provided to the proposed CNN model as input for training.

C.3.3 Convolution neural network (CNN)

The proposed CNN model is shown in Fig. C.2. The input to the CNN is the thermal image of size 160×120. This is followed by convolution (Conv2D) and max pooling layers. After the input layer, the image is fed into a convolutional layer with 8 filters and a kernel size of 3×3. Features are extracted in this layer and the reduced image is fed to a max pooling layer of dimension 2×2. The reduced image then undergoes a second convolutional operation with a L2 regularizer that has a regularization value of 0.001. This is further followed by max pooling, flatten, and dense layers. All the convolutional layers are activated using the ReLu non-linear activation function. Finally, the output layer uses softmax activation to provide outputs in terms of the probability of the number of UAVs present in the captured image. Additional architectural details are outlined in Table C.3.

(a) (b)

Figure C.3: RGB images from Picamera for (a) 6 UAV measurement scenario, (b) 8 UAV measurement scenario.



(a) (b)

Figure C.4: Thermal image from FLIR Lepton for (a) 6 UAV measurement scenario, (b) 8 UAV measurement scenario.

The UAV-thermal dataset that is developed for this work contains $10,000$ thermal images. The images are segregated into folders ranging from 1 to 10, where each folder represents a class depicting the number of UAVs in the scene. The dataset is divided into the train, test, and validation sets in the ratio 80%, 10%, and 10% respectively. The sparse categorical cross-entropy function is used to minimize the loss. Further, the model uses an adaptive momentum (Adam) optimizer with 10^{-4} as the learning rate during training.

C.4 Results

In this paper, we propose a low-cost approach to estimate the UAV count in a scene using lightweight CNN. Fig. C.3 shows the visual RGB images of the measurement scenarios with 6 and 8 UAVs flown in a random fashion. Fig. C.4 depict the

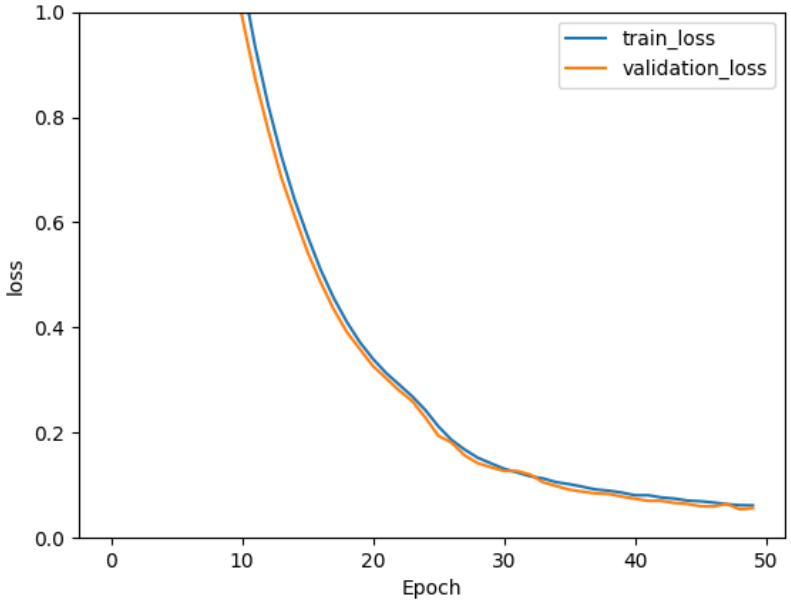


Figure C.5: Training and validation loss curve.

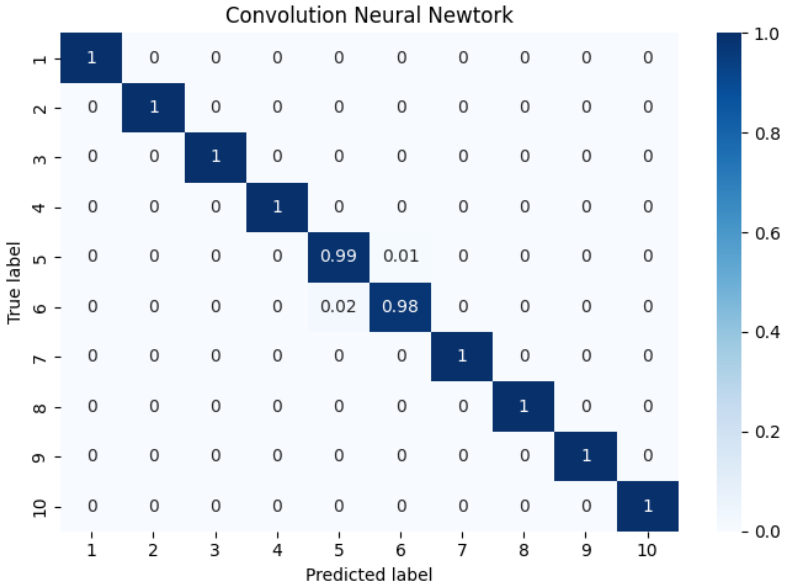


Figure C.6: Confusion matrix obtained on the test data set.

corresponding thermal images for the 6 and 8 UAV scenarios respectively. The proposed CNN model is trained on the thermal images from the created dataset. We use 10-fold cross-validation to train the proposed CNN model. The dataset is split with $(80, 10, 10)\%$ for train, testing, and validation respectively. Fig. C.5 shows the loss curve obtained during the training. It can be observed from Fig. C.5 that the training and validation loss converge to a minimum after 40 epochs. The model attains 99.91% validation accuracy from the 10-fold hold-out validation. Further, after the training, the model is tested with previously unseen data (10% from the dataset) to evaluate the performance. We obtained an average of 99.9% accuracy on the test set. The confusion matrix depicting the model performance is provided

in Fig. C.6.

C.5 Conclusion and Discussion

In this paper, we proposed a method to estimate the number of UAVs in a scene. An end-to-end technique is developed that captures thermal images using embedded hardware and later processes and estimates the number of UAVs. Additionally, a UAV-thermal dataset is also developed that consists of a combination of 10 UAVs flying in all directions in a random manner. Extensive experiments demonstrate that the proposed CNN architecture was able to provide 99.9% accuracy in estimating the exact number of UAVs present. Future extensions to this work include UAV model identification in a multi-UAV setting that can help address UAV security threats.

References

[1] A. N. Wilson, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Embedded sensors, communication technologies, computing platforms and machine learning for uavs: A review. *IEEE Sensors Journal*, 22(3):1807–1826, 2022.

[2] Matthew Ritchie, Francesco Fioranelli, and Hervé Borrion. Micro uav crime prevention: Can we help princess leia? *Crime Prevention in the 21st Century: Insightful Approaches for Crime Prevention Initiatives*, pages 359–376, 2017.

[3] Wilson AN, Ajit Jha, Abhinav Kumar, and Linga Reddy Cenkeramaddi. Estimation of number of unmanned aerial vehicles in a scene utilizing acoustic signatures and machine learning. *The Journal of the Acoustical Society of America*, 154(1):533–546, 2023.

[4] Jie Zhao, Jingshu Zhang, Dongdong Li, and Dong Wang. Vision-based anti-uav detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):25323–25334, 2022.

[5] A. N. Wilson, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Multitarget angle of arrival estimation using rotating mmwave fmcw radar and yolov3. *IEEE Sensors Journal*, 23(3):3173–3182, 2023.

[6] Linga Reddy Cenkeramaddi, Jyoti Bhatia, Ajit Jha, Santosh Kumar Vishkarma, and J. Soumya. A survey on sensors for autonomous systems. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1182–1187, 2020.

[7] A. N. Wilson, Khushi Anil Gupta, Balu Harshavardan Koduru, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Recent advances in thermal imaging and its applications using machine learning: A review. *IEEE Sensors Journal*, 23(4):3395–3407, 2023.

[8] Ye Wang, Yueru Chen, Jongmoo Choi, and C-C Jay Kuo. Towards visible and thermal drone monitoring with convolutional neural networks. *APSIPA Transactions on Signal and Information Processing*, 8:e5, 2019.

[9] Adrian Carrio, Yucong Lin, Srikanth Saripalli, and Pascual Campoy. Obstacle detection system for small uavs using ads-b and thermal imaging. *Journal of intelligent & robotic systems*, 88:583–595, 2017.

[10] Petar Andraši, Tomislav Radišić, Mario Muštra, and Jurica Ivošević. Night-time detection of uavs using thermal infrared camera. *Transportation Research Procedia*, 28:183–190, 2017.

[11] Naoual El-Djouher Mebtouche and Nadia Baha. Robust uav detection based on saliency cues and magnified features on thermal images. *Multimedia Tools and Applications*, 82(13):20039–20058, 2023.

[12] Lars Sommer and Arne Schumann. Deep learning-based drone detection in infrared imagery with limited training data. In *Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies IV*, volume 11542, page 1154204. SPIE, 2020.

[13] TELEDYNE FLIR. TELEDYNE FLIR Lepton. [Online]. Available: https://www.flir.eu/products/lepton/?model=500-0771-01&vertical=microcam&segment=oem.

[14] DJI. DJI Camera Drones. [Online]. Available: https://www.dji.com/no/camera-drones.

[15] Ryze Robotics. Tello EDU. [Online]. Available: https://www.ryzerobotics.com/tello-edu.

[16] SYMA. SYMA X30 Foldable Drone. [Online]. Available: https://www.symatoys.com/goodshow/x30-syma-x30-foldable-drone.html.

[17] Raspberry Pi. Raspberry Pi 4. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/.

[18] Picamera. Picamera Documentation. [Online]. Available: https://picamera.readthedocs.io/en/release-1.13/.

[19] Ubuntu. Ubuntu 18.04. [Online]. Available: https://releases.ubuntu.com/18.04/.

[20] ROS Wiki. ROS Melodic. [Online]. Available: http://wiki.ros.org/melodic.

Appendix D

PAPER D

Title: Estimation of number of unmanned aerial vehicles in a scene utilizing acoustic signatures and machine learning

Authors: **Wilson A. N.**, Ajit Jha, Abhinav Kumar, Linga Reddy Cenkeramaddi

Journal: The Journal of the Acoustical Society of America (J. Acoust. Soc. Am.)

Estimation of number of unmanned aerial vehicles in a scene utilizing acoustic signatures and machine learning

Wilson A. N., Ajit Jha, Abhinav Kumar, Linga Reddy Cenkeramaddi

Abstract: **With the exponential growth in unmanned aerial vehicle (UAV)-based applications, there is a need to ensure safe and secure operations. From a security perspective, detecting and localizing intruder UAVs is still a challenge. It is even more challenging to accurately estimate the number of intruder UAVs on the scene. In this work, we propose a simple acoustic-based technique to detect and estimate the number of UAVs. Our method utilizes acoustic signals generated from the motion of UAV motors and propellers. Acoustic signals are captured by flying an arbitrary number of 10 UAVs in different combinations in an indoor setting. The recorded acoustic signals are trimmed, processed, and arranged to create a UAV audio dataset. The UAV audio dataset is subjected to time-frequency transformations to generate audio spectrogram images. The generated spectrogram images are then fed to a custom lightweight convolutional neural network (CNN) architecture to estimate the number of UAVs in the scene. Following training, the proposed model achieves an average test accuracy of 93.33% as compared to state-of-the-art benchmark models. Furthermore, the deployment feasibility of the proposed model is validated by running inference time calculations on edge computing devices such as the Raspberry Pi 4, NVIDIA Jetson Nano, and NVIDIA Jetson AGX Xavier.**

D.1 Introduction

Advancements in chip miniaturization and wireless connectivity have made unmanned aerial vehicle (UAV) based solutions attractive in various applications such as agriculture [1], disaster management [2], aerospace [3], law enforcement [4], etc. Their widespread popularity can be attributed to their unparalleled maneuverability, decreasing cost, and increased sophistication [5]. However, the advantages of UAVs are also exploited for improper and illegal use [6]. Further, as UAVs are compact and small in size, concerns regarding collisions with other air-borne entities, privacy, security, delivery of dangerous payloads, etc. should also be addressed.

Ground control stations should be well-equipped with systems that can detect and monitor UAV activity based on requirements.

There have been multiple attempts throughout the literature pertaining to the detection and tracking of small-size UAVs. Some of these approaches include the use of WiFi signals, RF radiations, vision-based sensors, radar-based approaches, and acoustic signatures [7], [8]. However, compared to other sensors, utilizing acoustic sensors for UAV detection has been shown to exhibit a number of advantages. Acoustic sensors are low-cost compact devices that detect pressure fluctuations created by sound waves. Unlike traditional vision and radar-based sensors, acoustic sensors are typically omnidirectional in nature. This allows them to sense disturbances in all directions making them an ideal choice for collision-avoidance systems. Further, passive acoustic sensors do not emit any radiation and hence it is environment friendly. As acoustic signals are independent of the UAV form factor, it fairs well in comparison to radar systems that require a threshold radar cross-sectional area to enable detection. The output data rate for acoustic sensors is comparatively low as compared to vision and radar-based systems facilitating seamless data acquisition and processing. However, the limitation of acoustic systems is their low detection range. For small inexpensive acoustic sensors, the detection range is typically less than 300 m facilitating only short-range detection [9]. In this effort, we address the problem of detecting and estimating multiple UAVs in an indoor environment using acoustic sensors and machine learning techniques.

D.2 Related Work

In this section, we provide a brief overview of the various works in the literature pertaining to UAV detection using acoustic signatures. It was observed that small multi-rotor UAVs can produce complex acoustic fields due to UAV motorization and propeller motion. The acoustic field thus produced contains complex harmonics and is arguably a unique characteristic of the UAV. Investigations related to the study of the UAV acoustic field can be found [10], [11], [12] in the literature. In [10] investigated the effect of UAV acoustics on human subjects in an indoor occupational environment. They concluded that the ability to accurately identify and record UAV acoustic signatures primarily depends upon the sound pressure level of the UAV. The study also found that an efficient redesign of multi-rotor UAVs is essential to lower noise levels and regulate the noise frequency spectrum produced by UAVs. A similar study [11] utilizes a large-aperture scanning microphone array to measure the sound pressure level of a hovering UAV. The obtained results are then used to determine a better design of UAV acoustics and analyze the UAV acoustic field. In [12], the authors utilize the experimental data obtained from [11] to model the sound pressure level of a UAV acoustic field using a physics-infused machine learning algorithm. The physics-infused model is developed utilizing the interference from sound pressure waves that are produced from acoustic monopole sources. The above-mentioned works indicate that UAV acoustic signatures are unique and depend on a number

of factors. Our work utilizes these unique acoustic harmonic signatures generated by multiple UAVs to detect and estimate their presence in an indoor setting. In the following subsections we provide a brief account of the various techniques used for UAV detection based on acoustic signatures.

D.2.1 Conventional approach

The conventional approach relies on correlation and other signal-processing techniques to detect UAVs from acoustic signatures. One early approach by [13], uses a stationary microphone array to detect and track the flight trajectory of low-altitude UAVs using their engine sound. UAV detection is performed by exploiting the Doppler shift of the engine sound. On the other hand, tracking is carried out by estimating the direction of arrival of the UAV. The direction of arrival is obtained by utilizing the acoustic phase shift over the microphone array. The real-time performance of the proposed approach was evaluated by performing field experiments. It was found that the propagation delay of the acoustic delay impacts the UAV detection and tracking performance. Another approach in [14], utilized a modified cross-correlation technique for UAV detection. The proposed approach leverages the differential Doppler shift that is created due to the high-speed UAV motion and microphone array separation. Using the differential Doppler shift property, the received signals are successfully decorrelated from the ambient noise to enable UAV detection. Measurements were carried out in a controlled area with little noise. Ambient noise was later added to the obtained data to evaluate the performance. Another work [9] for UAV detection and classification involved the development of a drone acoustic detection system (DADS) using microphone nodes. In the DADS, the detection of UAVs is performed using the steered-response phase transform (SRP-PHAT) method while classification is obtained by utilizing the propeller frequencies from the spectrogram of the measured acoustic signatures. It is reported that the SRP-PHAT method provides reliable performance with real UAVs in real-world scenarios. However, the classification algorithm requires the UAVs to obtain a threshold distance to achieve better performance. The work by [15], utilizes the Barlett, Capon, and cross-correlation method to study and analyze the acoustic spectrum generated from UAVs. The proposed approach utilizes additional high-pass filters to obtain performance similar to the mel-frequency cepstral coefficients (MFCC) method for UAV detection. Further experiments also reveal that the cross-correlation method exhibited superior performance when followed by low-pass filtering to remove noise. In comparison, [16], utilizes the intrinsic harmonics inherent in the acoustic field signatures for detection and 3D localization. The acoustic signal's fundamental frequency and a few relevant harmonics are extracted using a pitch detection algorithm coupled with zero-phase bandpass filters. Experimental measurements have been carried out in anechoic and outdoor environments for performance evaluation. It was observed that the proposed approach fairs well when the UAV traverses in simple vertical trajectories. However, for complex trajectories with multiple UAVs, the performance was low and required additional research. The study by [17] also

provides insights into utilizing acoustic information to detect UAVs. In this work, the authors measure the noise level using a sound level meter in a controlled environment containing only one UAV. After obtaining the noise-free measurements, anthropogenic noise of people and background music was generated in the environment along with the UAV. Spectral methods were used to analyze the frequency spectrum and it was reported that UAV detection was confirmed upon observing a 5000 Hz frequency in the spectrum. The research however fails to reproduce and verify the results in a real-world environment with multiple UAVs. In [18], the authors have addressed the problem of UAV detection through a biologically inspired vision approach. The spectrogram signals obtained after performing time-frequency analysis on the audio signals provide meaningful information that is embedded in noise. By preprocessing these spectrogram images through a hoverfly vision model, useful representations of these audio signals can be retrieved. The extracted representations can be utilized for UAV detection. It was shown through outdoor field trials that the bioinspired technique can improve the maximum UAV detectable distance between 30% and 50% with respect to traditional narrowband and broadband techniques. However, the proposed approach requires additional verification by using more UAV experiments and flight scenarios.

D.2.2 Machine learning approach

The advent of machine learning has bought new capabilities for UAV detection and classification [19], [20]. Machine learning techniques identify the inherent hidden patterns from the data that aid in UAV detection and classification [21]. By utilizing these techniques with additional preprocessing techniques such as short-time Fourier transform (STFT), principal component analysis (PCA), etc., a significant improvement in detection and classification accuracy has been reported [22]. In the subsequent sections, we explore some of the existing literature that uses machine learning techniques for UAV detection and classification in terms of unsupervised and supervised learning algorithms.

D.2.2.1 Unsupervised learning

In unsupervised learning [23], the algorithm learns to extract the hidden patterns from the data. These algorithms work primarily on unlabeled data and learn the inherent structure of the data without the need for any human intervention. They are mainly used for tasks such as clustering, association, and dimensionality reduction. In [24], the authors study the acoustic fields generated by various small quadcopter UAVs. The data thus obtained is used along with simulation software such as COMSOL Multiphysics to perform numerical simulations and analysis. The study determined the influence of blade defects, directional patterns, and pressure variations caused by UAV propellers on UAV acoustic fields. The collected acoustic signatures were further provided to a neural network that is trained on the cepstrum coefficients to obtain UAV detection. In [25] the authors used multiple microphone nodes to detect and track a UAV in a real-world environment with background noise.

The work used MFCC and STFT for preprocessing the data. This was followed by using support vector machines (SVM) and convolutional neural networks (CNN) for training. Reported results indicate that the STFT-SVM model exhibited better performance to detect a single UAV when the UAV approaches the vicinity of a microphone node. Future work can include experimenting with multiple UAVs and also reducing background noise during preprocessing. The work in [26] utilizes a multi-class SVM for identifying UAVs in diverse environmental conditions. A dataset is created that contains five 70 minute audio from nature during the daytime, street traffic, train, crowd, and flying UAVs. The audio files are trimmed to 5 second and 20 millisecond segments for analysis. Preprocessing is then performed to extract temporal centroid, spectral roll-off, spectral centroid, zero crossing rate (ZCR), MFCCs, etc., as features. The extracted features are then fed to the SVM classifier to obtain a high UAV detection accuracy of 96.4%. In [27], the authors perform UAV detection using two classifiers, plotted image learning (PIL) and k nearest neighbors (KNN). Sound clips with a 1 second duration of DJI Phantom 1 and 2 are recorded separately both indoors and outdoors in a noise-free environment. Later, outdoor environment sounds are added to simulate real-world scenarios. The FFT is applied to the sound clips which are then fed to the different classifiers. The authors reported that PIL showed 83% accuracy in UAV detection as compared to KNN which accounted for 61%. In [28], the authors developed a distributed system using acoustic wireless sensor network for UAV detection and localization. Through trial experiments, it was observed that the power spectral density (PSD) of UAV sound differed significantly from the background spectrum. On the basis of this concept, an acoustic dataset was created. The dataset consisted of UAV sounds that were augmented with background environment sounds. The sound clips are low pass filtered at 15 kHz, after which the PSD is obtained using FFT. PCA is further performed for dimension reduction. The preprocessed signals are then divided for training, testing, and additional testing with overlapped signals and subsequently fed to the SVM classifier. It was reported that UAV detection was successful when the introduced signal-to-interference ratio (SIR) was greater than 10 dB. In this work [29], the authors use the blind source separation (BSS) method to detect UAVs in the presence of multiple source interference. Three different UAVs are used separately to capture the audio signatures. The proposed method works by first estimating the number of sources. After source estimation, three methods ICA, PCA, or variational mode decomposition (VMD) are applied based on the type of source separation (overdetermined, positive-definite, or underdetermined) required. The features extracted are then fed to different machine learning algorithms such as SVM, KNN, and decision trees to evaluate the performance. It was reported that SVM and KNN showed similar performance with SVM exhibiting slightly better performance. Both algorithms exhibited an accuracy of more than 90% for UAV detection outperforming traditional filtering and mixed-signal methods. Another approach for acoustic-based UAV detection was performed by [22]. In this work, the sounds of amateur UAVs, birds, airplanes, and thunderstorms are recorded in a noisy environment. The authors use MFCC and linear predictive cepstral coefficients

(LPCC) for feature extraction. The extracted features are then fed to SVMs with linear, cubic, and quadratic kernels to detect and identify UAV acoustics. Results show that SVM with the cubic kernel when coupled with MFCC features outperformed LPCC with a UAV detection accuracy of around 96.7%.

D.2.2.2 Supervised learning

In contrast to unsupervised learning, supervised learning [23] utilizes labeled datasets as inputs and outputs. The labeled data serves as a kind of supervision to help the model learn the structure of the data. Supervised algorithms can learn over time and improve its accuracy based on the amount of labeled data and its inherent structure. In [30], the authors perform a comparative study to determine the best classifier for acoustic UAV detection. Acoustic signatures from various UAVs are recorded individually and augmented with diverse environmental noise to simulate real-world UAV scenarios. The MFCCs from these signals are extracted and fed to different classifiers and their performance is evaluated. It was reported that RNN provided the best performance with an F-score of 80%, followed by the Gaussian mixture model (GMM) with 68% and CNN at 58%. The study in [31] uses normalized STFT on UAV acoustic signals. The UAV acoustic signatures are recorded using DJI Phantom 3 or 4 models. The recorded sound clips are trimmed to a length of 20 ms with 50% overlapping. Normalized STFT is performed on these sound clips to obtain 41958 non-UAV and 68931 UAV sound frames. The non-UAV sound frames contained acoustic signatures from scooters and motorcycles. The output obtained after performing the STFT is then fed to the proposed CNN architecture after adding additive white Gaussian noise (AWGN). Results reported a 98.97% UAV detection accuracy and 1.28 false alarm rate (FAR). In this work [32], the sensory substitution of pre-existing and ambient microphones along with CNN is used to detect remotely piloted aircraft systems (RPAS) in urban environments. Indoor and outdoor experiments were carried out individually with a diverse set of RPASs. Spectrogram images are generated from the recorded audio clips. The spectrogram images are then further used to train the Inception CNN model via transfer learning. Results showed an RPAS detection accuracy of greater than 90% for all RPAS classes. The work in [33] uses mel-spectrograms to extract the features from the audio signals of UAVs. The extracted features are then used with CNNs and CRNNs for UAV classification. It was concluded that CNNs exhibited superior performance in the classification of UAVs from the obtained mel-spectrograms. Further, the study also investigated the use of late fusion methods with ensemble techniques to improve UAV detection performance. Another work by [34] also utilizes the audio spectrograms along with CNN, recurrent neural network (RNN), and convolutional recurrent neural network (CRNN) to identify and detect UAVs. The authors conducted two experiments using two different UAVs in a controlled environment. Real-world background noises were augmented to obtain realistic audio information that can be used for inference. Reported results indicate that CNN and CRNN showed better performance over RNN in accurately detecting and

Table D.1: Summary of the latest works on machine learning-based acoustic detection of UAVs

Reference	Method	Results	Limitations
[30]	MFCC coefficients are fed to RNN, GMM, and CNN.	RNN showed best F-score with 80% followed by GMM with 68% and CNN with 58%.	• Augmented environmental noise. • Multi-UAV detection is absent.
[26]	Preprocessing using ZCR, MFCC, spectral centroid, etc. Extracted features are fed to multi-class SVM.	UAV detection accuracy - 96%	• Single UAV case. • Lacks real-world experiments and background noise.
[27]	Preprocesing using FFT. Trained using PIL and KNN.	PIL - 83% and KNN - 61% accuracy	• Sound clips from 2 UAVs recorded separately.
[28]	PSD using FFT followed by PCA for dimension reduction. Output fed to SVM.	Best accuracy when SIR was greater than 10 dB.	• SVM is more sensitive to bit error rate.
[31]	Normalized STFT features with CNN.	UAV detection accuracy of 98.97%	• Considered only single UAV scenario. • AWGN is added to simulate a noisy environment.
[25]	Preprocessing with MFCC and STFT. Obtained features fed to SVM and CNN.	eSTFT-SVM reported best performance.	• Considered only single UAV case. • Model accuracy is low.
[34]	Audio spectrograms with CNN, RNN, CRNN.	CNN reported best detection accuracy with 96.38% followed by CRNN with 94.72%. Experimented with two different types of UAVs.	• Lacks real-world experiment scenarios. • Doesn't estimate the number of UAVs.
[32]	Audio spectrograms with CNN. Used different RPAS classes individually for measurements.	Greater than 90% detection accuracy.	• Multiple RPAS scenario is absent.
[22]	MFCC and LPCC for feature extraction. Features are fed to SVM with linear, cubic, and quadratic kernels.	MFCC with SVM cubic kernel achieves 96.7% detection accuracy.	• Considers only single UAV scenario.
[24]	Trains neural network on cepstrum coefficients.	Relatively high UAV detection rate.	• Multiple UAV scenarios are absent.
[33]	Mel-spectrograms for feature extraction followed by CNN and CRNNs.	CNN (94.7% accuracy) outperformed CRNN (94.1% accuracy). Experimented with real-world scenarios.	• Multiple UAV scenarios are absent.
[29]	BSS using ICA, PCA, or VMD features. Obtained features are fed to SVM, KNN, and Decision trees	SVM and KNN reported more than 90% accuracy	• Lacks real-world scenarios with background noise.

classifying UAVs. Our work also revolves around a similar approach in which we utilize audio spectrograms to perform multiple UAV detection.

Table D.1 summarizes the latest works in the literature related to machine

learning-based acoustic detection of UAVs. As seen from Table D.1, the majority of the literature focuses on detection for a single UAV scenario [25], [26], [30], [31], [32], [22], [33]. The results obtained for detecting a single UAV can widely vary in a multiple UAV scenario. Similarly, the scenarios considered in the literature more or less replicate controlled and well-defined UAV trajectories [26], [34], [30], [31], [29]. Such scenarios may not completely provide a realistic UAV flight trajectory and may affect detection accuracy. Furthermore, some of the techniques described require the use of high-end sophisticated computing infrastructure which may not be always feasible and available [28]. Our work differs from the previous works in detecting multiple UAVs rather than a single UAV. To the best of our knowledge, this is the first time acoustic signatures have been employed to estimate the detection of maximum 10 UAVs in a scene. The scenarios considered comprise multiple UAVs maneuvering in random directions and speeds. Our work also uses the inherent background noise while performing detection. We have included one outdoor measurement that includes background noise from wind and birds chirping. We use supervised learning techniques in this work due to their superior performance in the detection and classification of targets. Although unsupervised techniques have the advantage of extracting the inherent features from unlabeled data, it fails in performance when the requirement calls for the ability to identify specific classes of targets. Further, our custom CNN architecture outperforms the current state-of-the-art machine learning models in terms of accuracy and model size. Owing to the relatively less model size, the custom CNN architecture consumes fewer resources thereby enabling it to be deployed on lightweight edge-computing devices such as Raspberry Pi 4, NVIDIA Jetson Nano, etc. We test the model on these devices and also provide inference time for the same. As such the major contributions of this paper are as follows:

- An UAV acoustic-based dataset is created by utilizing a total of 10 UAVs. An arbitrary number of UAVs are flown randomly within the measurement area and the acoustic field signatures are captured using a cardioid unidirectional microphone.
- Time-frequency algorithms such as continuous wavelet transform (CWT) are applied to transform the recorded acoustic field signatures to spectrogram images.
- A custom lightweight CNN architecture is designed to estimate the number of UAVs in the scene. The performance of the proposed model is compared with state-of-the-art benchmark machine learning models in terms of accuracy and model size.

The remaining sections of this paper are organized as follows: Section D.3 provides the methodology of the proposed approach. Section D.4 describes the measurement setup and details regarding dataset creation. Section D.5 focuses on the data preprocessing and the machine learning algorithm that is used. Section D.6

summarizes the standard benchmark machine learning models that are used to compare the performance of the proposed model. Section D.7 provides an overview of the edge computing devices on which the proposed model is executed. Section D.8 summarizes the results obtained with the proposed approach. Section D.9 discusses the implications of the proposed approach along with limitations and future work. Finally, the paper is concluded in Section D.10.

D.3 Methodology

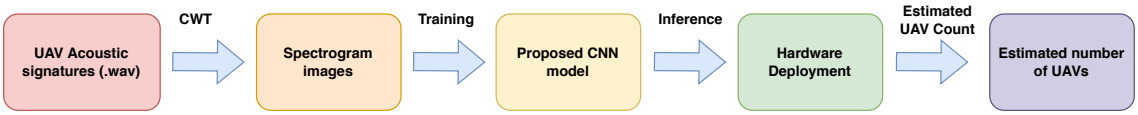


Figure D.1: Block diagram of the proposed work.

The proposed work utilizes the acoustic field generated from UAV rotors and propellers. The high-speed motion of rotors and propellers produces pressure differences leading to the generation of an acoustic field. In the proposed work, these acoustic field signatures are captured with the help of a cardioid microphone. Acoustic signatures from 10 UAV combinations are captured and processed to generate audio spectrogram images. These images are then used to train CNN models to estimate the number of UAVs in the scene. A simplified flow diagram of the methodology is shown in Fig. D.1.

D.4 Measurement setup and dataset

D.4.1 Measurement setup

The primary measurement area is an indoor lab environment that covers a semi-circular area of 5 meters in radius. Experiments were designed to capture acoustic signatures from a total of 10 UAV combinations that are flown in a random fashion within the prescribed area. The UAV models employed for the experiment include the DJI Mavic 2 Enterprise [35], DJI Mini 2 [36], DJI Mini SE [37], DJI Mini 3 Pro [38], DJI Tello EDU [39], and SYMA X30 [40]. Except for the DJI Mavic 2 Enterprise, all the other UAVs fall into the 250 grams category and have relatively smaller dimensions (approximately $251 \times 362 \times 70$ mm [38]). The small size of the UAVs makes them an excellent choice for testing the estimation performance in a multi-UAV scenario. The DJI Tello UAVs were operated programmatically to follow a prescribed trajectory. The remaining UAVs were operated manually to fly in a random fashion to simulate a near real-time scenario. Further details regarding the experiment are provided in Table D.2.

The UAV acoustic signatures are recorded using the Shure MV7 microphone [41]. The microphone has a unidirectional (cardioid) type polar pattern with an output impedance of 314 ohms. It has a frequency response ranging from 50 Hz to 16000

Table D.2: Measurement setup and experiment details

Experiment Parameters	**Details**
Semicircular area	Radius: 5 meters
Measurement duration	5 minutes
Microphone	**Shure MV7**
Frequency response	50 Hz - 16000 Hz
Output impedance	314 ohms
Sampling rate	48000 Hz
Total UAVs	**10**
DJI Mavic 2 Enterprise	1
DJI Mini 2	1
DJI Mini SE	2
DJI Mini 3 Pro	1
DJI Telllo EDU	4
SYMA X30	1

Hz with an adjustable gain spanning between 0 and +36 dB [42]. Additional details regarding the microphone are provided in Table D.2. The microphone is mounted on a tripod stand and faced toward the measurement area. As the microphone has a unidirectional cardioid polar pattern, the acoustic disturbances originating from UAVs flying in front of the microphone are captured and amplified. However, the disturbances that originate from the rear end of the microphone are attenuated and hence do not contribute to the output signal. Fig. D.2 shows the microphone setup that was used to capture the acoustic signatures from UAVs.

Each measurement of the experiment consisted of flying an arbitrary number of UAVs in the prescribed area for a duration of 5 minutes. For example, the fifth measurement captured acoustic field signatures from 5 randomly flown UAVs. The sixth measurement involved flying 6 UAVs in a random manner. To improve variability, each measurement of the experiment has been designed to use different types of UAVs as much as possible. However, due to availability constraints we resorted to similar UAV models for measurements that required more number of UAVs. Fig. D.3 depicts a 5 UAV measurement case. To provide additional variability in the acoustic field measurements, we performed the 2nd measurement outdoors. The outdoor measurement area is roughly the same semicircular area of radius 5 meters with additional noise related to wind, birds chirping, etc. Further, the first measurement has been taken independently using 3 different UAV models to increase the number of samples.

D.4.2 Dataset details

The acoustic signatures that are recorded and captured from the experiment are used to create a dataset comprising UAV audio clips. Each recorded acoustic signature is of 5 minute duration. The recorded signatures are carefully trimmed to retain only the portion pertaining to UAV audio. Each trimmed audio signal is of 4 minutes and 45 seconds (285 seconds) with a sample rate of 48000 Hz. To reduce latency and

Figure D.2: Shure MV7 setup used for measuring the UAV acoustic signals

Figure D.3: Capturing acoustic signals for a 5 UAV scenario.

ensure smooth processing, we divide each trimmed signal (285 second duration) into 95 equal parts. Each one of the 95 parts is a 3 second audio clip with a sampling

rate of 48000 Hz. In total, the dataset contains 1140 UAV audio clips of 3 second duration.

D.5 Preprocessing and algorithm details

In the preprocessing stage of the proposed approach, various signal processing transformations are applied on the prepared UAV audio dataset. After preprocessing, the resulting spectrogram images are then fed to lightweight CNN models to estimate the number of UAVs present in the scene.

D.5.1 Continuous Wavelet Transform (CWT)

The CWT is a wavelet transform that decomposes a signal into its time and frequency components [43]. Just like the STFT [44], the CWT measures the correlation between the original signal $f(t)$ and the analyzing wavelet ψ. Depending upon the correlation with the original signal, the analyzing wavelet is scaled and dilated by parameters p and q respectively. Assuming the scaling parameter $p > 0$, and dilation parameter q, then the CWT for a signal $f(t)$ is computed as,

$$C(p, q; f(t), \psi(t)) = \int_{-\infty}^{\infty} f(t) \frac{1}{p} \psi^* \left(\frac{t - q}{p} \right) dt, \tag{D.1}$$

where the * represents the complex conjugate [45]. If the CWT is applied to a real signal, then the obtained output is also real-valued. By varying the parameters p and q continuously, we obtain the $C(p, q)$ coefficients which are subsequently used to plot the spectrogram of the signal. In the proposed method, CWT is applied over the trimmed audio clips. As each audio clip is 3 seconds long with a sample rate of 48000 Hz, the resulting spectrogram exhibits time and frequency components corresponding to these parameters. The acoustic signatures along with their corresponding spectrogram outputs are plotted in Fig. D.4 and D.5.

D.5.2 Convolutional Neural Network (CNN)

CNNs are unique deep-learning architectures that utilize artificial neural networks to detect and classify objects from images. In the proposed approach, we develop a custom CNN architecture to extract feature information from spectrogram images.

While designing the custom CNN architecture, we first checked the performance by varying the number of layers as 5, 10, 15, and 20 layers. We used 10-fold cross-validation with a data set split of (80, 10, 10) for training, testing, and validation. It was observed that CNNs with 20 layers or more provided better performance as compared to the ones with a lower number of layers. Subsequently, we varied the number of layers along with the image resolution to obtain the best-performing architectures. Table D.3 provides the performance of the CNN architectures with 18 layers and more. The change in performance is also noted with respect to the change in image resolution. It can be observed from Table D.3 that the performance

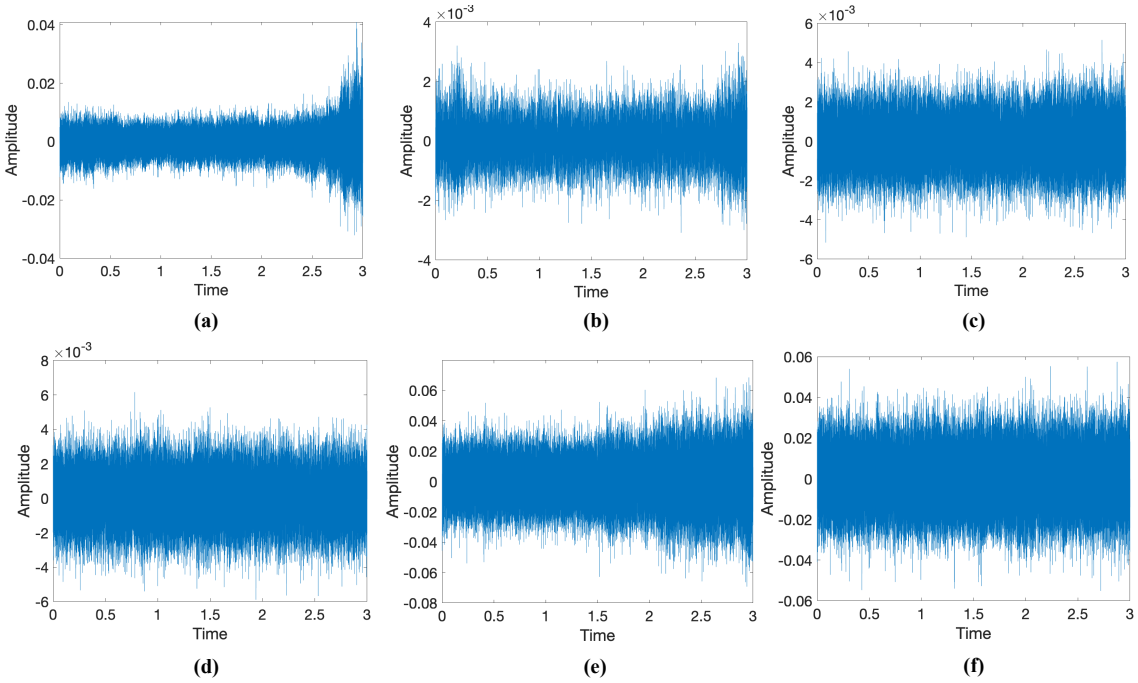


Figure D.4: UAV acoustic signatures for scenarios with (a) 1 UAV (b) 2 UAVs (c) 4 UAVs (d) 6 UAVs (e) 8 UAVs (f) 10 UAVs.

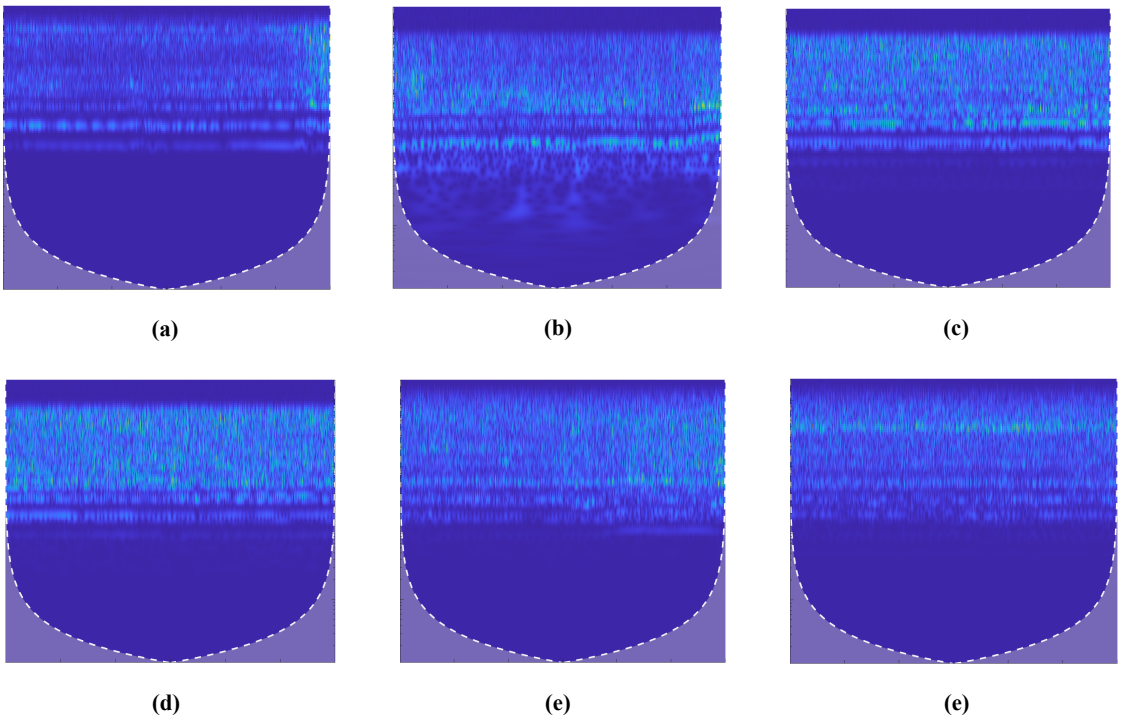


Figure D.5: Spectrogram images of scenarios with (a) 1 UAV (b) 2 UAVs (c) 4 UAVs (d) 6 UAVs (e) 8 UAVs (f) 10 UAVs.

gradually increases from 18 layers and peaks around 22 and 24 layers and then gradually decreases. Out of the five CNN architectures (shown in bold in Table D.3) that exhibited greater than 94% average test accuracy, we chose the CNN

Table D.3: Comparison of CNN architectures based on number of layers and image resolution

Sl. No.	**No of Layers**	**Image resolution**	**Avg. Test Accuracy (%)**
1	18	$300 \times 200 \times 3$	92.29
2	18	$400 \times 300 \times 3$	92.40
3	18	$500 \times 400 \times 3$	86.05
4	18	$600 \times 500 \times 3$	**94.73**
5	18	$700 \times 600 \times 3$	93.55
6	20	$300 \times 200 \times 3$	89.37
7	20	$400 \times 300 \times 3$	88.50
8	20	$500 \times 400 \times 3$	89.56
9	20	$600 \times 500 \times 3$	88.89
10	20	$700 \times 600 \times 3$	88.00
11	22	$300 \times 200 \times 3$	92.19
12	22	$400 \times 300 \times 3$	**94.24**
13	22	$500 \times 400 \times 3$	**94.74**
14	22	$600 \times 500 \times 3$	93.86
15	22	$700 \times 600 \times 3$	**94.05**
16	24	$300 \times 200 \times 3$	92.98
17	24	$400 \times 300 \times 3$	92.98
18	24	$500 \times 400 \times 3$	93.56
19	24	$600 \times 500 \times 3$	93.86
20	24	$700 \times 600 \times 3$	**94.24**
21	26	$300 \times 200 \times 3$	91.13
22	26	$400 \times 300 \times 3$	91.61
23	26	$500 \times 400 \times 3$	89.76
24	26	$600 \times 500 \times 3$	86.34
25	26	$700 \times 600 \times 3$	88.11

architecture (22 layers, $500 \times 400 \times 3$) for further analysis after considering other performance metrics and parameters.

The proposed CNN architecture that is made of 22 layers is shown in Fig. D.6 and Table D.4. The spectrogram image obtained after performing CWT on the audio clips have a resolution of $836 \times 716 \times 3$ pixels. To be trained by the custom CNN architecture, these spectrogram images are resized to $500 \times 400 \times 3$ pixels. The resized images are then fed to the proposed CNN architecture through the input layer. The proposed architecture consists of convolutional layers with kernel dimensions 4×4. We primarily use 8 or 16 convolutional kernels to extract the feature embeddings. The final convolutional layer however additionally uses dilation by a factor of 2. The process of dilation intentionally expands the kernel size by introducing holes between adjacent elements as shown in Fig. D.7. This provides a larger field of view that in turn helps in capturing intrinsical sequence information [46].

To expand the network with additional layers without compromising on performance, our model employs residual blocks. Our model uses a total of 2 residual layers. The architecture of the residual blocks used in the proposed architecture

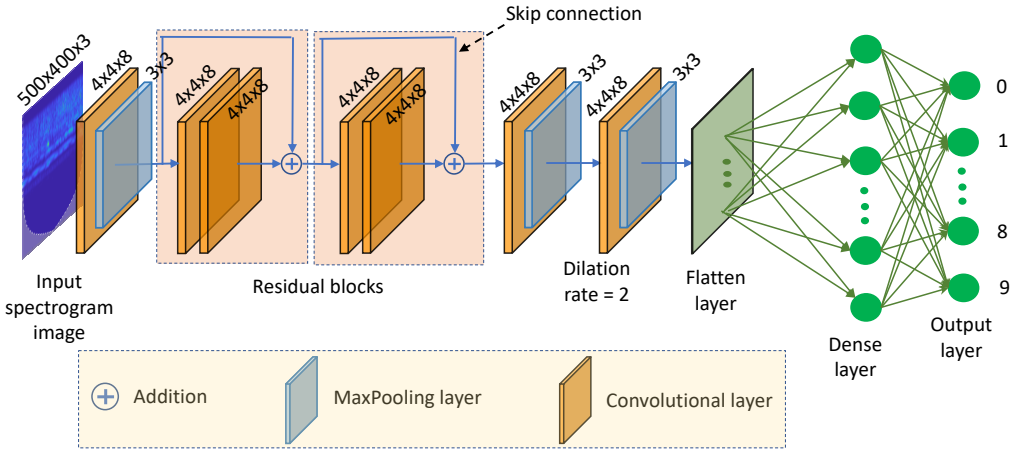
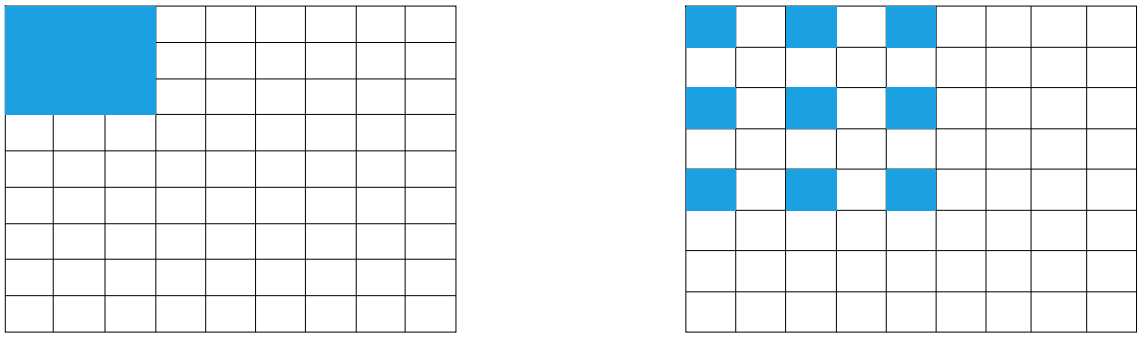


Figure D.6: Proposed CNN architecture.

Table D.4: Layerwise architecture details of the proposed CNN model

No.	Layer	Output Size	Parameter
1	Input	[(None, 500, 400, 3)]	0
2	Batch Normalization (1)	[(None, 500, 400, 3)]	12
3	Conv2D (1)	[(None, 500, 400, 8)]	392
4	MaxPooling2D (1)	[(None, 166, 133, 8)]	0
5	Conv2D (2)	[(None, 166, 133, 8)]	1032
6	Batch Normalization (2)	[(None, 166, 133, 8)]	32
7	Conv2D (3)	[(None, 166, 133, 8)]	1032
8	Batch Normalization (3)	[(None, 166, 133, 8)]	32
9	Add (1)	[(None, 166, 133, 8)]	0
10	Activation (1)	[(None, 166, 133, 8)]	0
11	Conv2D (4)	[(None, 166, 133, 16)]	1032
12	Batch Normalization (4)	[(None, 166, 133, 8)]	32
13	Conv2D (5)	[(None, 166, 133, 16)]	1032
14	Batch Normalization (5)	[(None, 166, 133, 8)]	32
15	Add (2)	[(None, 166, 133, 8)]	0
16	Activation (2)	[(None, 166, 133, 8)]	0
17	Conv2D (6)	[(None, 166, 133, 16)]	2064
18	MaxPooling2D (2)	[(None, 55, 44, 16)]	0
19	Conv2D (7), Dilation rate = 2	[(None, 55, 44, 64)]	16448
20	MaxPooling2D (3)	[(None, 18, 14, 64)]	0
21	Flatten	[(None, 16128)]	0
22	Dense	[(None, 10)]	161290

is depicted in Fig. D.6. Additionally, 3×3 max pooling is used throughout the architecture. Max pooling downsamples the input feature representation [47]. It essentially removes translational invariances from the input representation thereby improving computational efficiency for further layers. The proposed model also utilizes batch normalization at the input layer and residual layers as observed from Table D.4. Batch normalization resolves the problem of internal covariate shift [48] by standardization of the input distribution that involves re-centering and re-scaling. The final layers comprise the flatten layer and the dense layer. The flatten



(a) Dilation factor = 1

(b) Dilation factor = 2

Figure D.7: Effect of dilated convolution for a 3x3 kernel on a 9x9 feature map.

layer transforms the input vector to a 1-dimensional output which is subsequently fed into the dense layer. The dense layer outputs class probabilities which are finally used for detection and classification tasks. In the proposed architecture, since we are estimating a maximum number of 10 UAVs, there are 10 outputs from the dense layer.

The proposed CNN model is trained using the Adam optimizer [49] with the tanh activation function. We have adjusted the learning rate to 0.001 and batch size to 16 to reduce fluctuations in the accuracy/loss curve during training. Training is performed using the Keras deep learning library [50] on two Tesla V100-SXM3 GPU with 32 GB RAM [51].

D.6 Benchmark Models

Benchmark models are state-of-the-art models that have distinct architectural features. For example, DenseNets [52] are special CNNs where the feature maps from each layer are fed to all the subsequent layers thereby preserving the feed-forward nature of the network. ResNet architecture [53] introduces residual blocks to improve performance. The residual blocks are made up of skip connections that retain the abstractions lost in the standard path. The efficientNet family of CNNs utilizes uniform scaling of the depth, width, and resolution of the network to achieve better accuracy. MobileNet [54] models on the other hand are optimized to provide faster operations on mobile and embedded devices. These models have a low memory footprint and offer a better tradeoff between resource utilization and accuracy. We assess the performance of the proposed model by comparing it with these existing benchmark models. We compare it with 23 benchmark models that include models from DenseNet [52], EfficientNet [55], Inception [56], MobileNet [54], ResNet [53], NASNetMobile [57], VGG [58], and Xception [59]. The benchmark models are pre-trained on the ImageNet dataset [60]. The input image resolution fed to the benchmark models has a resolution of $224 \times 224 \times 3$. We compare the proposed model with benchmark models in terms of total parameters, model size, average test accuracy, and number of floating point operations per second (FLOPs). The

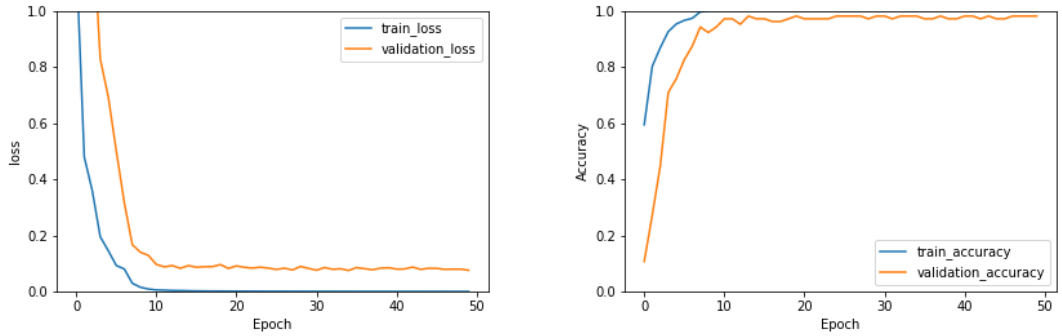
benchmark models are also further deployed on edge computing devices to measure inference time.

D.7 Hardware Deployment

To obtain real-time performance with edge computing devices, we deploy the proposed model on three embedded devices, namely, the Raspberry Pi 4 Model B, the NVIDIA Jetson AGX Xavier, and the NVIDIA Jeton Nano. The Raspberry Pi 4 Model B board contains a quad-core ARM Cortex-A72 processor with 1/2/4/8 GB of RAM. The board is well equipped with various communication interfaces such as Bluetooth 5.0, BLE, and 2.4/5.0 GHz wireless LAN for wireless information transfer. Additionally, the board also provides two USB 3.0, two USB 2.0, and a Gigabit Ethernet port to ensure seamless interfacing with other devices [61]. As compared to Raspberry Pi 4, the NVIDIA Jetson Nano comes with a 128-core Maxwell GPU architecture and quad-core ARM Cortex A5 CPU. With 4 GB RAM and support for multiple interfaces such as USB 2.0 Micro-B, USB 3.0, Gigabit Ethernet, I2C, I2S, SPI, and UART, the Jetson Nano serves to be an excellent choice for high computing edge computing devices [62]. For edge applications that require even more computing capability, the NVIDIA Jetson AGX Xavier is preferred. The Jetson AGX Xavier houses a 512-core Volta GPU architecture and an 8-core Carmel ARM CPU along with 32 GB RAM. It has dedicated deep learning and vision accelerators for various machine learning and computer vision tasks. To interface with other peripheral devices, the Jetson AGX Xavier provides standards such as USB-C, USB 2.0, UART, and RJ45 [63]. To obtain the inference time on the various edge computing devices, the proposed model and benchmark models are first converted to their equivalent Tensorflow Lite versions. TensorFlow Lite [64] is an open-source software developed by Tensorflow to deploy pre-trained models on edge computing devices. By converting the model to its equivalent TensorFlow Lite format, the model is optimized for inference time and model size allowing seamless deployment on various embedded devices. After converting to TensorFlow Lite versions, the proposed model and benchmark models are deployed on these embedded devices to obtain the inference time. The time taken to predict the exact number of UAVs from the spectrogram images is collected and the average inference time is calculated.

D.8 Results

The spectrogram images obtained after applying the CWT transform are used to train the proposed CNN model. We utilize 80% of the dataset for training, 10% for validation, and the remaining 10% for testing. We used 10-fold cross-validation [65] where each fold is trained for 50 epochs with a batch size of 16 and a learning rate of 0.001. The 10-fold cross-validation utilizes 80% of the dataset for training and 10% for validation. The testing is performed on the remaining 10% of the dataset to obtain detection accuracy. Fig. D.8a and D.8b show the loss and accuracy curves



(a) Loss curve (b) Accuracy curve

Figure D.8: Loss and accuracy curves.

obtained during training. It can be observed from Fig. D.8a that the training and validation loss decrease significantly after 10 epochs. Correspondingly, the accuracy curves for training and validation converge close to 1 after 10 epochs indicating that the proposed model requires less training time.

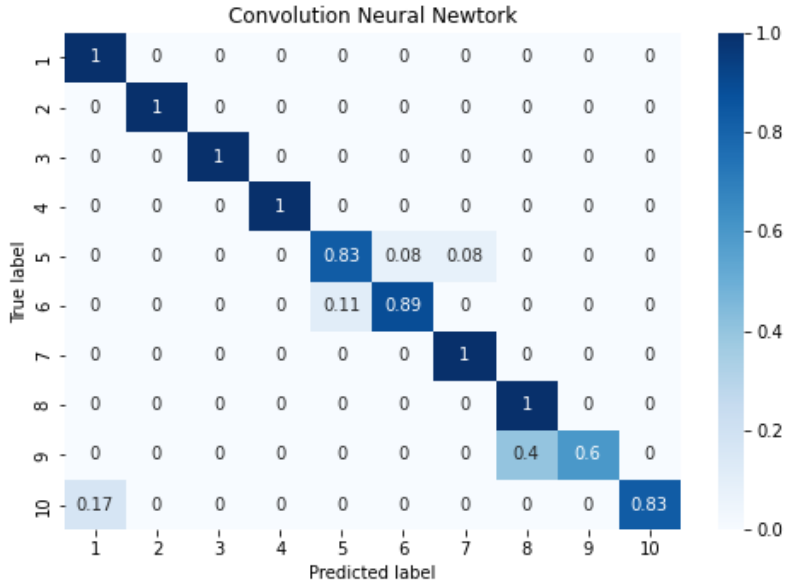


Figure D.9: Confusion matrix for the proposed model.

Upon training, the model performance was evaluated on the test set. Fig. D.9 depicts the confusion matrix obtained after evaluating the proposed model on the test set. It can be observed that the proposed model is able to correctly predict the number of UAVs for more than 90% of the cases. In the remaining cases, the model incorrectly estimates the number of UAVs present in the scene. This might be due to the superposition of acoustic signatures from similar UAV models that can render the obtained signal unresolvable. The performance of the proposed model is also compared with standard benchmark machine learning models as shown in Table D.5. We used the same data set split while calculating the performance metrics on the benchmark models. The spectrogram images are resized to $224 \times 224 \times 3$ pixels before

Table D.5: Performance metrics comparison between proposed CNN model and benchmark models

Sl. No.	Model	Total Parameters	Avg. Test Accuracy (%)	Model Size (MB)	Floating Point Operations (GFLOPs)
1	DenseNet121 [52]	7,047,754	83.77	28.98	2.88
2	DenseNet169 [52]	12,659,530	87.28	51.76	3.42
3	DenseNet201 [52]	18,341,194	88.68	74.68	4.37
4	EfficientNetB0 [55]	4,062,381	83.85	16.74	0.4
5	EfficientNetB1 [55]	6,588,049	81.05	27.01	0.59
6	EfficientNetB2 [55]	7,782,659	80.08	31.80	0.68
7	EfficientNetB3 [55]	10,798,905	84.21	43.94	0.99
8	EfficientNetB4 [55]	17,691,753	78.24	71.66	1.54
9	EfficientNetB5 [55]	28,534,017	79.73	115.20	2.41
10	EfficientNetB6 [55]	40,983,193	82.45	165.18	3.43
11	EfficientNetB7 [55]	64,123,297	81.40	257.98	5.27
12	InceptionResnetV2 [66]	54,352,106	77.63	218.77	6.55
13	InceptionV3 [56]	21,823,274	78.42	87.97	2.89
14	MobileNetV2 [54]	2,270,794	86.75	9.49	0.32
15	MobileNetV3Large [67]	4,239,242	86.92	17.42	0.23
16	MobileNetV3Small [67]	1,540,218	81.84	6.55	0.06
17	NASNetMobile [57]	4,280,286	77.71	18.48	0.27
18	ResNet101V2 [53]	42,647,050	89.64	171.37	8.28
19	ResNet152V2 [53]	58,352,138	87.63	234.50	12.5
20	ResNet50V2 [53]	23,585,290	90.35	94.82	3.97
21	VGG16 [58]	14,719,818	87.98	58.98	15.5
22	VGG19 [58]	20,029,514	87.98	80.22	19.6
23	Xception [59]	20,881,970	82.89	83.96	0.36
24	**Proposed model**	**184,462**	**93.3**	**2.34**	**0.25**

providing it as input to the standard benchmark models. As observed in Table D.5, the proposed model achieves a relatively high test accuracy of 93.33% as compared to the benchmark models. It can also be observed from Table D.5, that the proposed model requires just 2.34 MB of storage space as compared to the benchmark models ensuring seamless portability and deployability on various edge computing devices. Additionally, the total parameters employed by our model are less as compared to other benchmark models. Moreover, the majority of the total parameters used by the proposed model are trainable parameters showing efficient utilization of parameters. Table D.5 also lists the computational performance of our model with respect to other benchmark models in terms of the number of floating point operations (FLOPs) [68]. The FLOP count is measured as GFLOPs where 1 GFLOP is equal to 10^9 FLOPs. The FLOP count is obtained by using standard open-source software available from PyTorch [69] and TensorFlow [70]. It can be observed that our model has a relatively less number of FLOPs as compared to most of the benchmark models. Specifically, MobileNetV3Small and MobileNetV3Large have lower FLOP counts compared to the proposed model. This reduction in computational cost might be due to the width and resolution multiplier parameter introduced in the MobileNet series [54].

The proposed model has also been deployed on edge computing devices such as

Raspberry Pi 4 Model B, NVIDIA Jetson Nano, and NVIDIA Jetson AGX Xavier. We perform inference time calculation of the proposed model on all these devices. The inference time calculation can serve as a useful reference when deciding the deployment feasibility of the proposed model for time-critical applications. In all three devices, we used a randomly selected test set which is 10% of the overall dataset. The trained model was executed on the Raspberry Pi 4 board to obtain an average inference time of about 127 milliseconds over 10 iterations. On the other hand, the execution on the Jetson Nano reported an average inference time of about 219 milliseconds. This is expected as the TensorFlow Lite models are not utilizing the GPU resources onboard the Jetson Nano. TensorFlow Lite does not support CUDA for GPU operations [71]. However, the Jetson devices only support CUDA for GPU operations [72] and hence the observed increase in inference time is expected. Similarly, the trained model was able to provide an average inference time of about 81.4 milliseconds, when executed on the NVIDIA Jetson AGX Xavier. The Jetson AGX Xavier showcased faster inference time as compared to the Raspberry Pi 4 and Jetson Nano. Even though the GPU is not utilized, the faster inference time can be attributed to the availability of increased RAM of about 32 GB which can increase the performance of the system. Further, the Jetson AGX Xavier also has access to additional computing resources as compared to the other two devices. Table D.6 lists the inference time obtained for the proposed model along with the benchmark models on various edge computing devices. We used the same data set split while calculating the inference time on the benchmark models. It can be observed that the proposed model is faster than most of the benchmark models on all three embedded devices. The MobileNet series of models however have a lower inference time than the proposed model. This might be due to the width and resolution multiplier parameter in the MobileNet series that reduces the computational cost of the model.

D.9 Discussion

In this work, we have provided a robust solution to estimate the number of UAVs in a scene. The current setup employs only one unidirectional cardioid-type microphone to estimate the number of UAVs. It is to be noted that since the polar pattern of the microphone follows a cardioid pattern, the acoustic disturbances originating from UAVs flying at the rear of the microphone are attenuated. This can severely impact the estimated UAV number. A more practical approach to overcome this limitation is to position multiple cardioid microphones such that the acoustic disturbances originating from the full 360° of the scene are captured. Employing microphones that exhibit an omnidirectional polar pattern can also be utilized so that acoustic disturbances from all directions are captured without significant signal attenuation.

It can be observed from Table D.5 and Table D.6, that the proposed CNN architecture provides relatively high accuracy and fast inference time on embedded hardware all the while consuming fewer resources. The proposed model can thus be employed for time-critical and resource-constrained UAV detection scenarios. High

Table D.6: Inference time calculation on various edge computing devices

Sl. No	Model	Inference time (seconds)		
		Raspberry Pi	Jetson Nano	Jetson AGX Xavier
1	DenseNet121 [52]	0.692	0.841	0.413
2	DenseNet169 [52]	0.832	1.009	0.483
3	DenseNet201 [52]	1.088	1.275	0.614
4	EfficientNetB0 [55]	0.389	0.372	0.116
5	EfficientNetB1 [55]	0.591	0.546	0.178
6	EfficientNetB2 [55]	0.624	0.592	0.180
7	EfficientNetB3 [55]	0.843	0.794	0.248
8	EfficientNetB4 [55]	1.162	1.147	0.359
9	EfficientNetB5 [55]	1.724	1.672	0.528
10	EfficientNetB6 [55]	2.339	2.173	0.832
11	EfficientNetB7 [55]	3.252	2.971	1.158
12	InceptionResNetV2 [66]	1.637	1.828	0.797
13	InceptionV3 [56]	0.700	0.819	0.378
14	MobileNetV2 [54]	0.088	0.112	0.046
15	MobileNetV3Large [67]	0.072	0.090	0.036
16	MobileNetV3Small [67]	0.022	0.028	0.011
17	NASNetMobile [57]	0.320	0.256	0.125
18	ResNet50V2 [53]	0.912	1.008	0.440
19	ResNet101V2 [53]	1.879	2.053	0.858
20	ResNet152V2 [53]	2.820	3.159	1.285
21	VGG16 [58]	3.903	4.042	1.699
22	VGG19 [58]	5.031	6.981	2.184
23	Xception [59]	1.236	2.206	0.520
24	**Proposed model**	**0.127**	**0.219**	**0.081**

detection performance coupled with real-time scenarios also suggests that the proposed technique can be deployed in practical ground control stations to function as an anti-UAV detection system. It can be inferred from the obtained results that the proposed technique is capable of detecting more than 10 UAVs in a dynamic real-time scenario given additional UAV information. In the future, the accuracy of the proposed technique can be improved by utilizing other sensor modalities. With the help of additional sensors, the work can also be extended to identify the UAV model and/or type.

D.10 Conclusion

In this article, we addressed the problem of accurately estimating the total number of UAVs present in a scene. We developed a UAV acoustic dataset to recreate a real-world scenario comprising of 10 UAV combinations flown in a random manner. The acoustic information from the dataset was preprocessed using time-frequency transformations to obtain their respective spectrogram images. The generated spectrogram images are then fed into a custom lightweight CNN model to estimate the

number of UAVs in the scene. The proposed model provides a high average test accuracy in accurately estimating the number of UAVs. Subsequently, the proposed model has also been executed on various edge computing devices to measure inference time performance. In the future, this work can be extended to identify the UAV model and/or type by utilizing information from additional sensors.

References

[1] Praveen Kumar Reddy Maddikunta et al. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sensors Journal*, 21(16):17608–17619, 2021.

[2] Chunxue Wu, Bobo Ju, Yan Wu, Xiao Lin, Naixue Xiong, Guangquan Xu, Hongyan Li, and Xuefeng Liang. Uav autonomous target search based on deep reinforcement learning in complex disaster scene. *IEEE Access*, 7:117227–117245, 2019.

[3] Junbiao Zhang et al. Trajectory prediction of hypersonic glide vehicle based on empirical wavelet transform and attention convolutional long short-term memory network. *IEEE Sensors Journal*, 22(5):4601–4615, 2022.

[4] Rodrigo Saar de Moraes and Edison Pignaton de Freitas. Multi-uav based crowd monitoring system. *IEEE Transactions on Aerospace and Electronic Systems*, 56(2):1332–1345, 2020.

[5] A. N. Wilson, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Embedded sensors, communication technologies, computing platforms and machine learning for UAVs: A review. *IEEE Sensors Journal*, 22(3):1807–1826, 2022.

[6] Jian Wang, Yongxin Liu, and Houbing Song. Counter-unmanned aircraft system(s) (c-uas): State of the art, challenges, and future trends. *IEEE Aerospace and Electronic Systems Magazine*, 36(3):4–29, 2021.

[7] Wei Nie et al. UAV detection and identification based on WiFi signal and RF fingerprint. *IEEE Sensors Journal*, 21(12):13540–13550, 2021.

[8] Tianyuan Yang et al. An adaptive radar signal processor for UAVs detection with super-resolution capabilities. *IEEE Sensors Journal*, 21(18):20778–20787, 2021.

[9] Alexander Sedunov et al. Stevens drone detection acoustic system and experiments in acoustics UAV tracking. In *Proc. IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7, 2019.

[10] Jesse Callanan, Payam Ghassemi, James DiMartino, Maulikkumar Dhameliya, Christina Stocking, Mostafa Nouh, and Souma Chowdhury. Ergonomic impact of multi-rotor unmanned aerial vehicle noise in warehouse environments. *Journal of Intelligent & Robotic Systems*, 100:1309–1323, 2020.

[11] Jesse Callanan, Rayhaan Iqbal, Revant Adlakha, Amir Behjat, Souma Chowdhury, and Mostafa Nouh. Large-aperture experimental characterization of the acoustic field generated by a hovering unmanned aerial vehicle. *The Journal of the Acoustical Society of America*, 150(3):2046–2057, 2021.

[12] Rayhaan Iqbal, Amir Behjat, Revant Adlakha, Jesse Callanan, Mostafa Nouh, and Souma Chowdhury. Efficient training of transfer mapping in physics-infused machine learning models of uav acoustic field. In *AIAA SCITECH 2022 Forum*, page 0384, 2022.

[13] Jianfei Tong, Wei Xie, Yu-Hen Hu, Ming Bao, Xiaodong Li, and Wei He. Estimation of low-altitude moving target trajectory using single acoustic array. *The Journal of the Acoustical Society of America*, 139(4):1848–1858, 2016.

[14] Alexander Sedunov, Alexander Sutin, and Hady Salloum. Application of cross-correlation methods for passive acoustic unmannded aierial vehicle detection and tracking. *The Journal of the Acoustical Society of America*, 140(4):3119–3119, 2016.

[15] Vladimir Kartashov et al. Use of acoustic signature for detection, recognition and direction finding of small unmanned aerial vehicles. In *Proc. IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, pages 1–4, 2020.

[16] Torea Blanchard, J-H Thomas, and Kosai Raoof. Acoustic localization and tracking of a multi-rotor unmanned aerial vehicle using an array with few microphones. *The Journal of the Acoustical Society of America*, 148(3):1456–1467, 2020.

[17] Giuseppe Ciaburro, Gino Iannace, and Amelia Trematerra. Research for the presence of unmanned aerial vehicle inside closed environments with acoustic measurements. *Buildings*, 10(5), 2020.

[18] Jian Fang, Anthony Finn, Ron Wyber, and Russell SA Brinkworth. Acoustic detection of unmanned aerial vehicles using biologically inspired vision processing. *The Journal of the Acoustical Society of America*, 151(2):968–981, 2022.

[19] Michael J Bianco, Peter Gerstoft, James Traer, Emma Ozanich, Marie A Roch, Sharon Gannot, and Charles-Alban Deledalle. Machine learning in acoustics: Theory and applications. *The Journal of the Acoustical Society of America*, 146(5):3590–3628, 2019.

[20] Aro Ramamonjy, Eric Bavu, Alexandre Garcia, and Sébastien Hengy. A distributed network of compact microphone arrays for drone detection and tracking. *The Journal of the Acoustical Society of America*, 141(5):3651–3651, 2017.

[21] Bilal Taha and Abdulhadi Shoufan. Machine learning-based drone detection and classification: State-of-the-art in research. *IEEE Access*, 7:138669–138682, 2019.

[22] Muhammad Zohaib Anwar, Zeeshan Kaleem, and Abbas Jamalipour. Machine learning inspired sound-based amateur drone detection for public safety applications. *IEEE Transactions on Vehicular Technology*, 68(3):2526–2534, 2019.

[23] Michael W Berry, Azlinah Mohamed, and Bee Wah Yap. *Supervised and unsupervised learning for data science*. Springer, 2019.

[24] Valentin V Gravirov, Ruslan A Zhostkov, and Dmitriy A Presnov. Acoustic fields of unmanned aerial vehicles in the tasks of passive detection. *The Journal of the Acoustical Society of America*, 149(4):A35–A35, 2021.

[25] Bowon Yang et al. UAV detection system with multiple acoustic nodes using machine learning models. In *Proc. IEEE International Conference on Robotic Computing (IRC)*, pages 493–498, 2019.

[26] Andrea Bernardini, Federica Mangiatordi, Emiliano Pallotti, and Licia Capodiferro. Drone detection by acoustic signature identification. *Electronic Imaging*, 2017(10):60–64, 2017.

[27] Juhyun Kim, Cheonbok Park, Jinwoo Ahn, Youlim Ko, Junghyun Park, and John C. Gallagher. Real-time uav sound detection and analysis system. In *IEEE Sensors Applications Symposium (SAS)*, pages 1–5, 2017.

[28] Xuejun Yue, Yongxin Liu, Jian Wang, Houbing Song, and Huiru Cao. Software defined radio and wireless acoustic networking for amateur drone surveillance. *IEEE Communications Magazine*, 56(4):90–97, 2018.

[29] Wenshuai Wang, Kuangang Fan, Qinghua Ouyang, and Ye Yuan. Acoustic uav detection method based on blind source separation framework. *Applied Acoustics*, 200:109057, 2022.

[30] Sungho Jeon, Jong-Woo Shin, Young-Jun Lee, Woong-Hee Kim, YoungHyoun Kwon, and Hae-Yong Yang. Empirical study of drone sound detection in real-life environment with deep neural networks. In *Proc. European Signal Processing Conference (EUSIPCO)*, pages 1858–1862, 2017.

[31] Yoojeong Seo, Beomhui Jang, and Sungbin Im. Drone detection using convolutional neural networks with acoustic stft features. In *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018.

[32] Luke Russell, Rafik Goubran, and Felix Kwamena. Emerging urban challenge: Rpas/uavs in cities. In *Proc. International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 546–553, 2019.

[33] Pietro Casabianca and Yu Zhang. Acoustic-based UAV detection using late fusion of deep neural networks. *Drones*, 5(3), 2021.

[34] Sara Al-Emadi, Abdulla Al-Ali, Amr Mohammad, and Abdulaziz Al-Ali. Audio based drone detection and identification using deep learning. In *Proc. International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 459–464, 2019.

[35] DJI Mavic 2. [Online]. Available: https://www.dji.com/no/mavic-2-enterprise.

[36] DJI Mini 2. [Online]. Available: https://www.dji.com/no/mini-2.

[37] DJI Mini SE. [Online]. Available: https://www.dji.com/no/mini-se.

[38] DJI Mini 3 Pro. [Online]. Available: https://www.dji.com/no/mini-3-pro.

[39] DJI Tello EDU. [Online]. Available: https://store.dji.com/no/product/tello.

[40] SYMA X30 Foldable Drone. [Online]. Available: https://www.symatoys.com/goodshow/x30-syma-x30-foldable-drone.html.

[41] Shure MV7. [Online]. Available: https://www.shure.com/en-MEA/products/microphones/mv7?variant=MV7-K.

[42] Shure MV7 User Guide. [Online]. Available: https://pubs.shure.com/guide/MV7/en-US.

[43] Yoon Young Kim and Eung-Hun Kim. Effectiveness of the continuous wavelet transform in the analysis of some dispersive elastic waves. *The Journal of the Acoustical Society of America*, 110(1):86–94, 2001.

[44] L. Durak and O. Arikan. Short-time fourier transform: two fundamental properties and an optimal implementation. *IEEE Transactions on Signal Processing*, 51(5):1231–1242, 2003.

[45] Continuous Wave Transform (CWT. [Online]. Available: https://www.mathworks.com/help/wavelet/gs/continuous-wavelet-transform-and-scale-based-analysis.html.

[46] Aveen Dayal, Sreenivasa Reddy Yeduri, Balu Harshavardan Koduru, Rahul Kumar Jaiswal, J Soumya, MB Srinivas, Om Jee Pandey, and Linga Reddy Cenkeramaddi. Lightweight deep convolutional neural network for background sound classification in speech signals. *The Journal of the Acoustical Society of America*, 151(4):2773–2786, 2022.

[47] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proc. International Conference on Machine Learning (ICML)*, pages 111–118, 2010.

[48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning (ICML)*, pages 448–456. PMLR, 2015.

[49] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[50] Keras. [Online]. Available: https://keras.io.

[51] Tesla V100 GPU. [Online]. https://www.nvidia.com/en-us/data-center/v100/.

[52] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing.

[54] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[55] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[57] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proc. IEEE CVPR*, pages 8697–8710, 2018.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[59] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[60] ImageNet. [Online]. Available: https://www.image-net.org.

[61] Raspberry Pi 4 Model B - Datasheet. [Online]. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf.

[62] NVIDIA Jetson Nano. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[63] NVIDIA Jetson AGX Xavier. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/.

[64] TensorFlow Lite. [Online]. Available: https://www.tensorflow.org/lite.

[65] [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html.

[66] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proc. AAAI conference on artificial intelligence*, volume 31, 2017.

[67] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proc. IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[68] Hojun Son and James Weiland. Semantic segmentation optimized for low compute embedded devices. *IEEE Access*, 10:96514–96525, 2022.

[69] Torchstat (PyTorch). [Online]. https://pypi.org/project/torchstat/.

[70] Model Profiler (TensorFlow). [Online]. Available: https://pypi.org/project/model-profiler/.

[71] GPU Delegates - TensorFlow Lite. [Online]. Available: https://www.tensorflow.org/lite/performance/delegates.

[72] Shan Ullah and Deok-Hwan Kim. Benchmarking jetson platform for 3d point-cloud and hyper-spectral image classification. In *Proc. IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 477–482, 2020.

Appendix E

PAPER E

Title: Multi-target Angle of Arrival Estimation Using Rotating mmWave FMCW Radar and Yolov3

Authors: **A. N. Wilson**, A. Kumar, A. Jha, L. R. Cenkeramaddi

Journal: IEEE Sensors Journal

Multi-target Angle of Arrival Estimation Using Rotating mmWave FMCW Radar and Yolov3

A. N. Wilson, A. Kumar, A. Jha, L. R. Cenkeramaddi

Abstract: **It is still challenging to accurately localize unmanned aerial vehicles (UAVs) from a ground control station (GCS) using various sensors. The mmWave frequency modulated continuous wave (FMCW) radars offer excellent performance for target detection and localization in harsh environments and low lighting conditions. However, the estimated angle of arrival (AoA) of targets in the captured scene is quite poor. This article focuses on improving AoA estimation by combining cutting-edge machine learning algorithms with a mechanical radar rotor setup. A mmWave FMCW radar system is mounted on a programmable rotor to capture range-angle maps of targets at various locations. The range-angle images are then labeled and trained further with the Yolov3 algorithm. Subsequent testing reveals that for detected target objects, the centroid of the bounding boxes from the detected objects provides accurate AoA estimation with very low root mean square error (RMSE). The results show that the proposed approach outperforms traditional methods in terms of performance and estimation accuracy.**

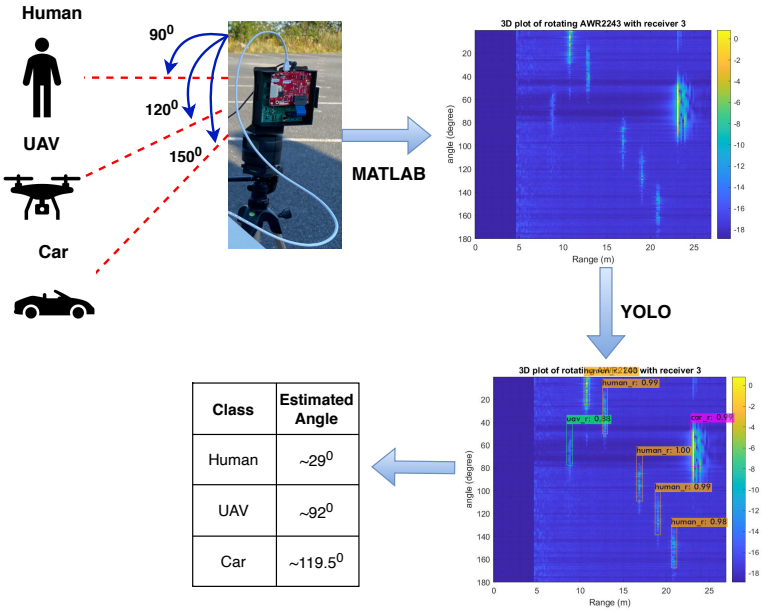


Figure E.1: Block diagram of the proposed multi-target AoA estimation approach.

E.1 Introduction

As the popularity of unmanned aerial vehicles (UAV) increases, there is a growing demand for better detection, localization and tracking techniques. Ground control stations (GCS) are equipped with sophisticated sensor technologies for improved UAV localization. Some of these sensors are RGB cameras, ultrasonic sensors, LiDARs, etc. However most of these sensors fail in adverse weather and lighting conditions. In this regard, the mmWave frequency modulated continuous wave (FMCW) radars have shown superior performance compared to other sensors. The mmWave FMCW radars are small, lightweight, and compact radars that offer excellent performance for target detection. The radars provide a radial distance measurement range from 0.2 meters to 300 meters in addition to excellent velocity estimation performance. Additionally, they offer high range and velocity resolutions. Further, their performance in adverse weather and environmental conditions makes them a suitable choice for UAV and GCS applications.

Despite the above-mentioned advantages, the mmWave FMCW radars suffer from poor angle of arrival (AoA) estimation and AoA resolutions [1]. Accurate AoA estimation from a single target requires atleast one transmitter and two receivers for the radar. Increasing the number of transmitter and receiver pairs can effectively improve the AoA estimation, however this results in a tradeoff with increased size and hardware complexity. Furthermore, estimation of AoA from multiple objects for better target localization is even more challenging and hence it is still an ongoing research topic.

There has been several attempts in the literature to improve the AoA estimation. The authors in [2] propose an adaptive radar signal processor for detection of multiple UAVs in the range-Doppler domain. The radar signal processor operates by initially performing pulse compression (PC), FFT and beamforming on the received signal samples. The processed signal is then evaluated for a possible detection based on constant false alarm rate (CFAR) logic for each range-Doppler angle bin. Both single and multi-target scenarios are considered and the angular parameters of the targets are retrieved. Simulated and real-world experiment results indicate that the proposed method is able to detect and resolve two UAVs placed $\approx 8^{\circ}$ apart with satisfactory performance. In [3], Aubry *et al.* attempt to solve the problem of adaptive radar detection in the presence of limited training data. The proposed method consists of three generalised likelihood ratio test (GLRT) based detection schemes that is based on the assumption of linear combination for inverse covariance matrices. Reported results show that the proposed GLRT-1 and GLRT-3 exhibit significant performance gain over conventional adaptive detectors in presence of limited training samples. In [4], a novel DoA estimation algorithm based on 2-D spectrum sensing is developed. The proposed approach exploits the intrinsic block-sparsity of the 2-D space-frequency profile to obtain 2-D occupancy awareness. By including a term to account for the block-sparsity, a non-convex regularized maximum likelihood (RML) estimation problem is solved using the block sparse learning via iterative minimization (BSLIM) algorithm to obtain the space-

frequency profile. The BSLIM algorithm exhibits superior performance in terms of detection rate and false alarms over traditional approaches. In [5], the authors devise a novel set of covariance matrix estimators called median matrices that are independent of the probability distribution of the samples. The matrices are constructed by exploiting its positive definite attributes. Based on the output from the new estimators, a generalised inner product (GIP) selection criteria is utilized to discard the secondary outliers in the training data. Reported results indicate that the log-Euclidean median-based estimator outperforms other estimators in terms of selection probability of secondary outliers in the training data. Other covariance estimators that can also be utilized to infer radar disturbances and aid in accurate UAV detection and localization include the fast maximum likelihood (FML) covariance estimator [6] and the multi-class inverse Wishart mixture (MC-IWM) filter [7].

The work by [8], uses a fast iterative adaptive algorithm employing a time-shared method to accurately estimate the AoA. The first phase of the technique involves a coarse azimuth angle estimation that is performed using just a single mmWave FMCW radar. The obtained intermediate results are then fed into a more refined iterative algorithm to accurately determine the direction of arrival (DOA) estimations within a small region. However, this method relies on multiple mmWave FMCW radars during the second phase making it computationally intensive. Another approach by [9] uses linear algebraic techniques to jointly calibrate the antennas and estimate the AoA. The iterative optimization technique provides excellent performance when there is an effective coupling between the antennae elements. Additionally, the algorithm is shown to work well in presence of several sensors and transmitter-receiver pairs. Daegun *et al.* in [10] proposed a joint angle and delay estimation algorithm which utilizes the dual-shift-invariant structure of the received signal to extend the one-dimensional pseudospectrum searching. Monte Carlo simulations were used to gauge the performance of the proposed method. Reported results show that the proposed technique exhibited superior performance with respect to state-of-the-art algorithms. However, the technique is computationally intensive and requires further experimentation with complex real-world scenarios. In [11], the authors combine the advantages of the 2D-FFT and MUSIC algorithms to develop a low complexity estimation method to approximate the various FMCW radar parameters such as AoA. However, this method fails to provide any performance insight when used to detect multiple targets. In [12] and [13], the authors use the notion of rotating the radar about its axis to scan an area. The signals obtained in each frame of the rotating radar are used to construct the range profile for the respective field of view. All the range profiles are then stitched and combined together to get the range-angle maps that offer a 180° view of the scene. However, the work by [12] focuses on target detection and classification and doesn't provide enough information for AoA estimation. Linga *et al.* in their work [13], has provided a novel AoA estimation technique with relatively good performance, however this can be further improved using machine learning techniques. Our work will be an extension to this work, where we will use state-of-the-art machine learning techniques

to estimate the AoA from the range-angle maps. We will be using convolutional neural networks (CNN) based algorithms which will serve to be useful for embedded and internet-of-things (IoT) related applications.

Hence, in this work, we propose a multi-target AoA estimation method that utilizes rotating mmWave FMCW radars. The proposed method employs a mechanical rotor setup for rotating the mmWave FMCW radar along the azimuthal axis followed by Yolov3 for post-processing the range-angle heatmaps. The AoA estimate is calculated from the centroid of the detected bounding box.

The remaining sections of this paper is structured as follows. Section II gives an overview of mmWave FMCW radars and signal processing aspects. Section III gives a brief overview of the machine algorithm that is used on the collected dataset. Section IV defines the various metrics used to measure the performance of the proposed method. Section V provides a high level system description along with the measurement and dataset details. Section VI summarizes the obtained results using the proposed method. And finally, Section VII concludes the paper by providing a brief summary and potential future research directions.

E.2 mmWave FMCW Radars and Signal Processing

The mmWave FMCW radars operate by transmitting high frequency chirp signals on to the scene. The transmitted chirp signal is reflected back upon encountering obstacles in its path. The reflected chirp is captured by the receiving antennae for further processing. The transmitted and received chirp signals are then fed into a mixer to obtain the intermediate frequency (IF) signal. The IF signal is further fed into an ADC which samples the analog signal and provides digital values. The raw IF samples then undergo additional processing to obtain the radial range, radial velocity and AoA estimation of the target [14].

Range estimation involves performing an FFT transform over the captured IF signal samples [15]. The range R is calculated as,

$$R = \frac{cf_{IF}}{2S}, \tag{E.1}$$

where f_{IF} is the frequency of the IF signal, c is the velocity of light in vacuum (3x10^8 m/s), and S is the slope of the radar.

Velocity estimation exploits the phase of the IF signal. Change in distance within a short time may not be accurately captured by the range estimation (due to limitations in range resolution) procedure. However, these changes can be captured by utilizing the phase difference between the received chirps. To obtain the velocity estimation, we initially compute the range profile. Next, we perform a second FFT across the received chirp signals to capture the small phase changes that can provide the velocity estimation values.

AoA estimation involves utilizing the number of receiver antennas on the mmWave FMCW radar. A differential distance exists from an object to each of the receiving antennas. This differential distance corresponds to a phase change that

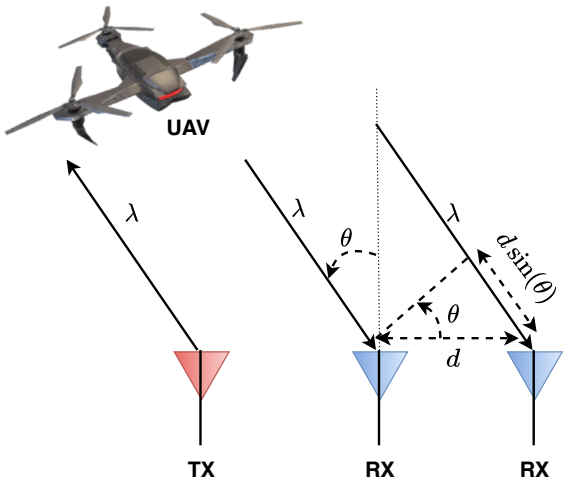


Figure E.2: Angle of arrival estimation.

can be obtained by performing a second FFT on the output of the range profile. The second FFT is applied across the different receiver antennas so as to obtain the angle-FFT. The angle-FFT can then be used to obtain AoA estimation of target objects in the scene. Note that in angle-FFT, the 2D-FFT is performed over the different receiver antennas separated in space whereas for velocity estimation the 2D-FFT utilizes the phase difference between the different chirps that are separated in time [16]. If ϕ is the phase difference between the received chirp signals and d the distance between adjacent receiving antennas, then the following holds true,

$$\phi = \frac{2\pi d \sin\theta}{\lambda}, \tag{E.2}$$

where λ is the wavelength of the chirp signal. Thus the AoA estimation θ is given as,

$$\theta = \sin^{-1}(\frac{\lambda\phi}{2\pi d}) \tag{E.3}$$

Fig E.2. gives a pictorial representation of the estimation of AoA. As seen from (E.2) and (E.3), when $\theta = 0°$, ϕ is more sensitive to changes in θ. As θ increases, the sensitivity of ϕ to θ decreases. This is due to the fact that ϕ and θ share a non-linear relationship as $\phi \propto \sin\theta$ and sensitivity of $\sin\theta$ decreases as θ increases. Thus AoA estimation is more accurate when the target is placed perpendicular to the radar [16].

Similar to AoA estimation, calculation of AoA resolution is also important to correctly distinguish between targets placed close to each other with a small AoA difference. The AoA resolution is defined as the least angle required between two targets so that they can be easily distinguished as separate objects in the angle-FFT [17]. The AoA resolution for mmWave FMCW radars is given as,

$$\theta_{res} = \frac{\lambda}{Nd\cos\theta}, \tag{E.4}$$

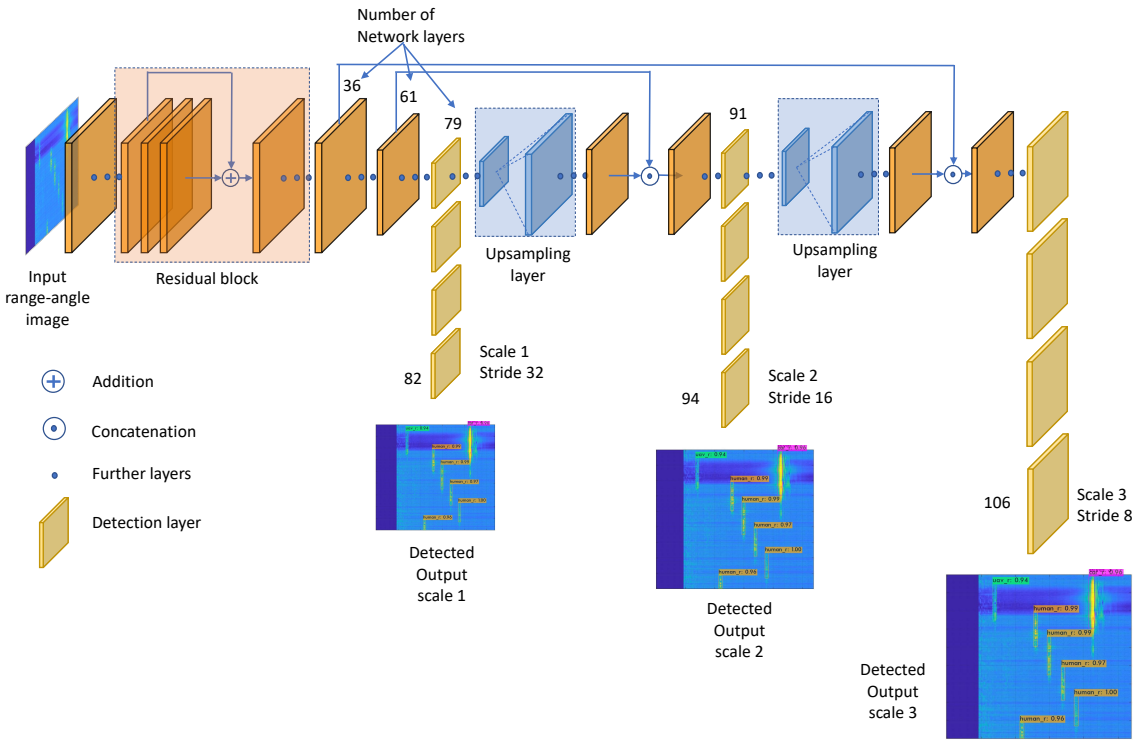


Figure E.3: Yolov3 Darknet Architecture

where N is the number of receiver antennas on the radar. If it is assumed that $d = \lambda/2$ and $\theta = 0$, then the expression for θ_{res} reduces to,

$$\theta_{res} = \frac{2}{N}, \tag{E.5}$$

implying that the AoA resolution improves with increased number of receiver antennas. For example, with 4 receiver antennas, the AoA resolution is 0.5 radians which is equivalent to 28°, a relatively high value. Since increasing the number of antennas can also increase space and hardware complexity, there is active research to improve the AoA resolution with minimum constraints.

E.3 You Look Only Once (Yolo) Architecture and Working

The Yolov3 is a state-of-the-art multi-scale object detection algorithm that can detect and classify objects from images and video with very fast inference time [18]. Just as the name suggests, the algorithm performs only a single-stage forward pass over the whole image to determine the class probabilities and predictions. In this work, we are using the Yolov3 version for our AoA estimation from range-angle maps.

The Yolov3 algorithm is based on the Darknet-53 architecture [19] that uses a combination of convolution layers and skip connections as shown in Fig. E.3. Inspired by ResNet and other architectures [20], Darknet-53 primarily consists of

feature extraction and feature detection stages. The feature extraction stage comprises of 53 convolution layers that are arranged as 3x3 and 1x1 consecutive layers followed by a skip connection. On the other hand, the feature detection stage is made up of 53 layers that helps to enhance the accuracy of the predictions as compared to previous versions. Thus the algorithm employs a sum total of 106 convolutional layers, both for feature extraction and feature detection.

Table E.1: Yolov3 configuration parameters

Parameter	**Value**
Batch Size	64
Number of classes	3
Max Batches	6000
Sub division	16
Learning rate	0.001
Momentum	0.9
Decay	0.0005
Filters	24
GPU version	Tesla V100-SXM3
CUDA version	11.4

The Yolov3 algorithm operates by initially passing the input image through the feature extractor to obtain multi-scale feature embeddings. Yolov3 supports three different feature scales namely, 13x13, 26x26, and 52x52. The obtained feature maps are then fed into the feature detection stage to predict bounding boxes on the detected object. To predict bounding boxes, the each feature map is divided into a collection of grid cells. Each grid cell is capable of predicting three bounding boxes. Each bounding box 'B' consists of '$5 + C$' attributes and class probabilities. Here, 'C' represents the number of classes in the dataset. In this work, $C = 3$, as we are using the model to detect three classes: UAV, car, and humans. The class probabilities provide the probability of existence of a particular class in the respective grid cell. The '5' represents bounding box attributes (t_x, t_y, t_h, t_w) and the objectness scores. Bounding box dimensions in Yolov3 are computed relative to the anchor box dimensions. Anchor boxes are prior boxes that have predefined aspect ratios. The predefined aspect ratios are determined by running a k-means algorithm on the entire dataset prior to training. Lastly, the objectness score denotes the presence of an object in the corresponding grid cell. Feature detection is performed by convolving the downsampled feature maps with a 1x1 detection kernel whose shape is given by 1x1x$(B * (5 + C))$. In Yolov3, feature detection occurs at the 82^{th}, 94^{th}, and 106^{th} layers.

Once the object is detected, a maximum of three bounding boxes per grid can be drawn based on the relative position of the object within the grid cell. To avoid multiple bounding boxes for the same object, a non-maximum suppression is performed. In non-maximum suppression, the bounding box that has the most

overlapping area with the detected object is retained while the other bounding boxes are discarded. A high level Yolov3 architecture diagram is given in Fig E.3. Table E.1 further lists the various parameters used for the Yolov3 algorithm to detect the various classes from the range-angle maps.

E.4 Performance Metrics

In this section, the various metrics used to quantify the performance of the machine learning model are discussed [21]. A machine learning model can predict the class of an object as either true (positive) or false (negative). When the model predicts the class as true and the actual class is positive, then the observation is defined as a true positive (TP). Similarly, when the model predicts the class correctly as negative, then the observation is a true negative (TN). However, when the predicted class is true and the actual class is negative, then the observation is a false positive (FP). Similarly, the observation is defined as a false negative (FN) when the predicted class is false and the actual class is positive.

E.4.1 Precision

Precision represents the fraction of total number of positively classified classes to the total number of positively predicted classes. Precision is given as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{E.6}$$

where TP and FP represents the true positives and false positives from the predicted result.

E.4.2 Recall

Recall is calculated as the ratio of true positive to the total number of positive classes. Recall is computed as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{E.7}$$

where FN is the false negatives in the predicted results.

E.4.3 F1-score

F1-score provides a means to measure the performance of a machine model by utilizing both precision and recall metrics. F1-score is calculated as follows:

$$\text{F1-score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{E.8}$$

E.4.4 Accuracy

The accuracy metric provides an indication of the performance of the model across all the classes. It is defined as follows:

$$\text{Accuracy} = \frac{TN + TP}{TN + FP + TP + FN} \tag{E.9}$$

E.4.5 Root mean square error (RMSE)

The RMSE value provides a measure of how much the estimated value has deviated from the actual true value. It is given as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{P}(x_{estimate} - x_{actual})^2}{P}} \tag{E.10}$$

where P is the sample size, $x_{estimate}$ is the estimated value, and x_{actual} is the actual or true value in the experiment. Lower RMSE value implies that the estimated measurement is closer to the true value and hence better model prediction.

E.5 System Overview and Dataset Details

E.5.1 System Description

The AoA estimation of multiple targets using mmWave FMCW radar is challenging. As described in Section E.2, accurate AoA estimation requires an increased number of transmitter and receiver antennas. This can lead to increased hardware complexity. Hence in our setup, the system at the GCS is made up of mmWave FMCW radar that utilizes only one transmitter-receiver antenna to estimate the AoA.

Our setup consists of a mmWave FMCW radar that is mounted firmly on a rotor. The rotor in itself is mounted on a static tripod. The rotor is battery-powered, programmable, and highly portable. The mmWave radar is mounted on the head of the rotor. The head of the rotor can rotate the radar through the entire 360° in the azimuth direction based on requirements.

The parameters of the rotor that can be programmed include the rotational direction, the rotational velocity, and the rotational angle. The rotational velocity of the rotor can be set according to the number of chirp frames transmitted by the radar per second. Here the chirp frame or simply frame is defined as a set of K equally spaced chirp waveforms that are transmitted by the radar [14]. The performance of the AoA estimation is directly influenced by the rotational velocity of the rotor. If the rotational velocity of the rotor is high, then the number of frames per second that are captured will be less. Similarly, if the rotational velocity is low, then the captured number of frames per second from the radar is high. Improved AoA estimation performance requires higher number of frames per second, as higher number of frames can capture more information from multiple targets present in the scene. Hence in our setup, we have programmed the rotational velocity of the rotor to be low so as to capture at least one frame per degree of rotation of the radar.

Table E.2: mmWave FMCW radar AWR1843 and AWR2243 Parameters

Radar Parameter	**Value**
No. of frames (AWR2243)	200
No. of frames (AWR1843)	800
Frame periodicity	40 ms
No. of Tx antennas	3
No. of Rx antennas	4
No. of ADC samples	256
Sampling Rate	10 MSPS
Frequency Range (RF)	$77 - 81$ GHz
Bandwidth (RF)	1798.92 MHz
No. of Chirps	128
Chirp Slope	29.982 MHz/μs
Rx Noise Figure	14 dB (76 to 77 GHz) 15 dB (77 to 81 GHz)
Transmission Power	12 dBm

Table E.3: Experiment Parameters

Experiment Parameter	**Value**
UAV Size	$32.24 \times 24.2 \times 8.4$ cm^3
Human Height	172 cm
Car Size	$431.5 \times 178.0 \times 160.5$ cm^3
Measurement Range	upto 26 meters

Based on the adjusted rotational velocity, let the desired FoV to be covered by the rotor in t seconds be θ_{FoV}. In a t second duration, the radar transmits n_f frames. By considering that the entire θ_{FoV} is divided into smaller angle bins (θ_b), we obtain the following relation,

$$\theta_b = \frac{\theta_{FoV}}{n_f} \tag{E.11}$$

In our experiment scenario, the desired FoV is 180°. The n_f is 800 and 200 frames for the radars AWR2243 [22] and AWR1843 [23] respectively. Each frame is made up of 128 chirp waveforms transmitted by the radar. This corresponds to 0.225° per frame for AWR2243 and 0.9° per frame for AWR1843. To increase the FoV in the elevation, the radar is placed vertical to the ground plane in our setup.

The two mmWave FMCW radars, AWR2243 [22] and AWR1843 [23] that we use in our experiment are manufactured from Texas Instruments. Both radars have identical frequency range spanning between $77 - 81$ GHz. Additionally, they have identical number of transmitter and receiver pairs, RF bandwidth, chirp slope, sampling rate, and ADC samples. However, the number of frames employed is 800 for the AWR1843 and 200 for the AWR2243. A detailed list of the key parameters of the radars can be found in Table. E.2.

Table E.4: Measurement cases for Set1_UavCarHumans (Range in meters and angle in degrees)

Range (m) → / Cases ↓	5	7	9	11	13	15	17	19	21	23	25
AA	U(0°)		H1(30°)	H2(60°)	H3(90°)	H4(120°)	H5(150°)	C(0°)			
BB		U(30°)		H5(180°)	H1(60°)	H2(90°)	H3(120°)	H4(150°)	C(30°)		
CC			U(60°)	H1(0°)	H2(30°)		H3(90°)	H4(120°)	H5(150°)	C(60°)	
DD				U(90°)	H1(0°)	H2(30°)	H3(60°)		H4(120°)	H5(150°)	C(90°)
EE	U(120°)	H1(60°)	H2(90°)		H3(150°)	H4(0°)	C(180°)	H5(30°)			
FF		U(150°)	H5(0°)			C(150°)	H1(30°)	H2(60°)	H3(90°)	H4(120°)	
GG	H1(60°)	H2(90°)	U(180°)	H3(150°)	C(120°)				H4(0°)	H5(30°)	
HH	U(30°)	H4(0°)	H3(150°)	H2(120°)				H1(90°)		C(180°)	H5(60°)
II	H1(90°)	H2(120°)		U(30°)	H3(180°)	H4(60°)	H5(0°)				C(150°)
JJ	U(150°)	H3(180°)	H1(120°)						H2(60°)	H4(90°)	C(30°)
KK	H3(150°)	U(60°)			H4(90°)		H5(30°)		H1(180°)	H2(0°)	C(120°)
LL	H2(180°)	H1(120°)	U(30°)	H3(60°)		H4(90°)		H5(150°)			C(0°)
MM		H2(0°)	U(90°)	H3(30°)	H4(120°)	H1(180°)	H5(150°)				C(180°)
NN	U(90°)	H1(30°)	H3(60°)	H4(120°)	H5(150°)		H2(0°)	C(180°)			

U - UAV, C - Car, H1 - Human 1, H2 - Human 2, H3 - Human 3, H4 - Human 4, H5 - Human 5.

Table E.5: Measurement cases for Set2_UavCar (Range in meters and angle in degrees)

Range (m) → / Cases ↓	5	7	9	11	13	15	17	19	21	23	25
AA		U(0°)									C(60°)
BB			U(0°)					C(60°)			
CC				U(0°)			C(90°)				
DD	U(60°)					C(120°)					
EE			U(60°)				C(150°)				
FF				U(60°)			C(180°)				
GG		U(90°)							C(180°)		
HH		U(120°)								C(150°)	
II			U(120°)			C(0°)					
JJ				U(120°)					C(0°)		
KK			U(150°)				C(0°)				
LL				U(150°)						C(90°)	
MM	U(180°)					C(60°)					
NN		U(180°)							C(120°)		
OO				U(180°)							C(30°)

U - UAV, C - Car.

The measurement scenario is an outdoor parking setting with UAV, car, and humans placed at varying locations with respect to the radar. Each measurement is taken by positioning UAV, car, and humans at different locations within a radial distance of $\approx$ 26 meters and 180° FoV in front of the radar. We limit the radial distance to $\approx$ 26 meters to remain within the permissible measurement area. However, the mmWave FMCW radars can measure targets with range upto 300 meters [22],[23]. Depending upon the radar version used (AWR1843/AWR2243), a total of 800/200 frames containing the raw IF signal is captured for each measurement. Each frame of the radar has a duration of about 40 ms and consists of 128 chirps. Further, each frame of the radar provides raw IF signal data corresponding to a

Table E.6: Measurement cases for Set3_OnlyHumans (Range in meters and angle in degrees)

Range (m) → / Cases ↓	5	7	9	11	13	15	17	19	21	23	25
AA	H1(0°)	H2(30°)	H3(60°)	H4(90°)	H5(120°)						
BB		H3(0°)	H4(30°)	H5(60°)		H1(150°)	H2(180°)				
CC			H5(0°)		H1(90°)	H2(120°)	H3(150°)	H4(180°)			
DD				H1(30°)	H2(60°)	H3(90°)	H4(120°)	H5(150°)			
EE				H2(0°)	H3(30°)	H4(60°)	H5(90°)		H1(180°)		
FF					H4(0°)	H5(30°)		H1(120°)	H2(150°)	H3(180°)	
GG							H1(60°)	H2(90°)	H3(120°)	H4(150°)	H5(180°)
HH						H1(0°)	H2(30°)	H3(60°)	H4(90°)	H5(120°)	
II							H2(0°)	H3(30°)	H4(60°)	H5(90°)	H1(150°)
JJ								H2(0°)	H3(30°)	H4(60°)	H5(90°), H1(120°)
KK	H4(150°)	H5(180°)							H1(0°)	H2(30°)	H3(60°)
LL	H3(120°)	H4(150°)	H5(180°)							H1(0°)	H2(30°)
MM	H2(90°)	H3(120°)	H4(150°)	H5(180°)							H1(0°)
NN	H1(60°)	H2(90°)	H3(120°)	H4(150°)	H5(180°)						
OO	H1(30°)	H2(60°)	H3(90°)	H4(120°)	H5(150°)						

H1 - Human 1, H2 - Human 2, H3 - Human 3, H4 - Human 4, H5 - Human 5.

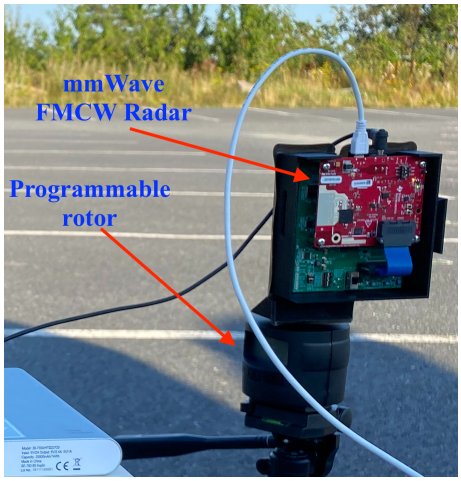


Figure E.4: The rotating mmWave FMCW radar measurement setup.

0.225°/0.9° FoV depending upon the radar model. The collected raw IF signal data is post-processed in MATLAB [24] to obtain the range-profile for each frame. Finally, the range profiles for all the frames are stitched together so as to obtain a 180° FoV range-angle map of the whole measurement scenario that comprises of multiple targets [12]. Fig. E.4 shows the experimental setup of the rotating mmWave FMCW radar that is used to capture the raw IF signals. Other relevant parameters related to the experiment can be found in Table. E.3.

The range-angle images obtained from MATLAB is fed into the Yolov3 algorithm for training. We implement a 10-fold scheme for validation. In this scheme, the training is performed for 10 folds of the experiment. In each fold, only 80% of the dataset images are used for training and the remaining are assigned as test images. We utilize the pretrained weights available from the Darknet-53 repository to initiate the training. Upon training, the optimum weights are obtained indepen-

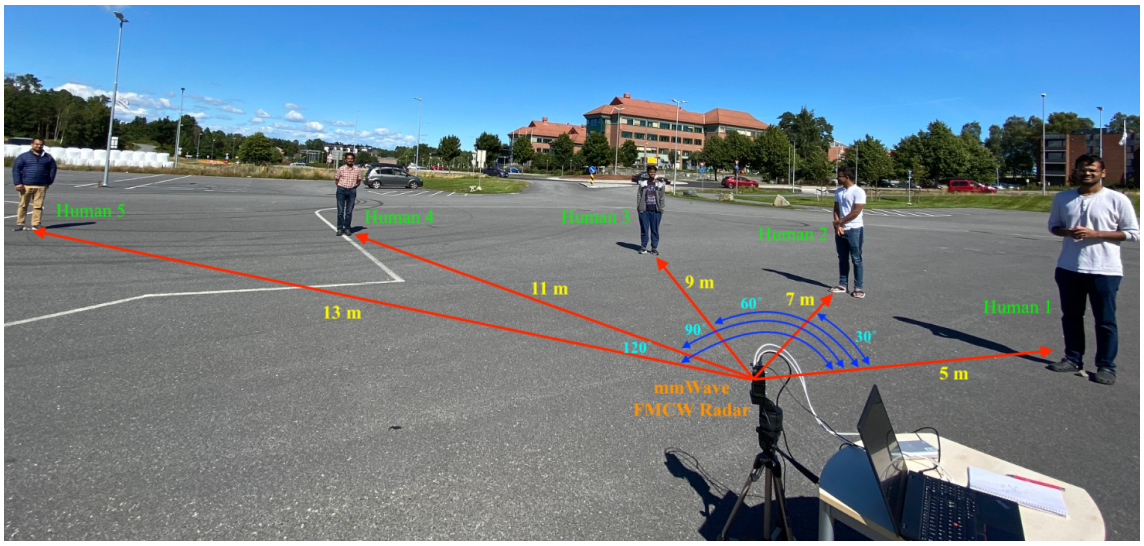


Figure E.5: Measurement setup for case AA from Set3_OnlyHumans.

dently for each fold. Thus the weights of the 1^{st} fold are independent of the 2^{nd} fold, and so on. By using this approach, we are implicitly removing any bias that is associated with the training. The weights obtained after training are used on the test images to obtain the predictions. The predictions from the Yolov3 algorithm are range-angle images that contain bounding boxes on the detected object. It is to be noted that each pixel dimension in the range-angle image corresponds to a specific range and angle in the measurement setup. Hence, the centroid coordinates of the detected bounding box correspond to the target object's estimated range and AoA in the measurement setup. Once the AoA estimation is obtained, the algorithm performance can be calculated by computing the root mean square error between the AoA estimation and the ground truth angle of the target.

E.5.2 Dataset Details

Based on the above measurement setup, we have collected the raw IF signal data from two mmWave FMCW radars, the AWR2243 and the AWR1843 for different target objects. The target objects include combinations of UAV, car, and humans positioned at various locations in front of the radar. We used the AWR2243 mmWave FMCW radar to capture the raw IF signals of UAV, car, and humans. Similarly, we used the AWR1843 mmWave FMCW radar to collect the raw IF signals of humans. The collected raw IF signals are processed in MATLAB to obtain the range profile for each frame. The range profiles for each frame are stitched together to obtain the radar range-angle images. We have created three datasets based on these range-angle images. The first dataset contains UAV, car, and humans while the second dataset comprises of only UAV and car combination. The third dataset contains range-angle images of only human targets. The first, second, and third datasets are named as Set1_UavCarHumans, Set2_UavCar, and Set3_OnlyHumans respectively. The range-angle images obtained after processing from MATLAB have a resolution of 875×656 pixels for Set1_UavCarHumans and Set2_UavCar. Similarly, the range-

Table E.7: RMSE and accuracy values for 10 folds

Sl No.	**UAV (RMSE)**	**Car (RMSE)**	**Human (RMSE)**	**Accuracy (%)**
1	1.4746°	1.3234°	0.9644°	99.3902
2	0.7869°	1.4576°	0.8635°	98.4939
3	1.3661°	1.0667°	1.0702°	97.9166
4	0.8700°	1.3807°	1.2312°	98.6111
5	0.9305°	1.2518°	1.1380°	98.7012
6	0.7274°	1.2540°	0.8898°	98.4423
7	0.8963°	1.6049°	1.1868°	96.8838
8	1.4128°	1.2237°	1.1005°	98.6413
9	1.3441°	1.2035°	0.9511°	95.5882
10	0.9922°	1.1559°	0.8297°	99.3569

angle images in Set3_OnlyHumans have a resolution of 1167×875 pixels.

The dataset measurement cases are detailed in Table. E.4, E.5, and E.6 respectively. The different cases are labelled as AA, BB, CC ....OO depending upon the dataset. The entries inside the table are represented as an object that is positioned at a specified distance and angle with respect to the radar. The abbreviation for the labels used in the table entries are given towards the bottom of the respective tables. Hence, the first entry in case AA of Table. E.6 depicts a person at 5 m and 0° with respect to the radar. The rest of the entries can be interpreted as follows: a second person at 7 m and 30°, the third person at 9 m and 60°, the fourth person at 11 m and 90°, and the fifth person positioned at 13 m and 120° with respect to the radar. Fig. E.5 shows one particular entry of the measurement setup for Set3_OnlyHumans.

The range-angle images depict the signatures of the different target objects, UAV, car, and humans. To train using the Yolov3 algorithm, we draw bounding boxes on the range-angle images using the LabelImg software [25]. The bounding boxes are drawn such that the centroid of the boxes represents the AoA of the target object.

E.6 Results

The labeled range-angle images are used to train the Yolov3 algorithm for optimum weight parameters. The training is performed for 10 folds of the experiment. In each fold of the experiment, the training is executed for 6000 iterations so that the average loss is minimized. The average loss vs iterations for the 7^{th} fold of the experiment can be observed in Fig E.6. Further, the accuracy for each fold of the experiment is listed in Table E.7. As observed and calculated from Table E.7, we obtain a relatively high average classification accuracy of 98.20%, demonstrating the reliability of our approach. The predicted classification can be observed on the range-angle image as

Table E.8: Comparing the advantages of this work with other techniques

Method	**Number of antennas**	**Targets with same range or angle**	**Target classification**
2D-ESPRIT [26]	1 Tx, 2 Rx	✗	✗
DFT-ESPRIT [27]	1 Tx, 2 Rx	✗	✗
Dual-Smoothing [28]	1 Tx, 2 Rx	✗	✗
Clustered ESPRIT [29]	1 Tx, 2 Rx or more. Rx antennas could be less than number of targets	✓	✗
Rotating Radar [13]	1 Tx, 1 Rx	✓	✗
Ours	**1 Tx, 1 Rx**	✓	✓

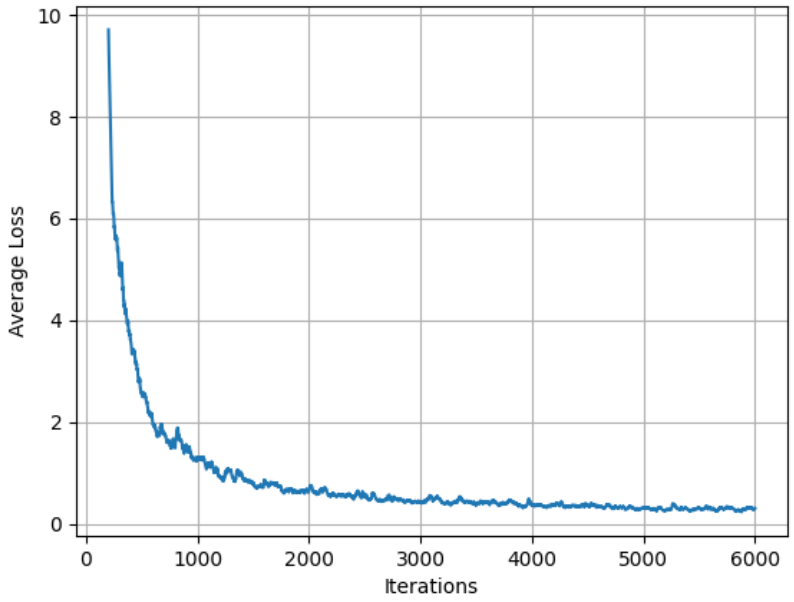
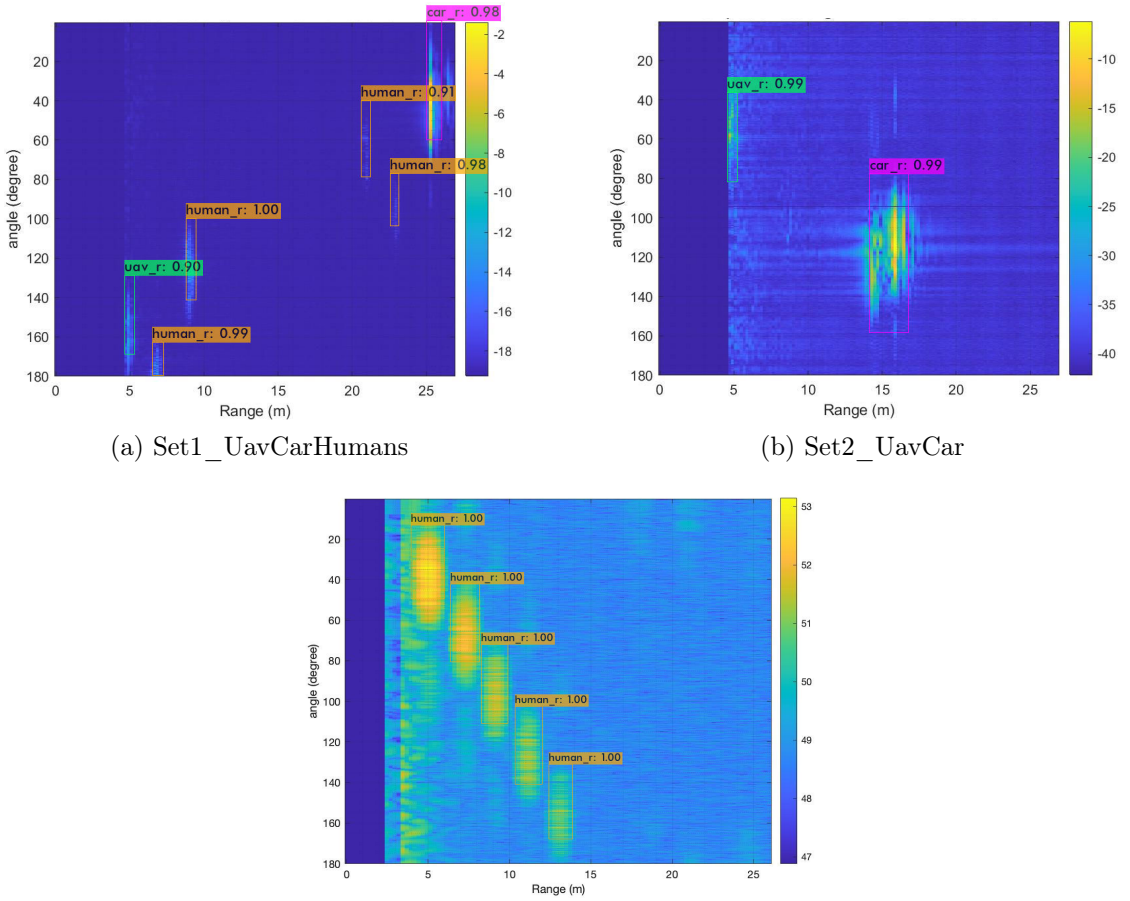


Figure E.6: Average loss vs iteration curve for the 7^{th} fold

shown in Fig. E.7. The prediction statistics can also be visualized with the help of a confusion matrix as shown in Fig E.8. The confusion matrix shows that UAV, car, and humans are classified 97.89%, 99.85%, and 99.28% respectively. The average precision, recall, and F1-score for 10 folds of the experiment are 0.991, 0.992, and 0.991 respectively.

The performance of the algorithm to estimate the AoA is calculated using the RMSE value. Table E.7 lists the different RMSE values obtained for each fold of the algorithm for different test scenarios. It can be seen that the average AoA RMSE value for UAV, car, and humans are 1.0800°, 1.2922°, and 1.0225° respectively.



(a) Set1_UavCarHumans

(b) Set2_UavCar

(c) Set3_OnlyHumans

Figure E.7: Images depicting Yolov3 class prediction for range-angle images from all three datasets.

Class	UAV	Car	Human	None	Total
UAV	650	0	0	14	664
Car	0	669	0	1	670
Human	0	0	1955	14	1969

Figure E.8: Confusion matrix

Interestingly, the average RMSE value of cars is greater than that of UAV and humans. This is expected as it aligns with the fact that the car has more AoA spread due to its large size. The proposed method implies that it is difficult to accommodate the car to a single AoA due to its large size as compared to UAV and human.

Table E.8 shows the advantages of the proposed method over other similar techniques such as 2D-ESPRIT [26], DFT-ESPRIT [27], dual-smoothing [28], clustered ESPRIT [29], and rotating radar [13] in terms of number of antennas, and target classification. It is observed that the proposed method is similar to the rotating radar technique as both utilize just a single transmitter-receiver antenna for detecting mul-

tiple non-interacting targets. However, our method provides an added advantage of target classification along with AoA estimation. Additionally, the proposed method also has a lower RMSE error as compared to the rotating radar technique.

E.7 Conclusion

The AoA estimation using mmWave FMCW radars is not accurate due to limited number of antennas. As a result, target localization is imperfect in critical applications involving UAVs and GCS. This work demonstrated that the FoV and AoA estimation of mmWave FMCW radars can be improved by combining a mechanical rotor setup with cutting-edge computer vision techniques such as Yolov3. The radar setup's range-angle images were fed into the Yolov3 algorithm, which detected, classified, and localized multiple targets in the scene. The proposed method achieved very high accuracy for target classification with low AoA estimation error outperforming traditional techniques. Evaluating the proposed approach's real-time performance on UAVs and GCSs to reduce latency in target detection and localization could be a potential future research direction.

References

[1] A. N. Wilson, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Embedded sensors, communication technologies, computing platforms and machine learning for UAVs: A review. *IEEE Sensors Journal*, 22(3):1807–1826, Feb. 2022.

[2] Tianyuan Yang, Antonio De Maio, Jibin Zheng, Tao Su, Vincenzo Carotenuto, and Augusto Aubry. An adaptive radar signal processor for uavs detection with super-resolution capabilities. *IEEE Sensors Journal*, 21(18):20778–20787, 2021.

[3] A. Aubry, Vincenzo Carotenuto, Antonio Maio, and G. Foglia. Exploiting multiple a priori spectral models for adaptive radar detection. *Radar, Sonar & Navigation, IET*, 8:695–707, 08 2014.

[4] Augusto Aubry, Vincenzo Carotenuto, Antonio De Maio, and Mark A. Govoni. Multi-snapshot spectrum sensing for cognitive radar via block-sparsity exploitation. *IEEE Transactions on Signal Processing*, 67(6):1396–1406, 2019.

[5] A. Aubry, Antonio Maio, Luca Pallotta, and Alfonso Farina. Median matrices and their application to radar training data selection. *Radar, Sonar & Navigation, IET*, 8:265–274, 04 2014.

[6] A. Aubry, A. De Maio, and V. Carotenuto. Optimality claims for the fml covariance estimator with respect to two matrix norms. *IEEE Transactions on Aerospace and Electronic Systems*, 49(3):2055–2057, 2013.

[7] Paolo Braca, Augusto Aubry, Leonardo Maria Millefiori, Antonio De Maio, and Stefano Marano. Multi-class random matrix filtering for adaptive learning. *IEEE Transactions on Signal Processing*, 68:359–373, 2020.

[8] Wenqiang Wei et al. DOA estimation of distributed mmWave radar system via fast iterative adaptive approach. In *Proc. International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 414–418, Xi'an, China, 2021.

[9] Muhammad Z. Ikram, Murtaza Ali, and Dan Wang. Joint antenna-array calibration and direction of arrival estimation for automotive radars. In *Proc. IEEE Radar Conference (RadarConf)*, pages 1–5, Philadelphia, PA, USA, 2016.

[10] Daegun Oh and Jong-Hun Lee. Low-complexity range-azimuth FMCW radar sensor using joint angle and delay estimation without SVD and EVD. *IEEE Sensors Journal*, 15(9):4799–4811, Sept. 2015.

[11] Sangdong Kim and Kyun-Kyung Lee. Low-complexity joint extrapolation-MUSIC-based 2D parameter estimator for vital FMCW radar. *IEEE Sensors Journal*, 19(6):2205–2216, March 2019.

[12] Siddharth Gupta et al. Target classification by mmWave FMCW radars using machine learning on range-angle images. *IEEE Sensors Journal*, 21(18):19993–20001, Sept. 2021.

[13] Linga Reddy Cenkeramaddi et al. A novel angle estimation for mmWave FMCW radars using machine learning. *IEEE Sensors Journal*, 21(8):9833–9843, Apr. 2021.

[14] The fundamentals of millimeter wave radar sensors. [Online]. Available: https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf.

[15] FMCW radars - Range estimation. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-1-range-estimation.

[16] FMCW radars - Angle estimation. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-5-angle-estimation.

[17] Introduction to mmwave sensing - FMCW Radars. [Online]. Available: https://training.ti.com/sites/default/files/docs/mmwaveSensing-FMCW-offlineviewing_0.pdf.

[18] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv*, 2018.

[19] Joseph Redmon. Darknet: Open source neural networks in C. [Online]. Available: http://pjreddie.com/darknet/.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv*, 2015.

[21] Khawaja Moyeezullah Ghori et al. Performance analysis of different types of machine learning classifiers for non-technical loss detection. *IEEE Access*, 8:16033–16048, Dec. 2019.

[22] Texas Instruments AWR2243. [Online]. Available: https://www.ti.com/document-viewer/AWR2243/datasheet.

[23] Texas Instruments AWR1843. [Online]. Available: https://www.ti.com/document-viewer/AWR1843/datasheet/device-overview-x3342#x3342.

[24] MATLAB. [Online]. Available: https://www.mathworks.com/products/matlab.html.

[25] LabelImg software. [Online]. Available: https://github.com/tzutalin/labelImg.

[26] DG Oh, YH Ju, and JH Lee. Subspace-based auto-paired range and DOA estimation of dual-channel FMCW radar without joint diagonalisation. *Electronics letters*, 50(18):1320–1322, Aug. 2014.

[27] Sangdong Kim, Daegun Oh, and Jonghun Lee. Joint DFT-ESPRIT estimation for TOA and DOA in vehicle FMCW radars. *IEEE Antennas and Wireless Propagation Letters*, 14:1710–1713, Apr. 2015.

[28] Daegun Oh, Yeonghwan Ju, Hyunsoo Nam, and Jong-Hun Lee. Dual smoothing DOA estimation of two-channel FMCW radar. *IEEE Transactions on Aerospace and Electronic Systems*, 52(2):904–917, Apr. 2016.

[29] Wen-Hsien Fang and Li-Der Fang. Joint angle and range estimation with signal clustering in FMCW radar. *IEEE Sensors Journal*, 20(4):1882–1892, Oct. 2019.

Appendix F

PAPER F

Title: Hybrid BLE/LTE/Wi-Fi/LoRa Switching Scheme for UAV-Assisted Wireless Networks

Authors: **W. A. N.**, Y. S. Reddy, A. Kumar, A. Jha, L. R. Cenkeramaddi

Conference: IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 2021

Hybrid BLE/LTE/Wi-Fi/LoRa Switching Scheme for UAV-Assisted Wireless Networks

Wilson A. N., Y. S. Reddy, A. Kumar, A. Jha, L. R. Cenkeramaddi

Abstract: The unmanned aerial vehicles are deployed in multiple layers to monitor an area and report the information to the ground control station. When we use a single communication protocol such as Bluetooth Low Energy (BLE)/Wi-Fi with low range, the data has to pass through multiple hops for data transfer. This in turn, increases the delay for data transmission. Even though LoRa protocol supports longer distances, the delay is more due to the limited bandwidth. Thus, in this work, we propose a hybrid BLE/LTE/Wi-Fi/LoRa switching scheme that consumes lower energy in addition to reducing the average delay in the network. The proposed scheme switches between the communication technologies based on the lower energy consumption. The performance of the proposed hybrid switching scheme is compared with the individual communication protocols in terms of both energy consumption and average delay. Through extensive numerical results, we show that the proposed hybrid switching scheme performs better in comparison to the individual communication technologies.

F.1 Introduction

Recent technological developments have brought unmanned aerial vehicles (UAVs) in the forefront for several leading applications ranging from precision agriculture [1], construction [2], mining [3], aerial photography [4], and disaster management [5]. Due to these vast applications, market experts predict that the revenue from the UAV market can exceed 8.5 billion dollars by 2027 [6]. UAVs are capable of providing enhanced services due to their inherent ability to fly and take useful measurements from the environment. The sensed information along with telemetry and other data is transmitted to other UAVs or ground control station (GCS) so as to facilitate safe and secure decision making. The transmission of this information is performed using hardware communication modules. Depending on the application, the size of data, and other factors, UAVs can be equipped with different communication modules [7].

Some of the communication modules used in UAVs are Bluetooth Low Energy (BLE) [8], Wireless Fidelity (Wi-Fi) [9], Long Term Evolution for machine-type

communication (LTE-M) [10], and Long Range (LoRa) [11]. BLE is capable of transmitting data with sufficiently low energy for short distances (around 300 m) [12, 13]. This makes them suitable for a variety of low power and short range UAV applications that includes smart agriculture [14]. In [8], the authors have proposed a scheme by which UAVs are utilized to collect the data from various sensor nodes deployed in a large farmland. In this scheme, the crop health information from the sensor nodes is sent to the nearby UAV by using BLE. In applications that demand high data rates, the Wi-Fi communication protocol is favoured. In [9], the authors have proposed a UAV system that can be used in disaster affected areas to send Voice over Internet Protocol (VoIP) information to the GCS for monitoring. Here, Wi-Fi is used as the communication protocol for sending the video information as it is reliable and offers high data rates. LTE-M on the other hand is an enhanced version of the popular LTE protocol which is designed for enabling seamless communication for Internet-of-Things (IoT) devices. An analysis of the usage of LTE-M and other Low Power Wide Area Network (LPWAN) protocols has been carried out in [10] for UAV-assisted wireless networks. Here, the performance has been evaluated in terms of latency and throughput by carrying out real world experiments. Finally, it has been concluded that LTE-M and other LPWAN protocols can be effectively used for reliable communication among high speed moving objects [10]. LoRa is one of the most commonly used communication protocol for long distance communication. LoRa stands for long range and it provides a coverage distance of around 10 km with low power consumption of 0.025 watts [15, 16]. In [11], the authors have discussed the reliability of various communication technologies such as LoRa, Wi-Fi, and LTE from a UAV swarm perspective. Further, they have developed an open source named EasySwarm in order to demonstrate the reliability of these protocols under different scenarios. It has been observed that LoRa protocol offers better reliability for long range communications with higher UAV swarm density when compared with Wi-Fi.

The usage of a single communication protocol may not be efficient to support the needs of an adaptive UAV-assisted wireless network. For example, the LoRa protocol can be used for long distance based communications. However, it fails to offer higher data rates when the UAV moves near to the GCS due to its mobility. In this case, switching to either BLE or Wi-Fi may offer higher data rates for the same distance.

Motivated by this, in [17], the authors have considered a multi-layer ad-hoc UAV network in which the BLE and Wi-Fi are used to improve the throughput and latency of the network. Initially, the UAV-assisted network is divided into multiple clusters. The authors have considered BLE for communication within a cluster and between each cluster head and gateway UAV. Further, Wi-Fi is considered for the communication between the gateway UAV and GCS. Simulations are performed in Optimized Network Engineering Tool (OPNET) and the performance of the proposed scheme and standalone communication protocols is evaluated and compared. It has been concluded that the proposed scheme outperforms the individual standalone communication protocol in terms of throughout and latency. However, this scheme has not provided sufficient emphasis on the energy consumption of the UAVs.

Moreover, this scheme has not considered other communication protocols that can facilitate long range communication between UAVs and GCS. Furthermore, the authors have considered only the free space path loss model. This may not work well for most practical scenarios wherein, there is multipath propagation.

Hence, in this paper, we aim to overcome some of these shortcomings and provide a detailed analysis of a novel hybrid switching scheme for UAV-assisted wireless networks. The key contributions of this work are given below:

- We propose a hybrid BLE/LTE/Wi-Fi/LoRa switching scheme for a multi-layer UAV-assisted network.
- The proposed scheme aid in selecting the communication technology that consumes low energy for transmitting the available data from a UAV to the GCS.
- We formulate the optimization problem in terms of total energy consumption for each communication technology for transmitting the available data from a UAV to GCS.
- We then propose an algorithm to solve the optimization problem for both free space (FS) and free space and multipath (FSMP) energy consumption models.
- Through extensive simulations, we compare the performance of the proposed scheme with individual communication protocols in terms of network energy consumption and average delay incurred.

The remaining sections of this paper is organized as follows: Section F.2 discusses the system model and the problem formulation. The proposed hybrid communication scheme is discussed in Section F.3. Section F.4 provides the definitions of the key performance metrics which are employed in this paper. The extensive numerical results are presented in Section F.5. Finally, Section G.8 concludes the paper with potential future work.

F.2 System Model and Problem Formulation

We consider a multi-layer UAV scenario as shown in Fig. F.1 where, N UAVs deployed randomly over an area of l x b m^2. Further, the UAVs randomly select the hovering height from the set $\in \{h_1, h_2, \cdots, h_m\}$ where, m is the number of altitude levels. These UAVs collect the data and transmit it to the GCS for monitoring which is situated on ground at $(l/2, b/2)$. We assume the packet arrival rate follows Poisson distribution with parameter chosen randomly [18]. After successful transmission of data to the GCS, each UAV moves to another location, in a randomly chosen direction, with a velocity of V m/s for a time interval t.

We consider four communication technologies such as BLE, Wi-Fi, LTE-M, and LoRa for communication between each UAV and GCS that exhibit their own unique attributes. Some of these attributes that are considered includes transmit power (P_T), delay (T_d), data rate (R), and path loss reference distance (d_{g_0}). We consider

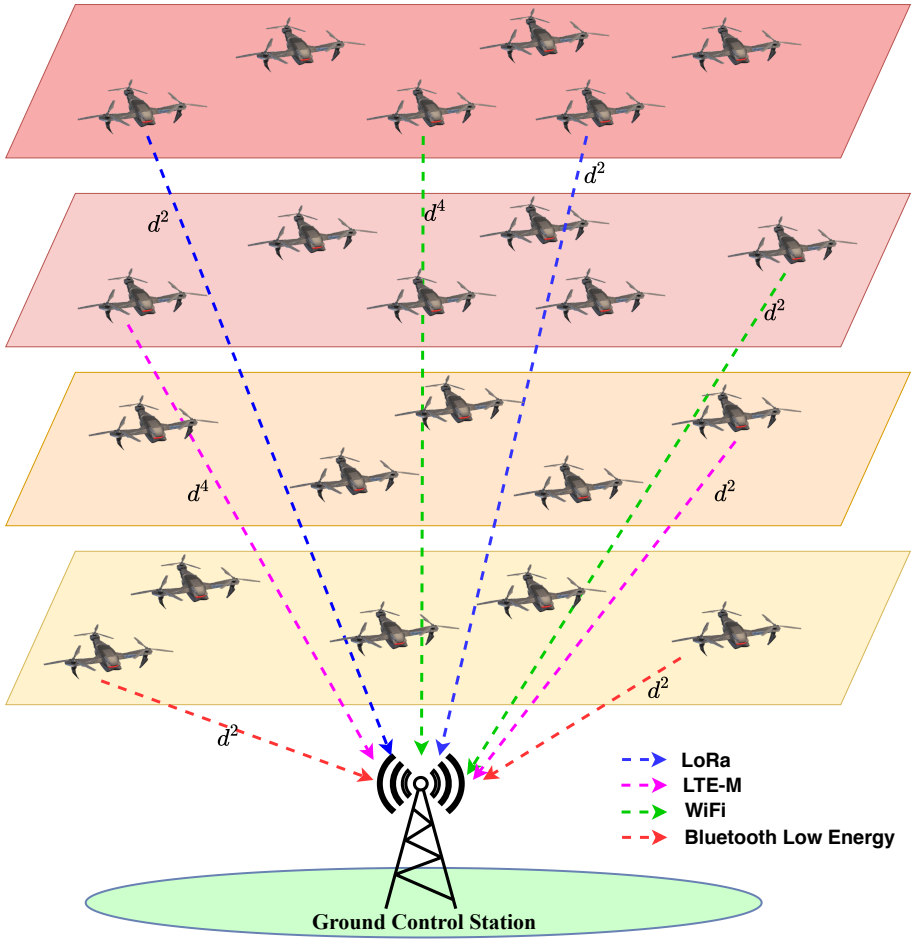


Figure F.1: System model.

FS path loss model that follows d^2 energy consumption model within a maximum range of r [19]. We also consider FSMP model wherein, the energy consumption follows d^2 model for $d_g \leq d_{g_0}$ and d^4 energy consumption model for $d_g > d_{g_0}$ [19], where d_g is the geographical distance from the UAV to the GCS.

F.2.1 Problem Formulation

In this section, we formulate the problem in terms of the total energy consumed for transmitting the data from a UAV which is located at a distance, d_g, to the GCS.

The overall energy consumption is obtained as the sum of the energy consumed for transmitting the total data and the energy consumed for transmitting this data over a distance. The energy consumed for transmitting k-bits of information is given by [20]

$$E_1 = \frac{P_T k}{R}, \tag{F.1}$$

where, P_T denotes the transmit power and R denotes the data rate. As described earlier, the distance based energy consumption model depends on the consideration of the path loss model [19]. We describe the energy consumption models for both FS model and FSMP model in the following sections.

F.2.1.1 Free space model

We consider the d^2 energy consumption model. However, this model is restricted by a maximum range for each communication technology. Thus, the amount of energy consumed for transmitting k-bits of information over a distance d_g ($d_g < r_i$) is given as [19]

$$E_2 = k\mathcal{E}_e + k\mathcal{E}_{fs}d_g^2 \tag{F.2}$$

where $\mathcal{E}_e$ represents the energy losses due to the electronic circuit per bit and $\mathcal{E}_{fs}$ is the power amplification energy in free space. Here, r_i for $i \in \{\text{Bluetooth, LTE-M, Wi-Fi, LoRa}\}$ represent the maximum range for transmitting with BLE, LTE, Wi-Fi, and LoRa communication technologies, respectively. Beyond this range, the communication protocol does not support the transmission. In this case, there will be multi-hop based communication. We consider conventional shortest path routing method for packet transmission from a UAV to the GCS.

F.2.1.2 Free space and multipath model

In this model, d^2 energy consumption model is followed for a distance $d_g \leq d_{g_0}$ (free space model). Beyond this threshold, the energy consumption follows d^4 model due to multipath. The energy consumed for transmitting k-bits of data over a distance d_g is obtained as [19]

$$E_2 = \begin{cases} k\mathcal{E}_e + k\mathcal{E}_{fs}d_g^2, & \text{when } d_g < d_{g_0}\,; \\ k\mathcal{E}_e + k\mathcal{E}_{mp}d_g^4, & \text{when } d_g \geq d_{g_0}\,. \end{cases} \tag{F.3}$$

For the simulation, we have used $\mathcal{E}_e = 25\times10^{-9}$ J/bit and $\mathcal{E}_{fs} = 10\times10^{-12}$ J/bit/m^2. The parameter $\mathcal{E}_{mp}$ refers to the power amplification energy in the multipath fading model and is given by,

$$\mathcal{E}_{mp} = \frac{\mathcal{E}_{fs}}{d_g^2} \tag{F.4}$$

Thus, the total energy consumption is given by,

$$E = E_1 + E_2 \tag{F.5}$$

Finally, the cost for transmitting the data from all UAVs to GCS is obtained as

$$C = \sum_{i=1}^{N} E_{i,GCS}\,. \tag{F.6}$$

where, $E_{i,GCS}$ is the total energy consumed for transmitting the data from i-th UAV to GCS.

F.3 Proposed Scheme

In this section, we describe the algorithm for the proposed hybrid switching scheme for minimizing the overall cost of the network described (F.4).

As the overall objective is to minimize the energy consumption of the network, each UAV should choose the communication protocol which uses minimum energy for transmitting the available data. Based on two energy models that are employed we have two different approaches:

F.3.0.1 For the case of free space model

Each UAV checks the possible communication technologies for transmission based on its distance from the GCS. Then, it calculates the energy consumed for transmitting the data. A UAV selects the communication technology that consumes less energy in comparison to other protocols. This in turn, reduces the energy consumption for transmitting the same data as described in Algorithm. 5.

F.3.0.2 For the case of free space and multipath model

As described earlier, a UAV can communicate with the GCS by using any of the available communication technologies. However, the energy consumption changes based on its distance with respect to the threshold. Thus, each UAV calculates the amount of energy consumed by each communication technology for transmitting the available data. Then, it switches to the communication technology that consumes less energy in comparison to other communication technologies as described in Algorithm. 5.

F.4 Performance Metrics

In this section, we describe the performance metrics such as average delay and network energy consumption for the evaluation of the proposed model.

F.4.1 Average Delay

It is defined as the ratio of the sum of delays for transmitting the data from all UAVs to GCS and the total number of UAVs. Total delay from a UAV to GCS is obtained as the sum of the propagation delay and transmission delay.

F.4.1.1 Transmission delay

It is the delay for transmitting a packet from one UAV to GCS. It usually depends on the data rate of the communication technology. For transmitting n packets of k-bits each over a communication channel, the incurred transmission delay is given by

$$T_{trans} = \frac{nk}{R}, \tag{F.7}$$

Algorithm 5: Proposed hybrid switching algorithm.

Input: N, d_g

Output: Average network delay and energy consumption

if *Employed energy model is free space model* **then**
  Calculate the energy for each communication technology using (F.1), (F.2) and (F.4);
  if $d_g < r_{Bluetooth}$ **then**
    Select protocol with minimum energy from the set $\{E_{Bluetooth}, E_{LTE-M}, E_{WiFi}, E_{LoRa}\}$;
  else if $d_g < r_{LTE-M}$ **then**
    Select protocol with minimum energy from the set $\{E_{LTE-M}, E_{WiFi}, E_{LoRa}\}$;
  else if $d_g < r_{WiFi}$ **then**
    Select protocol with minimum energy from the set $\{E_{WiFi}, E_{LoRa}\}$;
  else
    Select the LoRa communication protocol;
  end
else
  Employed energy model is free space and multipath model
  Calculate the energy for each communication technology using (F.1), (F.3) and (F.4);
  Select protocol consuming minimum energy from the set $\{E_{Bluetooth}, E_{LTE-M}, E_{WiFi}, E_{LoRa}\}$;
end
Obtain the average network delay and energy consumption ;

where, R is the data rate of the respective communication protocol.

F.4.1.2 Propagation delay

It is the delay incurred for propagating data from a UAV to GCS over a distance of d_g. The expression for the propagation delay is obtained as

$$T_{prop} = \frac{d_g}{c}, \tag{F.8}$$

where, c is the speed of light which is 3×10^8 m/s and d_g is the distance of the UAV from the GCS. From (F.7) and (F.8), the total delay is obtained as

$$T_d = T_{trans} + T_{prop}\,, \tag{F.9}$$

Finally, the expression for average delay of the network is given as

$$T_{avg} = \frac{1}{N}\sum_{i=1}^{N} T_{d_i}\,. \tag{F.10}$$

Table F.1: Various communication protocol attributes

Protocol	**Transmit Power (W)**	**Data rate (bits/second)**	**Path loss reference distance (Normalized)** d_{g_0}
Bluetooth [21]	0.01	1360 x 10^3	200
LTE-M [22], [23]	0.1	1 x 10^6	400
WiFi [24], [15]	2	1 x 10^7	600
LoRa [15], [16]	0.025	50 x 10^3	1500

F.4.2 Network Energy Consumption

The network energy is defined as the sum of the energies consumed for transmitting the data from each of the N UAVs to GCS. The expression for network energy consumption is given by

$$E_{total} = \sum_{i=1}^{N} E_{i,GCS}\,. \tag{F.11}$$

F.4.3 Packet Arrival Rate

We assume the packet arrival rate follows Poisson distributed random variable which is defined as

$$Pr(X = n) = \frac{\lambda^n e^{-\lambda}}{n!} \tag{F.12}$$

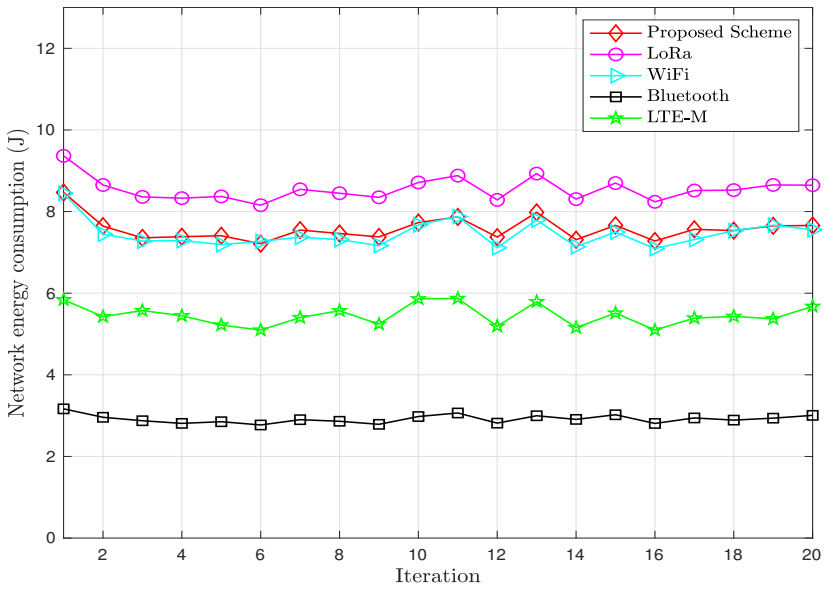
F.5 Numerical results

In this section, we provide the simulation comparison of the proposed scheme with the other protocols in terms of average delay and energy consumption.

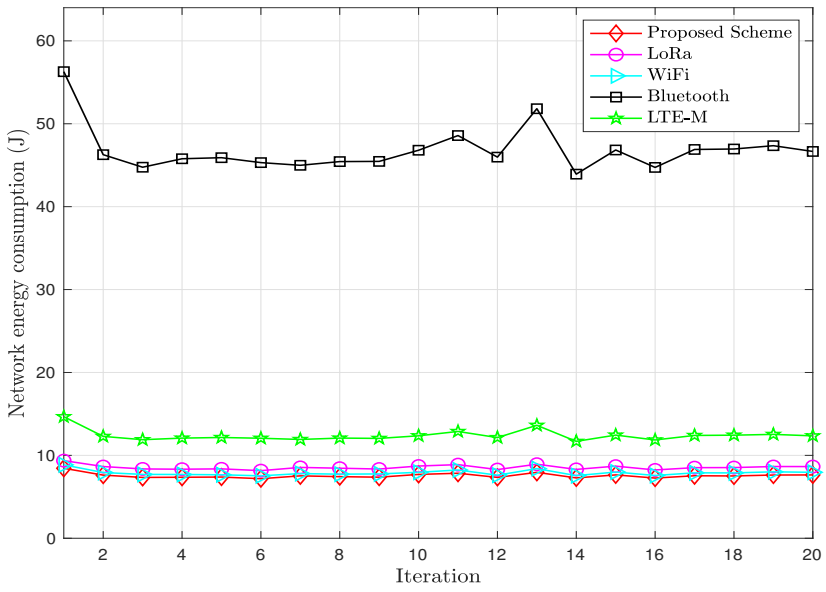
F.5.1 Experimental setup

We consider a scenario of $N = 500$ UAVs that are deployed randomly over a 1000 x 1000 m^2 area and a random hovering height h selected from the set $\{100, 200, 300, 400\}$. After each transmission, a UAV transits with a speed of 5 m/s for a duration of 60 seconds in a randomly chosen direction within the given area. Further, GCS is located at the center of the deployed area with coordinates $(50, 50, 0)$.

The UAVs are equipped with communication modules such as BLE, Wi-Fi, LTE-M, and LoRa. We have considered the typical values of different attributes for each communication technology as given in Table F.1. We consider that the packet arrival rate follows Poisson distribution with parameter chosen randomly from the set $\{1, 2, \cdots, 100\}$ and each packet is of 128 bits.



(a)



(b)

Figure F.2: (a) Total network energy consumption for free space energy model and (b) total network energy consumption for free space and multipath energy model.

F.5.2 Results

Fig. F.2a shows the variation of network energy consumption in each scenario with free space model. From Fig. F.2a, it is observed that the energy consumption of the proposed hybrid scheme is comparable to the energy consumption model when Wi-Fi is used alone for transmission. However, the proposed model provides better energy consumption than a standalone LoRa based scheme. The amount of energy consumed by stand alone BLE is less compared to the hybrid model due to the lower transmit power.

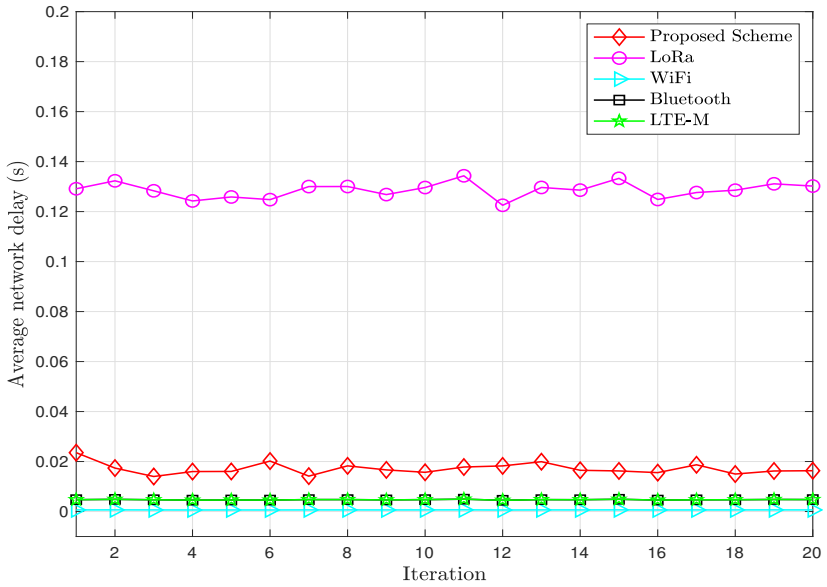
Fig. F.2b shows the simulation comparison of the proposed model with the individual communication technologies in terms of network energy consumption for FSMP model. From Fig. F.2b, it is observed that the hybrid scheme consumes less energy in comparison to the individual communication protocols due to the switching technique. In case of BLE, the energy consumption is more as most of the UAVs which are farther from threshold follows d^4 energy consumption model.

Figs. F.3a and F.3b show the variation of the delay for FS and FSMP models, respectively. The delay depends upon number of bits to be transmitted and the distance over which the transmission occurs. Hence, it can be observed that the delay doesn't show significant variation for the proposed hybrid switching technique even with both energy models. In case of standalone communication protocols, the delay depends primarily on the total number of bits to be transmitted. Here, the overall delay depends mostly on the transmission delay as the propagation delay does not contribute much. Hence, it is concluded that the FSMP energy model for the hybrid switching communication scheme outperforms existing standalone communication technologies by providing minimum energy consumption and reduced average network delay as can be observed from Figs. F.2b and F.3b, respectively.

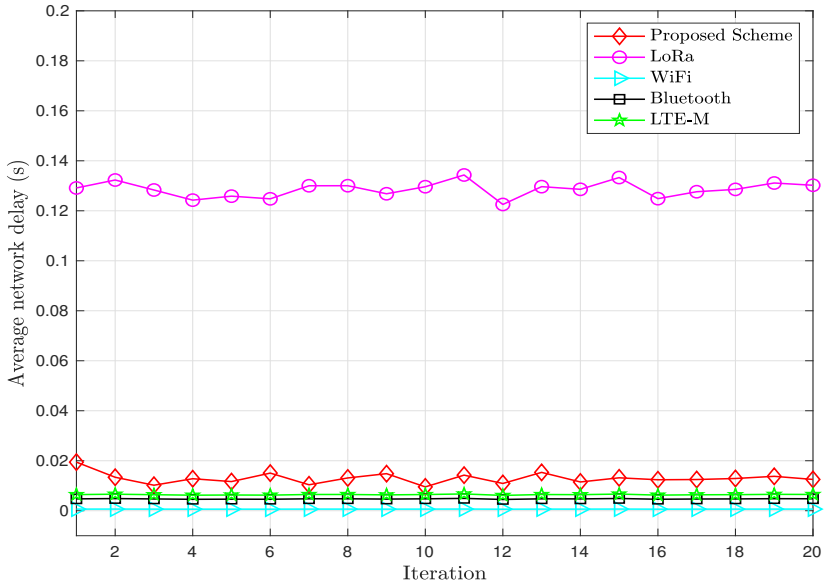
Figs. G.8a and G.8b show the simulation comparison in terms of energy consumption and average delay, respectively, for the hybrid scheme with FS and FSMP models. From Fig. G.8a, it is observed that the network energy consumption for both the models are similar with the FSMP model showing lower energy consumption. However, the average delay is varies greatly as can be observed from Fig. G.8b. This is due to the fact that the number of UAVs that are connected to BLE, Wi-Fi, LTE-M, and LoRa are 14, 279, 153, and 54, respectively, in the case of FS model. However, in case of FSMP model, the number of UAVs connected to BLE has increased to 21 and the number UAVs connected to LTE-M has increased to 162. Further, the number of UAVs connected to LoRa has reduced to 38. Since both BLE and LTE-M offers higher data rates in comparison to LoRa, the overall delay is less for FSMP model relative to FS model.

F.6 Conclusion

In this paper, we have proposed a hybrid BLE/Wi-Fi/LTE/LoRa switching scheme for UAV-assisted wireless networks. In the proposed scheme, each UAV switches the communication protocol based on the lower energy consumption for transmitting



(a)



(b)

Figure F.3: (a) Average network delay for free space energy model and (b) average network delay for free space and multipath energy model.

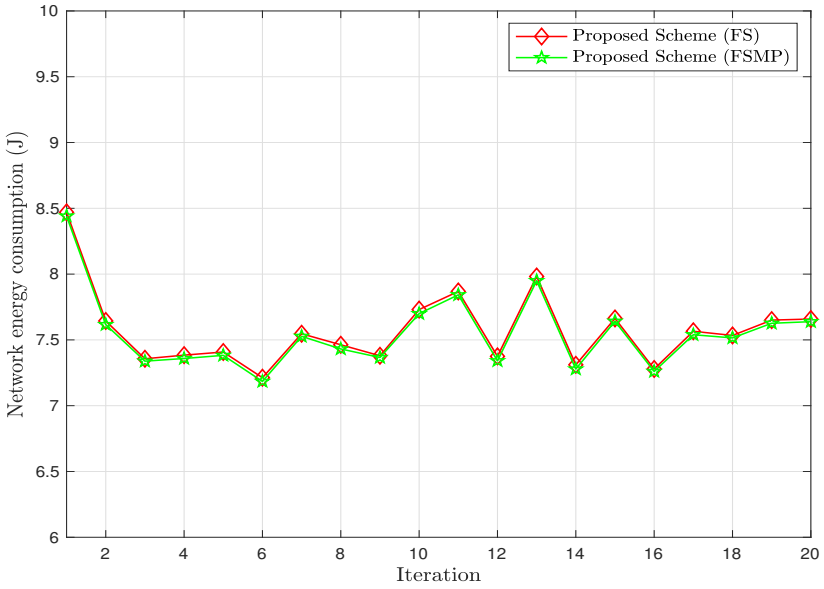


Figure F.4: Total network energy comparison with proposed method for free space and free space and multipath energy models.

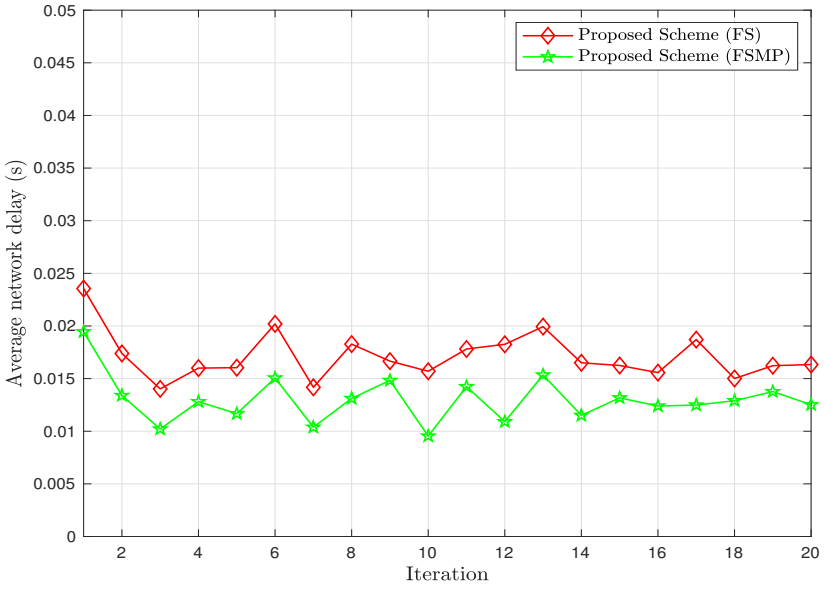


Figure F.5: Average network delay comparison with proposed method for free space and free space and multipath energy models.

the available data. Further, the performance of the proposed scheme has been evaluated for both free space (FS) and free space and multipath (FSMP) models. We have evaluated the performance of the proposed approach against the standalone communication protocols in terms of network energy consumption and average delay. It has been concluded that the proposed hybrid scheme outperformed other protocols for the FSMP model. Further, the proposed scheme performs well with FSMP model as compared to FS model. In future, we will analyze the performance of the proposed scheme in terms of additional parameters such as network lifetime and throughput. Additionally, the analytical expressions corresponding to all the performance metrics will be derived.

References

[1] Panagiotis Katsigiannis, Lazaros Misopolinos, Vasilis Liakopoulos, Thomas K. Alexandridis, and George Zalidis. An autonomous multi-sensor uav system for reduced-input precision agriculture applications. In *Proc. Mediterranean Conference on Control and Automation (MED)*, pages 60–64, Athens, Greece, Jun. 2016.

[2] Youngjib Ham, Kevin K. Han, Jacob J. Lin, and Mani Golparvar-Fard. Visual monitoring of civil infrastructure systems via camera-equipped Unmanned Aerial Vehicles (UAVs): a review of related works. *Visualization in Engineering*, 4(1):1, January 2016.

[3] He Ren, Yanling Zhao, Wu Xiao, and Zhenqi Hu. A review of UAV monitoring in mining areas: current status and future perspectives. *International Journal of Coal Science & Technology*, 6(3):320–333, September 2019.

[4] Petra Urbanová, Mikoláš Jurda, Tomáš Vojtíšek, and Jan Krajsa. Using drone-mounted cameras for on-site body documentation: 3D mapping and active survey. *Forensic Science International*, 281:52–62, Oct. 2017.

[5] Waleed Ejaz, Arslan Ahmed, Aliza Mushtaq, and Mohamed Ibnkahla. Energy-efficient task scheduling and physiological assessment in disaster management using UAV-assisted networks. *Computer Communications*, 155:150–157, Apr. 2020.

[6] Drone market outlook. [Online]. Available: https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts.

[7] G. Pantelimon, K. Tepe, R. Carriveau, and S. Ahmed. Survey of Multi-agent Communication Strategies for Information Exchange and Mission Control of Drone Deployments. *Journal of Intelligent & Robotic Systems*, 95(3):779–788, September 2019.

[8] Shota Nishiura and Hiroshi Yamamoto. Large-term sensing system for agriculture utilizing uav and wireless power transfer. In *Proc. International Conference on Information Networking (ICOIN)*, pages 609–614, Jeju Island, South Korea, Jan. 2021.

[9] Vicente Mayor, Rafael Estepa, Antonio Estepa, and German Madinabeitia. Deploying a Reliable UAV-Aided Communication Service in Disaster Areas. *Wireless Communications and Mobile Computing*, 2019:e7521513, April 2019.

[10] Shie-Yuan Wang, Jui-En Chang, Hsin Fan, and Yi-Hsiu Sun. Performance comparisons of nb-iot, lte cat-m1, sigfox, and lora moving at high speeds in the air. In *Proc. IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, Rennes, France, Jul. 2020.

[11] Zhenhui Yuan, Jie Jin, Lingling Sun, Kwan-Wu Chin, and Gabriel-Miro Muntean. Ultra-reliable iot communications with uavs: A swarm use case. *IEEE Communications Magazine*, 56(12):90–96, Dec. 2018.

[12] Bluetooth 5: Go Faster. Go Further. [Online]. Available: https://www.bluetooth.com/wp-content/uploads/2019/03/Bluetooth_5-FINAL.pdf.

[13] Nian Xia, Hsiao-Hwa Chen, and Chu-Sing Yang. Emerging technologies for machine-type communication networks. *IEEE Network*, 34(1):214–222, Feb. 2020.

[14] Praveen Kumar Reddy Maddikunta, Saqib Hakak, Mamoun Alazab, Sweta Bhattacharya, Thippa Reddy Gadekallu, Wazir Zada Khan, and Quoc-Viet Pham. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sensors Journal*, pages 1–1, early access, 06 Jan. 2021.

[15] Pycom OEM module. [Online]. Available: https://pycom.io/product/l04-oem-module/.

[16] Semtech SX1276 - Datasheet. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276.

[17] Muhammad Asghar Khan, Ijaz Mansoor Qureshi, and Fahimullah Khanzada. A hybrid communication scheme for efficient and low-cost deployment of future flying ad-hoc network (fanet). *Drones*, 3(1), Feb. 2019.

[18] Om Jee Pandey and Rajesh M. Hegde. Low-latency and energy-balanced data transmission over cognitive small world wsn. *IEEE Transactions on Vehicular Technology*, 67(8):7719–7733, Mays 2018.

[19] S. M. Mahdi H. Daneshvar, Pardis Alikhah Ahari Mohajer, and Sayyed Majid Mazinani. Energy-efficient routing in wsn: A centralized cluster-based approach via grey wolf optimizer. *IEEE Access*, 7:170019–170031, Nov. 2019.

[20] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[21] Bluetooth, Bluetooth Technology. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/.

[22] 3gpp Standards for the IoT. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1805-iot_r14.

[23] 3gpp Release 13 Overview. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1628-rel13.

[24] Evgeny Khorov, Anton Kiryanov, Andrey Lyakhov, and Giuseppe Bianchi. A tutorial on ieee 802.11ax high efficiency wlans. *IEEE Communications Surveys Tutorials*, 21(1):197–216, Apr. 2019.

Appendix G

PAPER G

Title: RL-based Energy-Efficient Data Transmission over Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network

Authors: **W. A. Nelson**, S. R. Yeduri, A. Jha, A. Kumar, L. R. Cenkeramaddi

Journal: IEEE/ACM Transactions on Networking

RL-based Energy-Efficient Data Transmission over Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network

W. A. Nelson, S. R. Yeduri, A. Jha, A. Kumar, L. R. Cenkeramaddi

Abstract: **The lifetime of a UAV-assisted wireless network is determined by the amount of energy consumed by the UAVs during flight, data collection, and transmission to the ground station. Routing protocols are commonly used for data transmission in a communication network. However, because of the mobility of UAVs, using a routing protocol with a single communication technology results in higher delay and more energy consumption in a UAV-assisted wireless network. To overcome this, we propose two reinforcement learning (RL) algorithms, Q-learning and deep Q-network (DQN), for energy-efficient data transmission over a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. We consider BLE, LTE, Wi-Fi, and LoRa for communication over a UAV-GS link. The RL algorithms take any random network as input and learn the best policy to output the network with less energy consumption. The reward/penalty is chosen in such a way that the network with the highest energy consumption is penalized and the one with the lowest is rewarded, thereby minimizing total network energy consumption. Based on learning, it creates a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network by assigning the best communication technology to a UAV-GS link. Further, we compare the performance of proposed RL algorithms with a rule-based algorithm and random hybrid scheme. In addition, we propose a theoretical framework for constructing hybrid network for both free space and free space multipath path loss models. We demonstrate the performance comparison of the proposed work with the conventional shortest path routing algorithm in terms of network energy consumption and average network delay using extensive results. Finally, the effect of the velocity of the UAV and the number of packets on the performance of the proposed framework is analyzed.**

G.1 Introduction

Technological advances and ease of regulations in today's world have helped unmanned aerial vehicle (UAV) technology to serve the ever-growing demand of modern applications. UAVs are used in a number of applications that include precision agriculture [1], space exploration [2], providing connectivity to terrestrial networks [3], and public safety [4]. Their ability to maneuver and access remote locations has made them suitable for remote sensing and disaster management applications [5]. UAV swarms with high precision control have recently paved the way for UAV light-shows in various events thereby reducing pollution due to explosives [6].

To ensure safe and secure UAV operations, UAV communication plays a vital role. UAVs are equipped with various communication modules such as Bluetooth Low Energy (BLE), Long Term Evolution (LTE), Wireless Fidelity (Wi-Fi), and Long Range (LoRa). These communication technologies ensure that UAVs are transmitting the required control and telemetry data to the ground control stations (GS). UAV health and trajectory are monitored by continuously sending signals to the GS and if required, to other UAVs.

Various works have been proposed in this regard. A dual radio internet-of-things (IoT) architecture has been proposed in [7] for the application of wildlife monitoring systems. The proposed approach in [7] leverages BLE in low power wide area network (LPWAN) based on the proximity among the wildlife animal herd. Finally, an analytical model for energy consumption has been presented to evaluate the performance of the proposed approach [7]. It has been shown that the proposed dual radio network improves energy efficiency when compared to a network utilizing LPWAN alone [7]. In [8], the authors have proposed an indoor hybrid RF/PLC/VLC communication system to switch the device connections among the RF, PLC, and VLC in order to improve the sum rate capacity. Further, the transmit power minimization problem has been formulated and analyzed in [9]. However, no literature is presented focusing on the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network for energy-efficient data transmission.

Further, many works have been presented in the literature focusing on the application of reinforcement learning (RL) in UAV-assisted wireless networks. A deep learning (DL) algorithm has been proposed in [10] for the collection of data in a UAV-assisted wireless network. In [11], several DL-based artificial intelligence (AI) methods such as point learning, multi-agent deep deterministic policy gradient, and federated DL have been proposed to solve the optimization problem of energy efficiency in a UAV-assisted wireless network. In [12], a combination of echo state learning and RL is used to solve the joint flight control and spectrum access problem in TeraHertz-band for UAV-assisted wireless networks. A deep reinforcement learning (DRL) approach has been proposed in [13] for resource allocation in terms of throughput, bandwidth, and power consumption in UAV-assisted wireless networks. In [14], the authors utilize the stochastic learning automata (SLA) algorithm to perform joint optimization of channel and relay selection in UAV-aided device-to-device (D2D) networks. However, none of these works focused on utilizing RL al-

gorithms for energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. Motivated by this, we propose two RL algorithms for energy-efficient data transmission over a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. The main contributions of this paper are:

- A hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network is formed with free space (FS) and free space multipath (FSMP) energy models.
- Analytical expressions corresponding to network energy consumption and average network delay are derived for the data transmission over the hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network.
- Further, two RL algorithms namely, Q-learning and deep Q-network (DQN) are proposed for energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. The performance of the proposed RL algorithms is compared with the rule-based algorithm and random hybrid scheme.
- Through extensive numerical results, we show that the proposed RL algorithms result in energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. We also compare the performance of the proposed hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network in terms of the network energy consumption and average network delay with conventional shortest path routing algorithm considering individual communication technology.

The remaining sections of this article are structured as follows: Section G.2 discusses the system model and problem formulation. The proposed theoretical framework for energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network is presented in Section G.3. The analytical expressions corresponding to network energy consumption and average network delay for FS and FSMP models are derived in Section G.4. Section G.5 describes the proposed RL algorithms for energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. Evaluation metrics are described in Section G.6 and the numerical results are discussed in Section G.7. Finally, Section G.8 concludes the paper by providing a summary and possible future research directions.

G.2 System model and problem formulation

We consider a UAV-assisted wireless network wherein, N UAVs are deployed randomly over B^2 area over l layers of height $h_1, h_2, \cdots, h_l$, respectively, as shown in Fig. G.1. These UAVs collect the data and send it to the GS which is situated on the ground at $(B/3, B/3)$. We consider GS to have access to all communication technologies to collect the data and process it. Here, the data arrival at each UAV is assumed to follow the Poisson process. After a successful transmission of data from one location, the UAV moves to another location randomly to collect the data.

Table G.1: Summary of important parameters used in this work.

Notation	Definition
C	Cost of the network in terms of energy consumption
N	Number of UAVs present in the network
h_i	Height of the i^{th} layer
l	Total number of layers
B^2	Area of each layer over which UAVs are deployed
P_T	Transmit power
R	Data rate over the transmission link
d_g	Radial distance
k	Number of bits
$\mathcal{E}_e$	Energy consumed by the electronic circuit for transmitting one bit
$\mathcal{E}_{fs}$	Free space power amplification energy
$\mathcal{E}_{mp}$	Power amplification energy in the multipath fading model
E_{Tr}	Energy consumed for transmitting k bits of data
E_d	Energy consumed for transmitting k bits of data over d_g distance
τ	Communication technology
r_τ	Range of the communication technology
λ	Learning rate for the proposed RL algorithms
γ	Reward parameter for the proposed RL algorithms
Δ	Discount parameter for the proposed RL algorithms
T_{Tr}	Transmission delay
T_{prop}	Propagation delay
A_τ	Area occupied by the communication technology
P_{pr}	Probability that a UAV present A_τ
W_1	Weighing parameter corresponding to E_{Tr}
W_2	Weighing parameter corresponding to E_d
$E_{N/W}$	Total energy consumed by the network
w	Weights of the policy neural network
$\hat{w}$	Weights of the target neural network
ζ	Interval after which target network weights are updated (in terms of number of steps)
ϕ	Denotes if terminal state is reached

A UAV-GS link is considered to select one of the four communication technologies from BLE, LTE, Wi-Fi, and LoRa. Once a communication link is established to a UAV-GS link, the data will be transmitted with a transmit power of P_T and a data rate of R. This will incur a delay of T_d seconds. Further, two path loss models

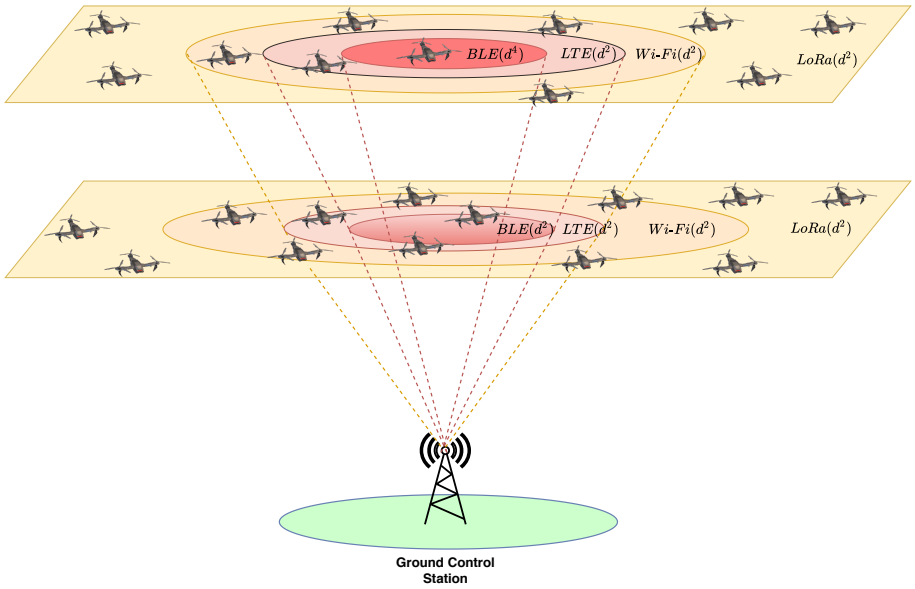


Figure G.1: System model for the hybrid BLE/LTE/Wi-Fi/LoRa scheme for two layers.

such as FS and FSMP are considered for inserting a communication technology for a UAV-GS link. In the FS model, a UAV-GS link can select one of the communication technologies, if the radial distance of the link is less than r_τ where τ represents one of the four communication technologies BLE, LTE, Wi-Fi, or LoRa. In the FS model, the energy consumption follows d^2 model. In the FSMP model, a UAV-GS link can select any of the communication links irrespective of its distance. However, the energy consumption follows d^2 model if the radial distance is less than r_τ, else, the energy consumption follows d^4 model.

G.2.1 Problem Formulation

In this work, our aim is to minimize the network energy consumption. Thus, we first obtain the overall network energy consumption. The total network energy consumption is obtained as the sum of the energy consumption at each UAV-GS link. The energy consumption at a UAV-GS link is the sum of the energy consumption for transmitting the data with a transmit power and the energy consumed for transmitting this data over a radial distance d_g. With P_T being the transmit power and R as the data rate, the energy consumed for transmitting k bits of data is obtained as [15]

$$E_{Tr} = \frac{kP_T}{R}. \tag{G.1}$$

As mentioned earlier, the energy consumption depends on the path loss model and the radial distance. Next, we derive the energy consumption for the FS model and FSMP model.

G.2.1.1 FS Model

In the FS model, a link can be connected to a communication technology in case the radial distance is less than the threshold distance, r_τ, of that communication technology. With d_g being the radial distance, the energy consumed for transmission of data over a distance d_g is obtained as [16]

$$E_{d,fs} = k\mathcal{E}_e + k\mathcal{E}_{fs}d_g^2, \tag{G.2}$$

where $\mathcal{E}_e$ is the energy consumed by the electronic circuit for transmitting one bit and $\mathcal{E}_{fs}$ denotes the free space power amplification energy.

G.2.1.2 FSMP Model

In the FSMP model, a UAV-GS link will be assigned with any of the communication technologies irrespective of its radial distance. The energy consumed for transmitting k bits of data over a radial distance of d_g is obtained as [16]

$$E_{d,fsmp} = \begin{cases} k\mathcal{E}_e + k\mathcal{E}_{fs}d_g^2, & \text{when } d_g < r_\tau\,; \\ k\mathcal{E}_e + k\mathcal{E}_{mp}d_g^4, & \text{when } d_g \geq r_\tau\,. \end{cases} \tag{G.3}$$

Here, $\mathcal{E}_{mp} = \mathcal{E}_{fs}/r_\tau^2$ denotes the power amplification energy in the multipath fading model. We consider $\mathcal{E}_e = 25 \times 10^{-9}$ J/bit and $\mathcal{E}_{fs} = 10 \times 10^{-12}$ J/bit/m^2 for the simulation evaluation [17].

Finally, the total cost of the UAV-assisted wireless network in terms of energy consumption is obtained as

$$C = \sum_{i=1}^{N}(E_{Tr,i} + E_{d,i}). \tag{G.4}$$

The aim of this work is to minimize the cost, defined in (G.4), of the network in terms of energy consumption.

G.3 Energy-Efficient Data Transmission over Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network Formation

The proposed scheme aims at the creation of a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network for energy-efficient data transmission. We consider two path loss models for generating the network which are described below.

G.3.1 FS model

In the FS model, a UAV-GS link can select a communication technology in case the radial distance is less than the threshold distance r_τ.

Algorithm 6: Algorithm to obtain the FS model-based Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network.

Input: N, d_g

Output: Network energy consumption and average network delay

if $d_g < r_{BLE}$ **then**
  Calculate the energy consumed over a UAV-GS link using (G.1) and (G.2) for all communication protocols;
  $E_{\min} = \min\{E_{BLE}, E_{LTE}, E_{Wi\text{-}Fi}, E_{LoRa}\}$;
  Choose the communication protocol with $E_{\min}$;
else if $d_g < r_{LTE}$ **then**
  Calculate the energy consumed over a UAV-GS link using (G.1) and (G.2) for LTE, Wi-Fi, and LoRa;
  $E_{\min} = \min\{E_{LTE}, E_{Wi\text{-}Fi}, E_{LoRa}\}$;
  Choose the communication protocol with $E_{\min}$;
else if $d_g < r_{Wi\text{-}Fi}$ **then**
  Calculate the energy consumed over a UAV-GS link using (G.1) and (G.2) for Wi-Fi and LoRa;
  $E_{\min} = \min\{E_{Wi\text{-}Fi}, E_{LoRa}\}$;
  Choose the communication protocol with $E_{\min}$;
else
  Choose the LoRa communication protocol;
end
Obtain network energy consumption and average network delay

Table G.2: Transmit power, data rate, and the path loss reference distance for all the communication protocols considered in this work.

Protocol	$\boldsymbol{P_T}$ **(W)**	$\boldsymbol{R}$ **(Mbps)**	$\boldsymbol{P_T/R}$	$\boldsymbol{r_\tau}$
BLE [18], [19]	0.01	1.36	0.007×10^{-6}	200
LTE [20], [21]	0.1	1	0.1×10^{-6}	400
Wi-Fi [22], [23]	2	10	0.2×10^{-6}	600
LoRa [23], [24], [25]	0.025	0.050	0.5×10^{-6}	1500

Algorithm 6 describes the steps involved in the selection of a communication protocol for a UAV-GS link. From Table G.2, it is observed that the threshold distance of BLE is less than that of LTE. This will be followed by Wi-Fi and LoRa. When the radial distance of a link is less than r_{BLE}, the link will be assigned to one of the four communication technologies that consume less energy. When the radial distance is less than r_{LTE}, the link will be assigned to one of the communication technologies from LTE, Wi-Fi, and LoRa based on minimum energy consumption. When the radial distance is less than $r_{Wi\text{-}Fi}$, the link will be assigned with one of the communication technologies from Wi-Fi and LoRa based on minimum energy consumption. Else, it will be connected to LoRa.

Algorithm 7: Algorithm to obtain the FSMP model-based Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network.

Input: N, d_g

Output: Network energy consumption and average network delay

1 Calculate the energy consumed over a UAV-GS link using (G.1) and (G.3) for all communication protocols

2 $E_{min} = \min\{E_{BLE}, E_{LTE}, E_{Wi\text{-}Fi}, E_{LoRa}\}$

3 Choose the communication protocol with $E_{\min}$

4 Obtain network energy consumption and average network delay

G.3.2 FSMP model

In the FSMP model, a link will be assigned with one of the communication links based on the energy consumption irrespective of its radial distance.

Algorithm 7 describes the procedure for the creation of hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network to evaluate the energy consumed over a link for different communication technologies. Assign the communication technology that consumes less energy for transmission.

G.4 Analytical model for Energy-Efficient Data Transmission over Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network

In this section, we derive the mathematical expressions for the network energy consumption for both FS and FSMP models.

G.4.1 FS Model

In the FS model, a link can be assigned to a communication technology in case the radial distance is less than the threshold distance. This implies that the energy consumption follows d^2 model. Thus, the connection to a communication protocol completely depends on the E_{Tr} as the E_d is the same for all the communication protocols. From Table G.2, it can be observed that the P_T/R ratio is less for BLE, followed by LTE, Wi-Fi, and LoRa. Thus, the number of UAV-GS links connected to any of the communication technologies depends on the presence of the UAV in its communication range. For example, if a UAV is present in the BLE range, it will be connected to BLE. In case the UAV is present in the LTE range and not in the BLE range, then it will be connected to LTE. The same applies for Wi-Fi. In case the UAV is not present in the Wi-Fi range, it will be connected to LoRa. Thus, each communication technology occupies a circular range for its connection. Thus, the circular area over which each communication technology occupies in a layer of

height h_i is obtained as

$$A_\tau = \begin{cases} \pi x_{BLE,fs}^2, & \text{for BLE} \\ \pi x_{LTE,fs}^2 - \pi r_{BLE,fs}^2, & \text{for LTE} \\ \pi x_{Wi\text{-}Fi,fs}^2 - \pi r_{LTE,fs}^2, & \text{for Wi-Fi} \\ B^2 - \pi x_{Wi\text{-}Fi,fs}^2, & \text{for LoRa} \end{cases} \tag{G.5}$$

where, τ represents the different communication technologies and x_τ is given as

$$x_\tau = \begin{cases} \sqrt{r_\tau^2 - h_i^2}, & \text{if } r_\tau > h_i; \\ 0, & \text{otherwise,} \end{cases} \tag{G.6}$$

Let P_{pr} represent the probability that a UAV is present in the area corresponding to a communication technology which is obtained as

$$P_{pr,\tau} = \frac{A_\tau}{B^2}. \tag{G.7}$$

Thus, the number of UAV-GS links connected to a communication technology is obtained as

$$n_\tau = P_{pr,\tau} N \tag{G.8}$$

The energy consumed by communication technology is obtained as

$$E_\tau = P_{pr,\tau} N (E_{Tr} + E_{d,fs,avg}) \tag{G.9}$$

Finally, the total energy consumed by the network is obtained as

$$E_{N/W} = \sum_\tau P_{pr,\tau} N (E_{Tr,\tau} + E_{d,fs,\tau,avg}) \tag{G.10}$$

Here, $E_{d,fs,\tau,avg}$ represents the average energy consumption incurred by the communication technology τ. The average energy $E_{d,fs,\tau,avg}$ over an area D can be obtained as [26]

$$E_{d,fs,\tau,avg} = \frac{1}{D} \iint_D f(r,\theta)\, r\, dr\, d\theta \tag{G.11}$$

Evaluating the above integral over the region D defined as the area between two concentric circles with radius r_a and r_b ($r_b > r_a$) gives

$$E_{d,fs,\tau,avg} = \frac{1}{\pi(r_b^2 - r_a^2)} \int_0^{2\pi} \int_{r_a}^{r_b} (k\mathcal{E}_e + k\mathcal{E}_{fs} r^2) r\, dr\, d\theta \tag{G.12}$$

Upon solving, we obtain the generalized closed form expression for $E_{d,fs,\tau,avg}$ as

$$E_{d,fs,\tau,avg} = k\mathcal{E}_e + \frac{k\mathcal{E}_{fs}}{2}(r_a^2 + r_b^2) \tag{G.13}$$

where r_a and r_b are defined as

$$r_a = \begin{cases} 0, & \text{for BLE} \\ r_{BLE,fs}, & \text{for LTE} \\ r_{LTE,fs}, & \text{for Wi-Fi} \\ r_{Wi\text{-}Fi,fs}, & \text{for LoRa} \end{cases} \tag{G.14}$$

and

$$r_b = \begin{cases} r_{BLE,fs}, & \text{for BLE} \\ r_{LTE,fs}, & \text{for LTE} \\ r_{Wi\text{-}Fi,fs}, & \text{for Wi-Fi} \\ r_{LoRa,fs}, & \text{for LoRa} \end{cases} \tag{G.15}$$

Similarly the average delay of the network $T_{avg,fs}$ can be derived as,

$$T_{avg,fs} = \frac{1}{\pi(r_b^2 - r_a^2)} \int_0^{2\pi} \int_{r_a}^{r_b} (T_{trans} + T_{prop}) r \, dr \, d\theta \tag{G.16}$$

where the expressions for T_{trans} and T_{prop} are detailed in Section G.6. On solving, we obtain the expression for the average network delay $T_{avg,fs}$ as,

$$T_{avg,fs} = \frac{k}{R} + \frac{2(r_a^2 + r_b^2 + r_a r_b)}{3c(r_a + r_b)} \tag{G.17}$$

where r_a and r_b are defined as above. The parameter c is the speed of light in vaccum which is equal to 3×10^8 m/s.

G.4.2 FSMP Model

In contrast to the FS model, a link can be assigned with any of the communication technology irrespective of its radial distance. Thus, the connection to a communication technology depends on both E_{Tr} and E_d. As described in the previous section, the link will be connected to a communication protocol if the UAV is present in the range of that communication protocol. Now we need to obtain the range beyond which a communication protocol extends for energy-efficient data transmission over hybrid UAV-assisted wireless network. For example, the BLE can be assigned to a link even if the radial distance is greater than the threshold distance. The requirement is that the total energy consumed with d^4 model of BLE should be less than the total energy consumed with d^2 model of LTE which is defined as

$$E_{Tr,BLE} + E_{d,fsmp,BLE} < E_{Tr,LTE} + E_{d,fs,LTE} \tag{G.18}$$

We need to derive the range of BLE, r_{BLE} by considering equality in (G.18).

$$\frac{kP_{T,BLE}}{R_{BLE}} + k\mathcal{E}_e + k\mathcal{E}_{mp} d_g^4 = \frac{kP_{T,LTE}}{R_{LTE}} + k\mathcal{E}_e + k\mathcal{E}_{fs} d_g^2 \tag{G.19}$$

Solving the above equation, we get the threshold distance corresponding to BLE which is defined as

$$r_{BLE} = \sqrt{\frac{\mathcal{E}_{fs} \pm \sqrt{\mathcal{E}_{fs}^2 - 4(\frac{P_{T,BLE}}{R_{BLE}} - \frac{P_{T,LTE}}{R_{LTE}})}}{2\mathcal{E}_{mp}}} \tag{G.20}$$

Similarly, we can obtain the range for LTE, Wi-Fi, and LoRa. With the new range, we can obtain the area over which BLE, LTE, Wi-Fi, and LoRa exist.

$$A_\tau = \begin{cases} \pi x_{BLE,fsmp}^2, & \text{for BLE} \\ \pi x_{LTE,fsmp}^2 - \pi r_{BLE,fsmp}^2, & \text{for LTE} \\ \pi x_{Wi\text{-}Fi,fsmp}^2 - \pi r_{LTE,fsmp}^2, & \text{for Wi-Fi} \\ B^2 - \pi x_{Wi\text{-}Fi,fsmp}^2, & \text{for LoRa} \end{cases} \tag{G.21}$$

where, τ represents the different communication technologies and x_τ is given as

$$x_\tau = \begin{cases} \sqrt{r_\tau^2 - h_i^2}, & \text{if } r_\tau > h_i; \\ 0, & \text{otherwise,} \end{cases} \tag{G.22}$$

Let P_{pr} represents the probability that a UAV is present in the area corresponding to a communication technology which is obtained as

$$P_{pr,\tau} = \frac{A_\tau}{B^2}. \tag{G.23}$$

Thus, the number of UAV-GS links connected to a communication technology is obtained as

$$n_\tau = P_{pr,\tau} N \tag{G.24}$$

The energy consumed by communication technology is obtained as

$$E_\tau = P_{pr,\tau} N (E_{Tr} + E_{d,fsmp,avg}) \tag{G.25}$$

Finally, the total energy consumed by the network is obtained as

$$E_{N/W} = \sum_\tau P_{pr,\tau} N (E_{Tr,\tau} + E_{d,fsmp,\tau,avg}) \tag{G.26}$$

Similar to (G.12) for the FS model, the average FSMP energy consumption $E_{d,fsmp,\tau,avg}$ for a communication technology τ is obtained as

$$E_{d,fsmp,\tau,avg} = \frac{1}{\pi(r_b^2 - r_a^2)} \int_0^{2\pi} \Bigg[\int_{r_a}^{r_c} (k\mathcal{E}_e + k\mathcal{E}_{fs} r^2) r \, dr \, d\theta + \int_{r_c}^{r_b} (k\mathcal{E}_e + k\mathcal{E}_{mp} r^4) r \, dr \, d\theta \Bigg] \tag{G.27}$$

Solving the above equation, the expression for the average energy consumption $E_{d,fsmp,\tau,avg}$ is given as

$$E_{d,fsmp,\tau,avg} = k\mathcal{E}_e + \frac{k}{(r_b^2 - r_a^2)}\left[\frac{k\mathcal{E}_{fs}}{2}(r_c^4 - r_a^4) + \frac{k\mathcal{E}_{mp}}{3}(r_b^6 - r_c^6)\right] \tag{G.28}$$

where r_a, r_b, and r_c are defined as

$$r_a = \begin{cases} 0, & \text{for BLE} \\ r_{BLE,fsmp}, & \text{for LTE} \\ r_{LTE,fsmp}, & \text{for Wi-Fi} \\ r_{Wi\text{-}Fi,fsmp}, & \text{for LoRa} \end{cases} \tag{G.29}$$

$$r_b = \begin{cases} r_{BLE,fsmp}, & \text{for BLE} \\ r_{LTE,fsmp}, & \text{for LTE} \\ r_{Wi\text{-}Fi,fsmp}, & \text{for Wi-Fi} \\ r_{LoRa,fsmp}, & \text{for LoRa} \end{cases} \tag{G.30}$$

and

$$r_c = \begin{cases} r_{BLE,fs}, & \text{for BLE} \\ r_{LTE,fs}, & \text{for LTE} \\ r_{Wi\text{-}Fi,fs}, & \text{for Wi-Fi} \\ r_{LoRa,fs}, & \text{for LoRa} \end{cases} \tag{G.31}$$

Similarly the average delay of the network $T_{avg,fsmp}$ can be derived as,

$$T_{avg,fsmp} = \frac{1}{\pi(r_b^2 - r_a^2)}\int_0^{2\pi}\int_{r_a}^{r_b}(T_{trans} + T_{prop})r\,dr\,d\theta \tag{G.32}$$

where the expressions for T_{trans} and T_{prop} are detailed in Section G.6. Solving the above equation, we obtain the expression for the average network delay $T_{avg,fsmp}$ as,

$$T_{avg,fsmp} = \frac{k}{R} + \frac{2(r_a^2 + r_b^2 + r_a r_b)}{3c(r_a + r_b)} \tag{G.33}$$

where r_a, r_b, and c are defined as above. It can be observed that the average network delay for both the hybrid FS and FSMP schemes is equal. This is expected as the delay in both cases is directly proportional to d_g. This is also verified through simulations in the subsequent sections.

G.5 RL-Based Energy-Efficient Data Transmission over Hybrid BLE/LTE/Wi-Fi/LoRa UAV-Assisted Wireless Network Formation

RL is a branch of machine learning in which an agent learns the optimal policy strategy through a set of actions. In a RL framework, the agent is the entity that

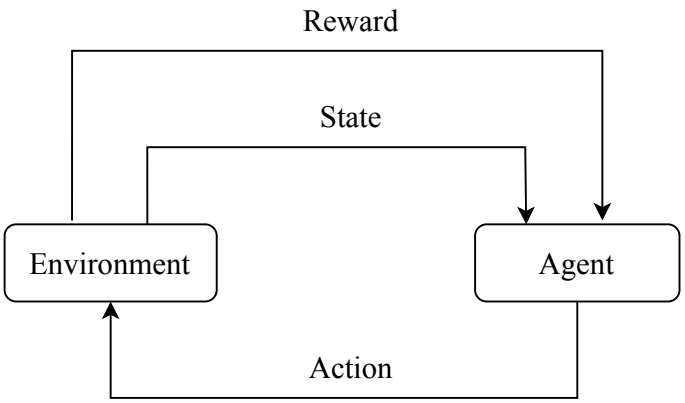


Figure G.2: An illustration of Q-learning.

Table G.3: Q-table corresponding to state-action pairs.

States	Actions			
	A_1	A_2	$\cdots$	A_M
S_1	$Q(S_1, A_1)$	$Q(S_1, A_2)$	$\cdots$	$Q(S_1, A_M)$
S_2	$Q(S_2, A_1)$	$Q(S_2, A_2)$	$\cdots$	$Q(S_2, A_M)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_N	$Q(S_N, A_1)$	$Q(S_N, A_2)$	$\cdots$	$Q(S_N, A_M)$

tries to learn and model the environment that it interacts with. The agent interacts with the environment through a set of trial-and-error actions. Based on these actions the environment either rewards or penalizes the agent. The feedback thus obtained allows the agent to navigate the environment by transitioning to new states, thereby updating its optimal policy strategy. RL is generally used to solve problems involving Markov Decision Process (MDP) where the state transition probability of moving to the next state is dependent only on the present state and not the previous other states. In this paper, we consider two RL algorithms such as Q-learning and DQN which are described next.

G.5.1 Q-Learning

In this section, we describe the proposed Q-learning approach for energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. Q-learning is a model-free reinforcement learning algorithm in which the agent directly interacts with the environment to achieve the optimal policy. The algorithm is value-based and finds the value function without any prior knowledge of the state transition probabilities [27]. Compared to model-based algorithms, Q-learning offers relatively less execution time and storage cost, provided the state and action space are not large. Since the algorithm operation is reward-dependent, proper design of the reward function is crucial for the algorithm's performance and conduct [28, 29]. In this work, the goal of the Q-learning algorithm is to transmit the data from UAV to GS with minimum energy consumption. In our hybrid system, the states, actions, and rewards as shown in Fig. G.2 are detailed as follows:

G.5.1.1 States

Let $S = \{s_n\}$ represent a set of N states for the Q-learning algorithm where $n = 1, 2, ..., N$. In this work, each state represents the communication link between the UAV and GS. Since there are N UAVs in the network, the number of UAV-GS communication links is N, and hence the number of states is N.

G.5.1.2 Action

Let each state $s_n \in S$ be associated with a set of M actions $A = \{a_m\}$, where each action represents the agent choosing a communication technology to assign to a UAV-GS link. In our hybrid system, there are four communication technologies and hence, there are four actions i.e., $M = 4$ in our Q-learning algorithm.

G.5.1.3 Reward/Penalty

Based on the action, the agent receives a reward/penalty from the environment. If γ is denoted as the reward/penalty, then $\gamma = \gamma(s_n, a_m)$ is the reward/penalty that an agent obtains when it is present in state s_n and performs an action a_m. In our hybrid system, the agent will receive the reward/penalty after selecting the communication technology for the UAV-GS link. Since the objective of the proposed work is energy-efficient data transmission over a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network, γ is considered in terms of the network energy consump-

tion. The expression for γ is defined as

$$\gamma = \begin{cases} \dfrac{10^{-}6}{E_{N/W}} & \text{(G.34)} \\ \dfrac{1}{E_{N/W}} & \text{(G.35)} \\ e^{-E_{N/W}} & \text{(G.36)} \\ \dfrac{W_1}{E_{1,N/W}} + \dfrac{W_2}{E_{2,N/W}}, & \text{(G.37)} \end{cases}$$

where, $E_{1,N/W}$ and $E_{2,N/W}$ denote the network energy consumed for transmitting k-bits of information and the network energy consumed for transmitting over a distance. Further, W_1 and W_2 are the weights assigned for $E_{1,N/W}$ and $E_{2,N/W}$, respectively. Since a higher reward translates to an optimized network, we consider γ as the inverse function of energy.

G.5.1.4 Updating the Q-value

The reward/penalty from the agent is used to update the Q-value corresponding to a state-action pair as given in Table G.3. When a UAV-GS link is assigned to a communication protocol, it results in a reward. The new Q-value is obtained as

$$Q(s_t, a_t) \leftarrow (1 - \lambda)Q(s_t, a_t) + \lambda(\gamma + \Delta \max_a Q(s_{t+1}, a)), \tag{G.38}$$

where, λ and Δ are the learning rate and discount corresponding to the Q-learning approach, respectively. Here, $\max_a Q(s_{t+1}, a)$ takes the maximum of the future reward and applies it to the reward for the current state.

Algorithm 8 describes the Q-learning approach proposed in this work. In the proposed Q-learning algorithm, initialize the Q-matrix with the number of UAV-GS links as rows and the communication protocols as columns. The UAV-GS links are ordered as rows in decreasing order of their distance to the ground station. Now, starting from the first row of the Q-matrix, randomly select a communication protocol. With the selected communication protocol, calculate the network energy consumption. Calculate the reward using (G.34), (G.35), (G.36), and (G.37) and update the Q-value of the link using (G.38). Next, transition to the next state based on the decreasing distance to the ground station. The state transition rule employed in this work is partially motivated by [30], where the authors have used decreasing channel state information as a means to transition to the next state. Repeat the above procedure until the UAV-GS link closest to the ground station is updated. The above steps are repeated again for a large number of iterations such that all links and communication protocols are covered. Once the training is completed, connect the UAV-GS link with a communication protocol which results in the highest Q-value. Calculate the network energy consumption and average network delay for the energy-efficient data transmission over a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. With the increase in the number of state-action pairs, the

Algorithm 8: Q-learning approach to determine the optimized hybrid UAV-assisted communication network

Input: N UAVs, M communication technologies, Maximum number of episodes K_{max}, λ, Δ, exploration-exploitation factor ε

Output: Network energy consumption and average network delay

Initialize the rows and columns of the Q-matrix with zeros;
Assign the UAV-GS links to the rows of the Q-matrix in descending order of distance to the ground station;
Assign the communication technologies to the columns of the Q-matrix;
Training
for $i = 1$ **to** K_{max} **do**
    Select the first state s_0;
    for $t = 1$ **to** N **do**
        Generate a random number $u \in (0, 1)$;
        if $u > \varepsilon$ **then**
            Select action from Q-matrix which has maximum Q-value $a_t = \max_a Q(s_t, a_t)$;
        else
            Select a random communication technology as action a_t;
        end
        Calculate the network energy consumption using (G.1), (G.3), and (G.4);
        Obtain γ using (G.34), (G.35), (G.36), and (G.37);
        Update the Q-value using (G.38);
        if $s_t == s_N$ **then**
            break;
        else
            Update to next state $s_t = s_{t+1}$ in the order of decreasing distance to ground station;
        end
    end
end
Validation
In each row of the Q-matrix, select the indices with maximum Q-value;
Assign the communication technology with the highest Q-value to the UAV-GS link;
Calculate the network energy consumption and average network delay;

Q-matrix employed in the Q-learning algorithm would require additional memory to store the Q-values. This can often lead to increased memory overhead thereby affecting the performance of the algorithm [30]. The DQN algorithm is set to resolve these issues by using a neural network to approximate the Q-values [31]. By using a neural network, the DQN algorithm essentially preserves the relative significance

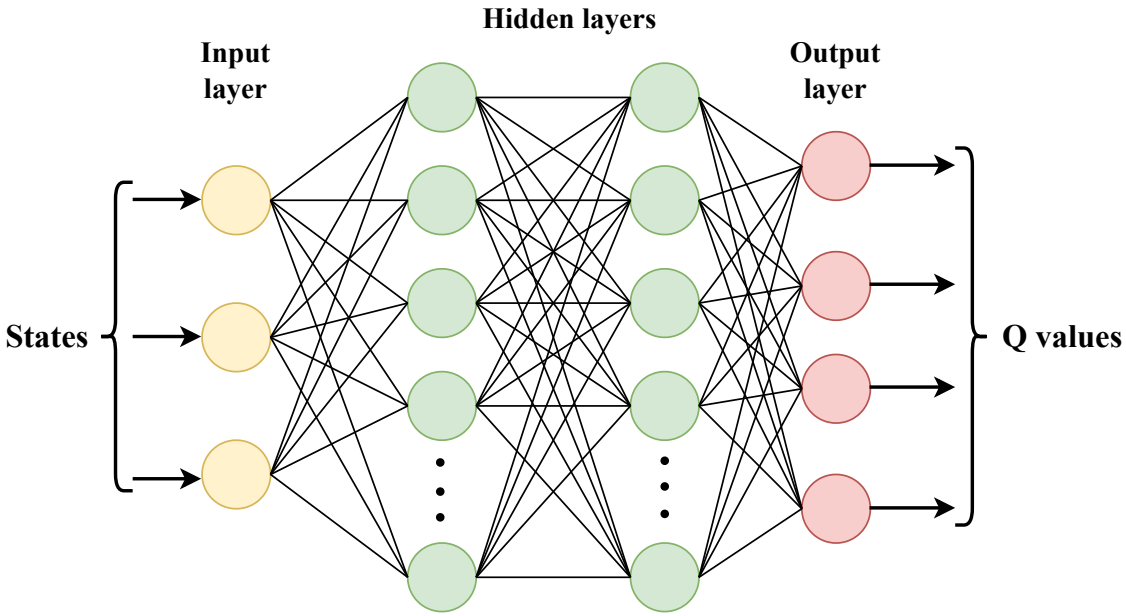


Figure G.3: Neural network model used in DQN.

between the Q-values and hence provides similar results as that of Q-learning [32]. Next, we discuss the details of the proposed DQN algorithm.

G.5.2 Deep Q-Network (DQN)

In this section, we present the DQN algorithm for energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network.

The DQN algorithm when applied to this work follows a similar construct as the Q-learning algorithm. States and actions follow the same descriptions as defined for Q-learning. Hence, for a UAV-assisted wireless network with N UAVs, the number of UAV-GS communication links is N, and hence the number of states is N. As there are four communication technologies to choose from, there are four actions. For a state s_t, the DQN algorithm follows a ε-greedy policy to select an action a_t. This means that a random action is selected (exploration) with a probability ε and the action corresponding to maximum $Q(s_t, a_t)$ is selected with probability $(1 - \varepsilon)$ (exploitation). Thus an exploration-exploitation tradeoff is provided so that the DQN algorithm is able to train successfully without falling into a local optima [30]. At any step t, the reward/penalty γ_t for state s_t and action a_t is calculated based on (G.35).

The DQN network is primarily made up of two neural network models, a policy network and a target network. Each UAV-GS link with its corresponding length is given as input to the neural networks. As shown in Fig. G.3, a fully connected neural network with two hidden layers forms the policy network. The first hidden layer consists of 256 neurons while the second hidden layer contains 128 neurons. All the layers are activated using the ReLu activation function during training. The target network is a cloned replica of the policy network. It has the same architecture as the

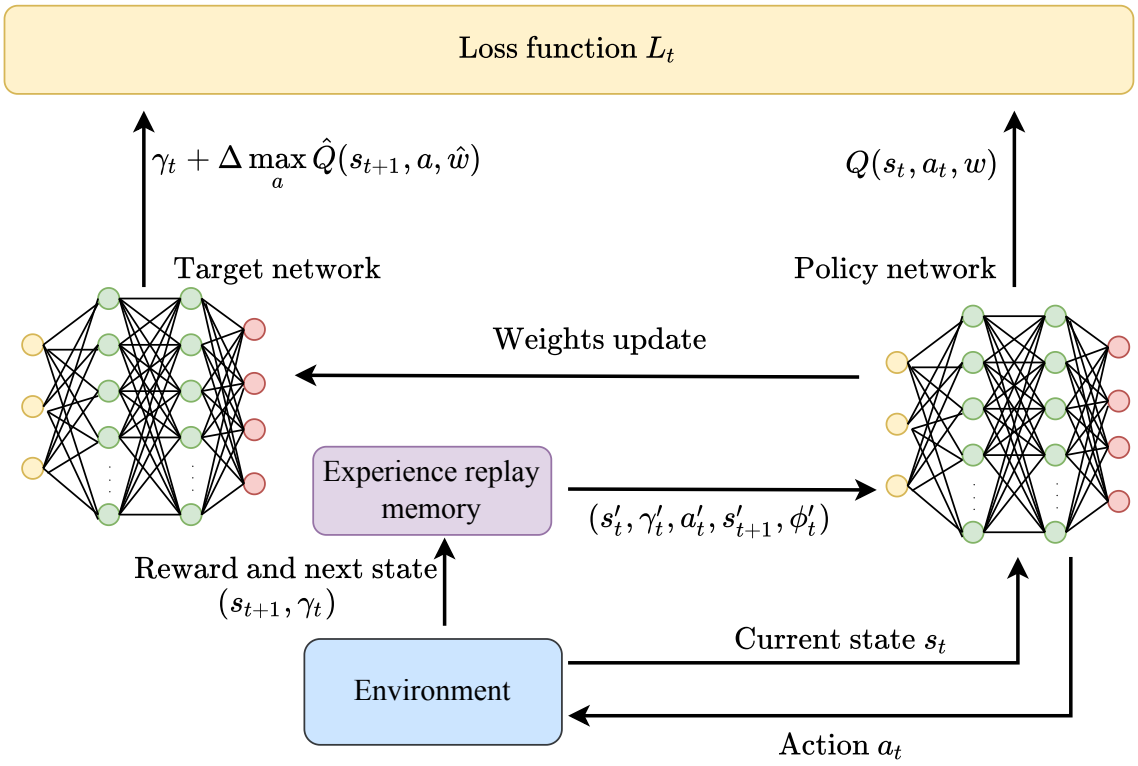


Figure G.4: Block diagram for the DQN algorithm.

policy network and serves the purpose of providing stability to the DQN algorithm while training. The output layer of the policy network consists of 4 outputs with a linear activation. The outputs correspond to the prediction of the state-action values $Q(s_t, a_t)$ at step t.

As shown in Fig. G.4, the action a_t for state s_t is selected based on ε-greedy policy. Choosing action a_t results in a state transition from state s_t to state s_{t+1} and a reward γ_t. The state transition follows the same distance-based rule used in Q-learning. The tuple $K = \{s_t, \gamma_t, a_t, s_{t+1}, \phi_t\}$ is then stored as a deque in the experience replay memory. Here, ϕ_t is a variable that indicates the final state. In the next step, a random mini-batch sample $(s'_t, \gamma'_t, a'_t, s'_{t+1}, \phi'_t)$ is selected which is used for training the policy network. The intention of selecting a random sample is to reduce the correlations between states and provide stability to the training process. The training is now carried out and the loss function is minimized. The loss function for step t is obtained as [30]

$$L_t = E\left[\left(\gamma_t + \Delta \max_a \hat{Q}(s_{t+1}, a, \hat{w}) - Q(s_t, a_t, w)\right)^2\right] \tag{G.39}$$

where Δ represents the discount factor ($\Delta \in [0, 1]$), $Q(s_t, a_t, w)$ represents the Q-value predicted using the policy network when trained with weights w, and $(\gamma_t + \Delta \max_a \hat{Q}(s_{t+1}, a, \hat{w})$ represents the output from the target network that is trained with weights $\hat{w}$. It is to be noted that the weights of the target network $\hat{w}$ are copied from the policy network every ζ step. Both the policy network and target network are trained using the Adam optimizer. The policy network is updated using the

below equation [30]

$$Q^*(s_t, a_t) \leftarrow Q(s_t, a_t) + \lambda(\gamma_t + \Delta \max_a \hat{Q}(s_{t+1}, a) - Q(s_t, a_t)) \tag{G.40}$$

where λ is the learning rate and $Q^*(s_t, a_t)$ is the new updated Q-value. The detailed steps of the proposed DQN algorithm are described in Algo. 9.

G.6 Evaluation Metrics

In this section, we derive the expressions for the performance metrics such as network energy consumption, and average network delay considered for the evaluation of the performance of the proposed method.

G.6.1 Average Network Delay

It is defined as the ratio of the sum of the delays incurred for transmitting the data in each UAV-GS link, to the total number of UAVs. The total delay of a UAV-GS link is the sum of the propagation delay and the transmission delay of the link.

G.6.1.1 Propagation delay

It is the delay incurred over a UAV-GS link for propagating the data over a radial distance d_g [33]. The expression for propagation delay is obtained as

$$T_{prop} = \frac{d_g}{c}, \tag{G.41}$$

where c is the velocity of the light which is equal to 3×10^8 m/s.

G.6.1.2 Transmission delay

It is the delay incurred for transmitting k bits of information over a UAV-GS link [33]. The expression for transmission delay is obtained as

$$T_{Tr} = \frac{k}{R}. \tag{G.42}$$

Thus, the average network delay is obtained as

$$T_{avg} = \frac{1}{N} \sum_{i=1}^{N} T_{total,i}, \tag{G.43}$$

where $T_{total} = T_{prop} + T_{Tr}$.

G.6.2 Network Energy Consumption

It is the total energy consumed for transmitting the data from each UAV to the GS. It is expressed as

$$E_{total} = \sum_{i=1}^{N} (E_{Tr,i} + E_{d,i}). \tag{G.44}$$

Algorithm 9: DQN approach to determine the optimized hybrid UAV-assisted wireless network

Input: N UAVs, M communication technologies, Number of episodes K_{max}, λ, Δ, ε, ζ

Output: Network energy consumption and average network delay

Initialize the experience replay memory, policy network weights w, and target network weights $\hat{w}$;
Arrange the UAV-GS links (states) in decreasing order of distance to the ground station;
Training
for $i = 1$ **to** K_{max} **do**
    Select the first state s_0 (first UAV-GS link) from the sorted list;
    $\phi_0 = 0$, $j = 0$;
    for $t = 1$ **to** N **do**
        Generate a random number $u \in (0, 1)$;
        if $u > \varepsilon$ **then**
            Select action $a_t = \max_a Q(s_t, a_t, w)$;
        else
            Select a random action a_t;
        end
        Calculate the network energy consumption using (G.1), (G.3), and (G.4);
        Obtain reward γ_t, using (G.35) ;
        Obtain next state s_{t+1} in order of decreasing distance to ground station;
        if $s_{t+1} == s_N$ **then**
            $\phi_t = 1$, $j = j + 1$;
        end
        Update experience replay memory with $K = (s_t, \gamma_t, a_t, s_{t+1}, \phi_t)$;
        if $\phi_t = 1$ **then**
            break;
        end
        When replay memory is full, randomly select a minibatch sample $(s'_t, \gamma'_t, a'_t, s'_{t+1}, \phi'_t)$;
        Predict $Q(s_t, a_t, w)$ from policy network;
        Predict $(\gamma_t + \Delta \max_a \hat{Q}(s_{t+1}, a, \hat{w})$ from target network;
        Compute loss using (G.39);
        Update weights w of policy network using (**??**);
        if $j == \zeta$ **then**
            Update the weights of target network $\hat{w}$;
        end
    end
    Decrease ε with a decay rate;
end

Validation
For each UAV, obtain the Q-values from the trained policy network;
Assign the communication technology with the highest Q-value to the UAV-GS link;
Calculate the network energy consumption and average network delay;

G.6.3 Energy Consumption for UAV Movement

The energy consumed by a UAV (in joules) to move a distance d is given by [34]

$$E_v = T \cdot d \tag{G.45}$$

where T represents the thrust force to move forward that is measured in Newtons (kg-m/sec^2). The thrust force is obtained as

$$T = \frac{m \times g}{r} \tag{G.46}$$

where m is the total weight of the aircraft (in kg), g is the acceleration due to gravity (9.8 m/sec^2), and r is a unitless parameter defined as the lift-to-drag ratio [34]. The lift-to-drag ratio essentially denotes the efficiency of aircraft design. A recommended lift-to-drag ratio is required to keep the aircraft airborne during steady flight. Commercial passenger aircraft have a lift-to-drag ratio between 10-20 whereas r for cruising helicopters is about 4. Typical lift-to-drag value for small and large scale UAVs is 3 [34].

G.7 Numerical Results

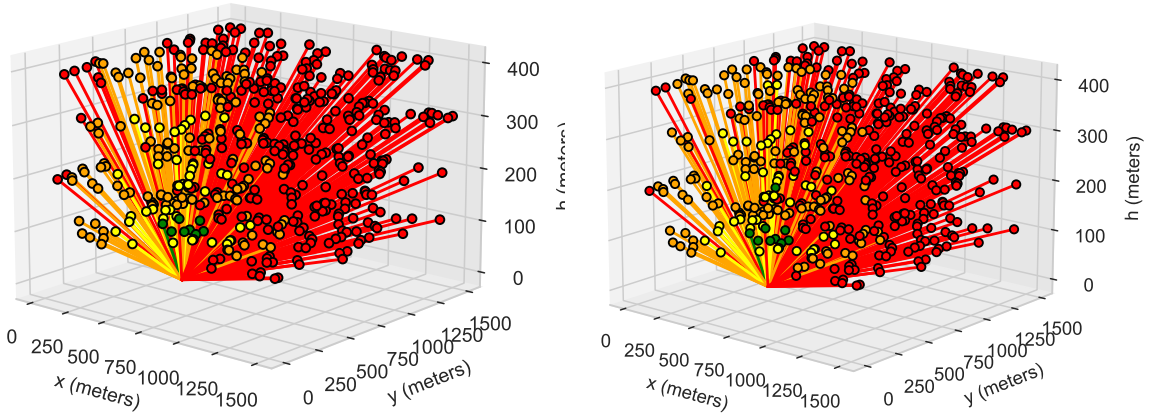
In this section, we first present the simulation setup considered for generating the simulation results. Then, we present the simulation results corresponding to both FS and FSMP models to verify the analytical derivations. We also present the simulation results to show the effect of the velocity of UAVs and packet size on the network energy consumption and average delay. Finally, the simulation results are presented to compare the performance of the proposed RL algorithms with other models.

G.7.1 Simulation Setup

The simulation setup is considered over a 1500 $\times$ 1500 m^2 area where the GS is located at $(500, 500, 0)$. A total of 500 UAVs are deployed over the area with varying heights of 100, 200, 300, and 400 meters. Each UAV is equipped with four different communication technologies such as BLE, LTE, Wi-Fi, and LoRa. The packet arrival rate follows Poisson distribution with randomly chosen mean taken from the set $\{1, 2, \cdots, 100\}$. Each packet contains 128 bits and hence the total number of bits received for each UAV follows a random distribution. The various simulation

Table G.4: The values of the parameters used for the performance evaluation and comparison of different methods.

Parameter	**Value**
Horizontal area	1500×1500
Hovering heights	$\{100, 200, 300, 400\}$
Number of UAVs, N	500
Location of GS	$(500, 500, 0)$
Velocity of UAV, ν	2
Packet length, k	128 bits
The parameter of Poisson distribution, λ	100
Speed of the transmission, c	3×10^8
$\mathcal{E}_e$	25 nJ/bit
$\mathcal{E}_{fs}$	10 pJ/bit/m^2
Initial energy at a UAV E	10 Joules



(a) (b)

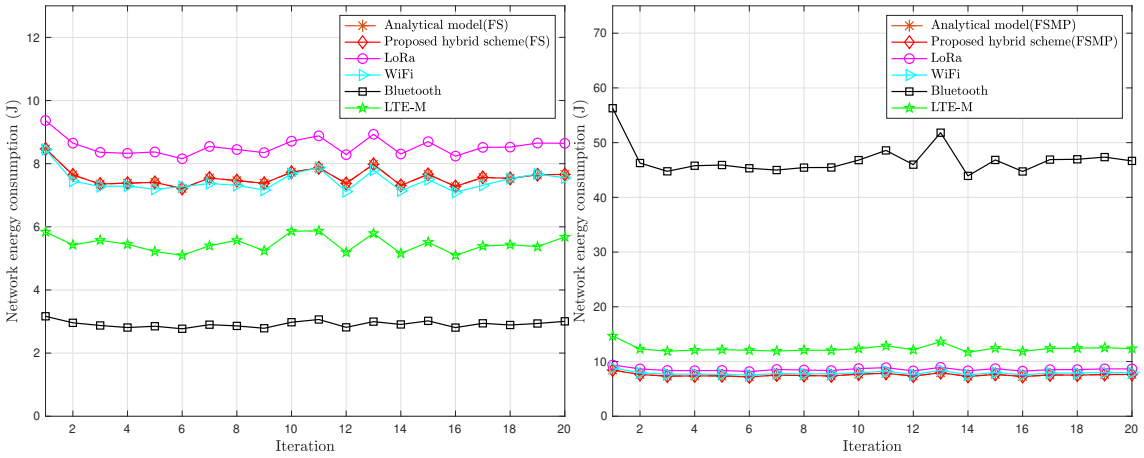
Figure G.5: An illustration of the UAV-assisted wireless network topology for (a) FS configuration and (b) FSMP configuration. Here, GS is considered to be located at $(500, 500, 0)$ which is the center of the terrestrial area. The colors green, yellow, orange, and red depict BLE, LTE, Wi-Fi, and LoRa communication protocols respectively.

parameters utilized for the different runs are listed in Table G.4. Additionally, the communication parameters for each specific technology are also provided in Table G.2.

G.7.2 Simulation Results (FS and FSMP)

Based on the above simulation parameters, we discuss the various numerical results obtained.

Fig. G.5a depicts the topology of the proposed hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network following the FS energy model. It can be observed from Fig. G.5a that the majority of the UAVs utilize the LoRa protocol (depicted

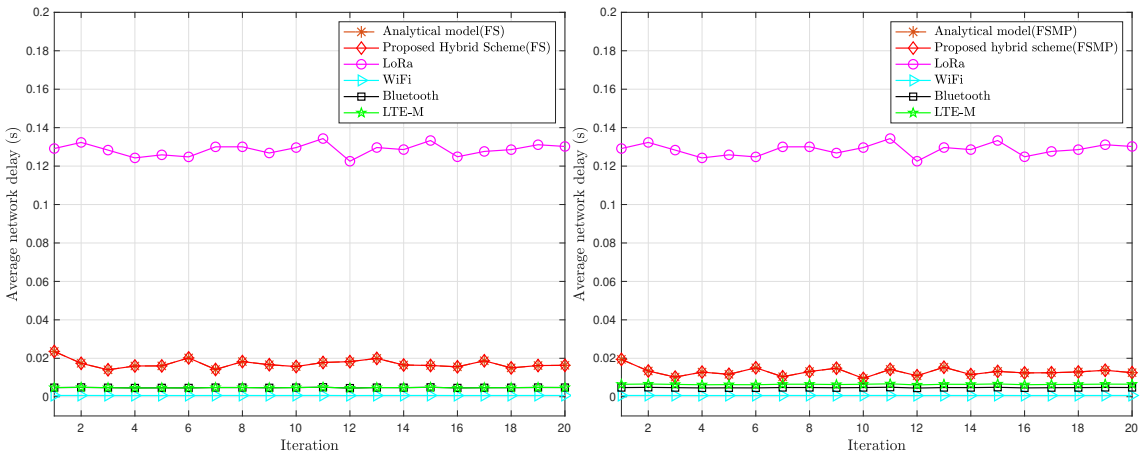


(a) (b)

Figure G.6: Variation of network energy consumption for (a) free space and (b) free space and multipath model for random UAV configurations. The analytical plots are obtained using equations (G.10) and (G.26), respectively. The ground station is located at $(500, 500, 0)$ for all the random configurations.

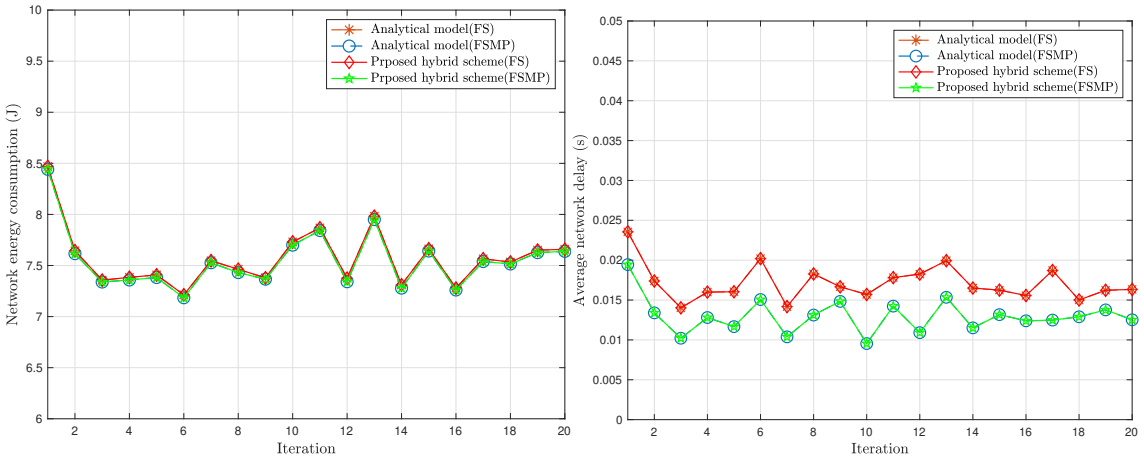
in red) to connect to the GS. The UAVs at medium range utilize Wi-Fi (yellow) and LTE (orange) to establish the UAV-GS link. The UAVs that are closer to the GS are connected via the BLE (green) communication protocol. When UAVs are closer to the GS, the BLE communication protocol consumes minimum energy as compared to other protocols. Hence, a small concentration of UAVs connected to BLE can be observed closer to the GS in Fig. G.5a. For UAVs situated farther than the path loss distance of BLE, the communication protocol with minimum energy consumption is connected. When the UAV-assisted wireless network follows the proposed hybrid scheme according to the FSMP energy model, the UAV-GS link connections are switched as shown in Fig. G.5b. It can be observed from Fig. G.5b that a small number of UAVs switch their communication protocol from LTE to BLE. This is due to the fact that the energy consumption of BLE while following the d^4 energy model is less than the energy consumption of LTE that is following the d^2 energy model. Similar changes can be observed between other communication protocols.

In Fig. G.6a, the network energy consumption for the proposed hybrid approach is compared with the conventional shortest path routing algorithm with individual communication technology while considering the FS path loss model. It can be observed from Fig. G.6a that the energy consumption exhibited by the proposed hybrid approach matches the energy consumption of the conventional approach for Wi-Fi protocol. However, the energy consumption is greater than that of BLE. This is due to the lower transmit power of BLE as compared to the average transmit power in the proposed hybrid scheme. The conventional approach for individual communication technology utilizes the hop-based shortest path algorithm to send packets from the UAV to GS. It is observed from Fig. G.6a that the energy consumption from the analytical model is equal to that of the simulation results, which is as expected. In Fig. G.6b, the proposed hybrid approach is compared to the



(a) (b)

Figure G.7: Variation of average network delay for (a) free space and (b) free space and multipath energy models for random UAV configurations. The analytical plots are obtained using equations (G.17) and (G.33), respectively. For all the configurations, the GS is located at $(500, 500, 0)$.



(a) (b)

Figure G.8: Variation of (a) network energy consumption and (b) average network delay for the proposed hybrid network. The plots corresponding to the analytical results are obtained using equations (G.10), (G.17), (G.26), and (G.33), respectively. In all these configurations, the GS is located at $(500, 500, 0)$.

conventional shortest path routing algorithm with individual communication technology in terms of network energy consumption. Here, we consider the FSMP path loss model. It can be observed from Fig. G.6b that the energy consumption with the proposed hybrid approach is lower than the conventional approach. It is also observed that the conventional approach utilizing BLE consumes more energy as compared to other approaches. This is due to the fact that r_τ is very low for BLE, resulting in a higher number of UAV-GS links following the d^4 model. Further, Fig. G.6b also illustrates the variation of the analytical results of the proposed approach.

It is observed from Fig. G.6b that the analytical results are matching with the simulation results verifying the analytical derivations.

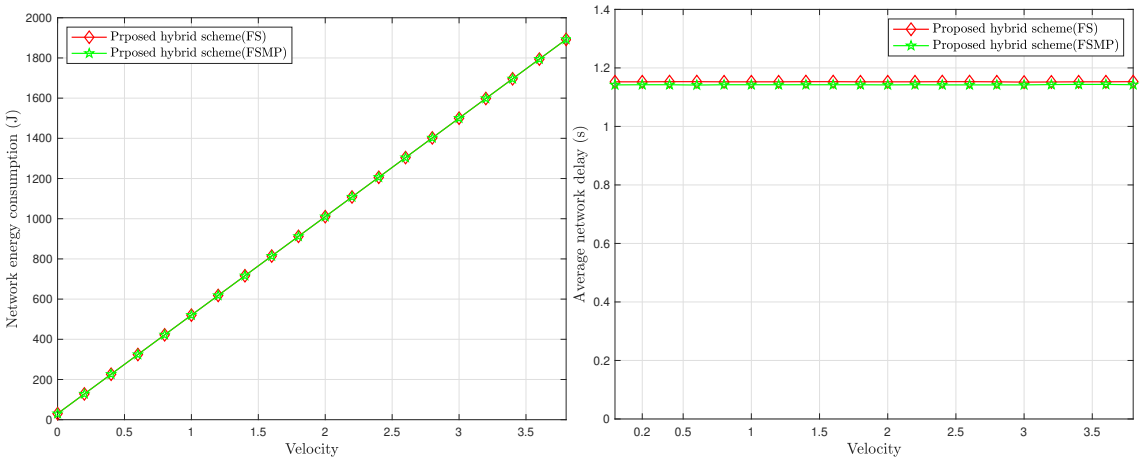
Figs. G.7a and G.7b provide the variation of average network delay for the proposed hybrid scheme with the conventional shortest path routing algorithm for individual communication technology. It can be observed from Figs. G.7a and G.7b that there is no significant variation in the delay for the proposed hybrid scheme that is following the FS and FSMP energy models. This can be attributed to the delay parameter that primarily depends on the transmission delay. The transmission delay depends on the data rate of the communication protocol. As LoRa has a low data rate compared to other communication protocols, LoRa has a higher delay. Thus, as observed from Figs. G.6b and G.7b, the proposed hybrid scheme following the FSMP energy model offers superior performance in terms of minimum energy consumption and low average network delay compared to the conventional shortest path routing algorithm with individual communication technology.

The variation of network energy consumption for the proposed hybrid scheme following the FS and FSMP energy models is depicted in Fig. G.8a. It is observed from Fig. G.8a, that the network energy consumption for the proposed hybrid scheme is similar across both the FS and FSMP energy models. Fig. G.8b shows the average network delay for the proposed hybrid scheme following the FS and FSMP energy models. It can be observed from Fig. G.8b that the proposed hybrid scheme with the FSMP energy model exhibits lower delay as compared to the proposed hybrid scheme following the FS energy model. This can be explained by observing the individual connections for each communication technology in both schemes. In the proposed hybrid scheme following the FS energy model, the distribution of UAVs connected to the different communication technologies are as follows: 14 connected to BLE, 153 connected to LTE, 279 connected to Wi-Fi, and 54 connected to LoRa. This distribution is altered when the proposed hybrid scheme with the FSMP energy model is used. The connections are switched with 21 UAVs connected to BLE and 162 UAVs connected to LTE. Moreover, the number of UAVs connected to LoRa reduces to 38. This change in the connection distribution decreases the delay in the proposed hybrid scheme utilizing the FSMP energy model. As BLE and LTE offer higher data rates as compared to the LoRa protocol, a lower delay is observed in the proposed hybrid scheme following the FSMP energy model. Thus the proposed hybrid scheme with FSMP energy model offers minimum energy consumption and reduced delay for the overall connected network.

G.7.2.1 Effect of UAV Velocity

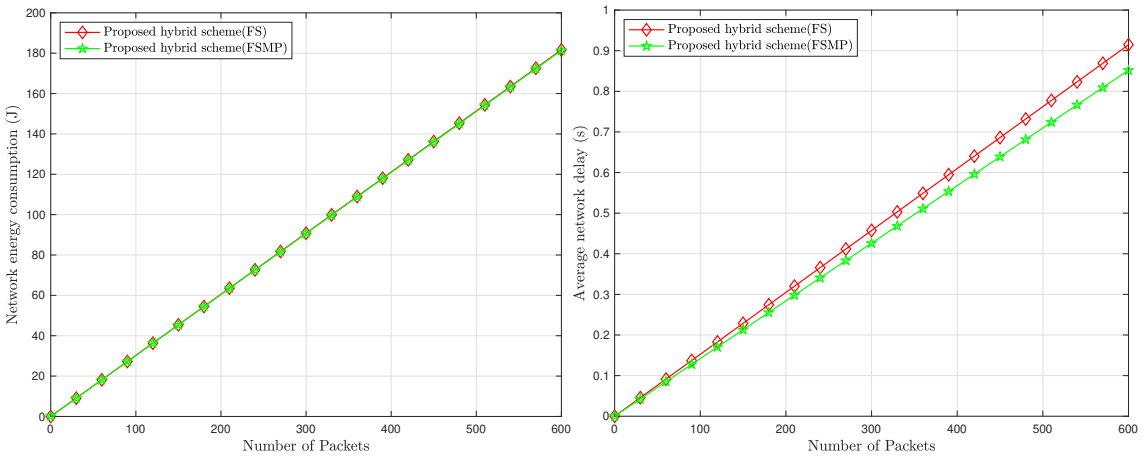
In this section, we show the effect of UAV velocity on the performance of the proposed hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. To obtain the variation of network energy consumption and average network delay, the velocity of each UAV is increased linearly from 0 to 3.8 m/s in steps of 0.2 m/s.

Figs. G.9a and G.9b show the variation of the network energy consumption and average delay with increasing velocity of the UAV. Here, the time of travel (moving



(a) (b)

Figure G.9: Effect of UAV velocity on (a) network energy consumption and (b) average network delay for both FS and FSMP model for random UAV configurations. GS is located at $(500, 500, 0)$ for all the random configurations.



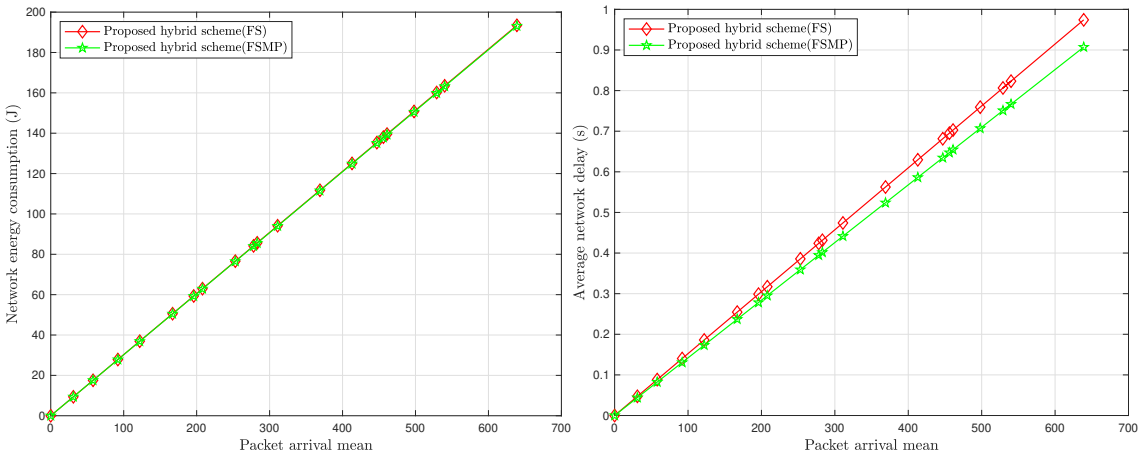
(a) (b)

Figure G.10: Effect of increasing the number of packets on (a) network energy consumption and (b) average network delay for both FS and FSMP model for random UAV configurations. GS is located at $(500, 500, 0)$ for all the random configurations.

time) for each UAV is considered to be 1 second. It is observed from Fig. G.9a that the energy consumption increases linearly with velocity. This is because, for a fixed time of travel, the distance traveled increases with increased velocity which consumes more energy. From Fig. G.9b, it can be observed that the velocity has a constant effect on the average delay as the time of travel is fixed for all velocities of UAVs.

G.7.2.2 Effect of Packet Size

In this section, we show the effect on the performance of the proposed hybrid system



(a) (b)

Figure G.11: Effect of increasing the mean of the packet arrival rate on (a) network energy consumption and (b) delay for both FS and FSMP model for random UAV configurations. GS is located at $(500, 500, 0)$ for all the random configurations.

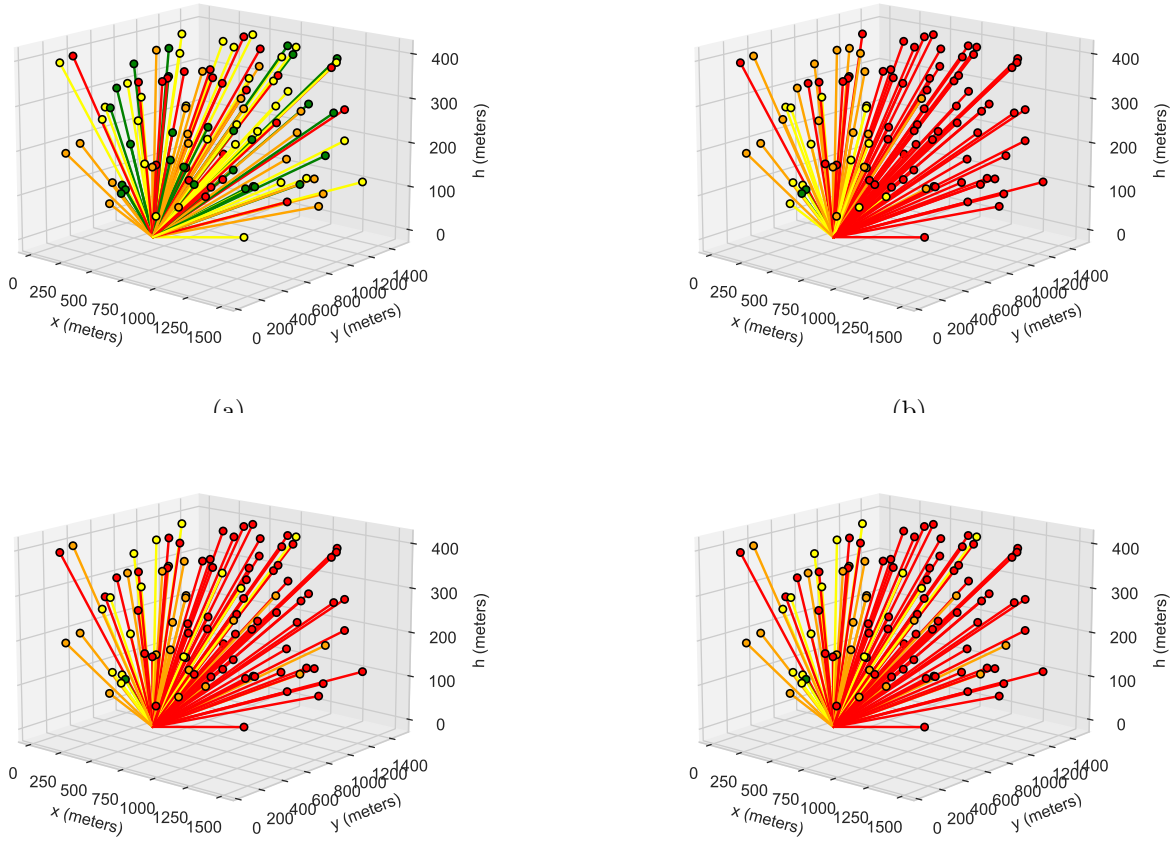
when the number of packets is increased. First, we consider a constant number of packets at each UAV which we increase from 1 to 600. Thereafter, we vary the mean of the Poisson distribution (packet arrival distribution) from 1 to 600.

Figs. G.10a and G.10b show the variation of the network energy consumption and average delay, respectively, for increasing values of the available packets at each UAV. Here, we varied the number of packets from 1 to 600. It is observed from Fig. G.10b that the energy consumption and average delay increase linearly with the number of packets. Further, Fig. G.10b also shows that the average delay is more for the FS model as compared to the FSMP model. This is due to the presence of multi-hop communication in the FS model which adds the queuing delay and processing delay at each hop.

Figs. G.11a and G.11b show the variation of the network energy consumption and average delay, respectively, for increasing values of the mean of the packet arrival rate. It is observed that an increase in the mean linearly increases the energy consumption and average delay as can be seen from Figs. G.11a and G.11b, respectively. Further, it is observed from Fig. G.11b that the average delay is more for the FS model as compared to the FSMP model. This is due to the presence of multi-hop communication in the FS model which adds the queuing delay and processing delay at each hop.

G.7.3 Performance Evaluation of the Proposed RL Algorithms

In this section, we present the numerical results comparing the performance of the proposed Q-learning (Algorithm 8) and DQN (Algorithm 9) algorithms with rule-based algorithm and hybrid random approach in which UAV-GS links are selected uniformly at random from BLE, LTE, Wi-Fi, and LoRa. Further, we show the



(a) (b)

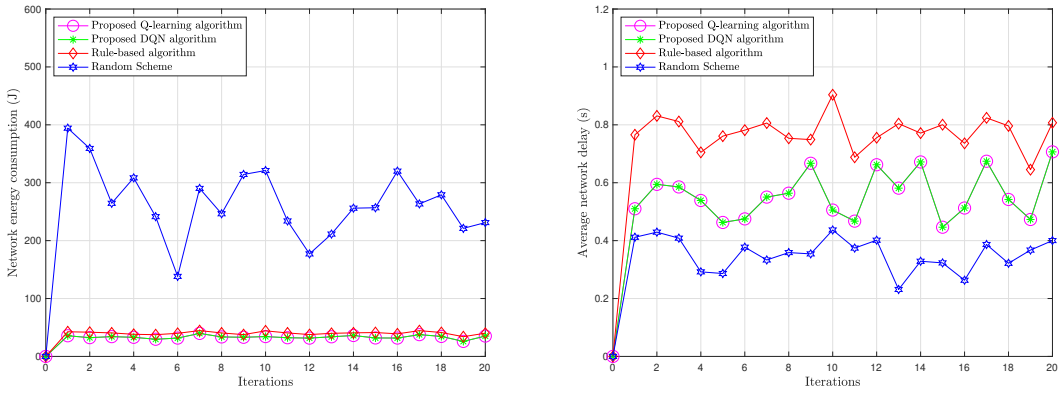
(c) (d)

Figure G.12: An illustration of hybrid UAV-assisted wireless network topology formed with (a) random hybrid scheme, (b) rule-based algorithm, (c) proposed Q-learning algorithm, and (d) proposed DQN algorithm. Here, GS is considered to be located at (500, 500, 0) which is the center of the terrestrial area. The colors green, yellow, orange, and red depict BLE, LTE, Wi-Fi, and LoRa communication protocols, respectively.

effectiveness of the proposed algorithms, by considering the PHY layer parameter, which denotes the number of transmissions required for a packet to get successfully delivered. The simulation setup is similar to the previous scenario wherein an area of $1500 \times 1500\ m^2$ is considered and the GS is placed at $(500, 500, 0)$. We consider a total of 100 UAVs deployed randomly over the considered area with varying heights of 100, 200, 300, and 400 meters. All other simulation parameters remain the same as defined in the previous section.

To study the behavior of Q-learning parameters we carried out extensive simulations using the learning rate and discount with different reward/penalty expressions defined in (G.34), (G.35), (G.36), and (G.37). When the reward is chosen from (G.34), (G.35), and (G.36), the Q-learning algorithm results in the optimized hybrid network at $\lambda = 0.7$ and $\Delta = 0.7$.

For the reward defined in (G.34), (G.35), and (G.36), we varied the Q-learning



(a) (b)

Figure G.13: Performance comparison of the proposed Q-learning algorithm, proposed DQN algorithm, rule-based algorithm, and random hybrid scheme in terms of (a) network energy consumption and (b) average delay for random UAV configurations. The ground station is located at $(500, 500, 0)$ for all the random configurations.

Table G.5: Variation of network energy consumption as a weighted sum of individual energy components $E_{1,N/W}$ and $E_{2,N/W}$. Here, W_1 and W_2 represent the weights.

$\boldsymbol{W_1}$	$\boldsymbol{W_2}$	$\boldsymbol{E_{total}}$ **(J)**
0.1	0.9	0.0204
0.2	0.8	0.0287
0.3	0.7	0.0392
0.4	0.6	0.0512
0.5	0.5	0.0650
0.6	0.4	0.0797
0.7	0.3	0.1002
0.8	0.2	0.1430
0.9	0.1	0.2589

parameters λ and Δ from 0.1 to 0.9 in intervals of 0.1. It is observed that the Q-learning algorithm started to attain the hybrid network with minimum energy consumption when λ is between 0.4 to 0.9 for all Δ. Table G.5 lists the network energy consumed for different weights when the reward is chosen as (G.37). Here, the weights are chosen from $\{0.1, 0.2, \cdots, 0.9\}$. We consider $\lambda = 0.7$ and $\Delta = 0.6$ which are best fit for the proposed Q-learning algorithm. As seen from Table G.5, $W_1 = 0.1$ and $W_2 = 0.9$ lowers the energy consumption. This means that $E_{2,N/W}$ has a higher impact than $E_{1,N/W}$ to attain the hybrid network with minimum energy consumption. This is due to the fact that for a UAV-GS link with k-bits of data, the energy consumed for transmission, $E_{1,N/W}$, is fixed, whereas, the energy consumed for the propagation, $E_{2,N/W}$ varies with the radial distance of the link. The DQN algorithm is trained under the same simulation setting as that of Q-learning. The

Table G.6: Parameters used for the DQN algorithm.

Parameter	Value
Number of hidden layers	2
Number of neurons in first layer	256
Number of neurons in first layer	128
Learning rate (neural network)	0.01
Discount factor	0.4
Batch size	64
Replay memory size	50000
Number of episodes	6400
Minibatch size	64
Epsilon	1 to 0.001

policy and target network are made up of fully-connected neural networks each with two hidden layers that contain 256 and 128 neurons, respectively. The exploration-exploitation factor ε is set to vary from 1 to 0.001. Other parameters related to the DQN algorithm are provided in Table G.6. We achieved the best performance when the learning rate and discount factor were set to 0.01 and 0.4, respectively. The DQN algorithm converged after 6400 episodes resulting in the formation of the hybrid UAV-assisted wireless network with minimum network energy consumption.

Fig. G.12a depicts the topology of the hybrid random network. Here, BLE is represented with green lines, LTE with yellow lines, Wi-Fi with orange lines, and LoRa with red lines. It can be observed from the figure that the connections are evenly distributed among the four protocols irrespective of any criteria due to the random distribution. Fig. G.12b shows the UAV-assisted wireless network obtained with the rule-based algorithm. From Fig. G.12b, it can be noticed that UAVs further away from the GS with high radial distance are connected to the LoRa protocol while the ones closer to the GS are connected to BLE. Figs. G.12c and G.12d show the hybrid network obtained with the proposed Q-learning and DQN algorithms, respectively. From Fig. G.12c and G.12d, it can be noticed that a UAV-GS link is connected to a communication technology that consumes minimum energy with the number of transmissions required.

We compare the performance of the proposed algorithms with the rule-based algorithm [35] and random hybrid scheme in terms of network energy consumption and average network delay as shown in Figs. G.13a and G.13b, respectively. The Q-learning algorithm utilizes the reward defined in (G.35) for generating these results. It is observed that the Q-learning and DQN algorithms outperform the rule-based algorithm in terms of both energy consumption and average delay as shown in Figs. G.13a and G.13b, respectively. This is primarily due to the learning characteristic of RL algorithms. In RL algorithms, the environment characteristics are learned during the training or exploration phase by which an accurate estimation of the network energy consumption is obtained. However, the rule-based algorithm assigns the communication technology based on the distance irrespective of the PHY layer

characteristics. The unknown PHY layer characteristics increase the number of retransmissions required for successful transmission of bits which in turn will further increase the energy consumption and delay. The proposed RL algorithms thus exhibit an edge over other schemes in learning and incorporating additional parameters that can affect the communication link in a seamless manner. It is also observed that the random hybrid scheme consumes a large amount of energy when compared to the proposed and rule-based algorithms. This is due to the fact that the number of UAVs connected with BLE for hybrid random network configuration is higher than the RL-based hybrid network as can be observed from Figs. G.12a and G.12c. This increases energy consumption as most of these links' geographical distance is greater than r which leads to d^4 energy consumption. Since the BLE offers a higher data rate than other communication technologies, the transmission delay is much less, which lowers the average network delay as can be observed from Fig. G.13b. To provide additional clarity on the performance of all the schemes, we provide explicit values from the simulation. In the 10-th iteration, the hybrid random network exhibits a network energy consumption of 1160 Joules and the rule-based algorithm expends 190 Joules of energy. The proposed algorithms however outperform all the schemes with the minimum network energy consumption of about 151 Joules as can be seen from Fig. G.13a. In terms of average network delay, the rule-based algorithm offers an average delay of about 0.71 seconds, the random hybrid scheme offers an average delay of about 0.344 seconds, and the proposed RL algorithms exhibit 0.5 seconds at 10-th iteration as can be seen from Fig. G.13b. From Figs. G.13a and G.13b, it can be observed that the DQN algorithm displays similar performance as that of the Q-learning algorithm. This is expected as DQN essentially follows the same mathematical principles as that of Q-learning. Compared to Q-learning, the DQN algorithm utilizes a neural network to learn features from the input and provide an approximate Q-value. If the amount of data increases, storing and searching for Q-values using Q-learning can lead to performance degradation. In this scenario, DQN outperforms Q-learning in terms of reduced memory consumption and computational efficiency. However, the DQN algorithm requires significantly large training time and fine-tuning as compared to Q-learning. For example, the simulation time required to train Q-learning is 20 minutes whereas DQN requires approximately 6 hours.

G.8 Conclusion and Future Work

In this work, we have proposed two RL algorithms such as Q-learning and DQN for energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network. The proposed RL algorithms take any random network as an input and learn it. Based on the learning, a hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network is formed by assigning the best communication technology to a link based on learning. We have also proposed the theoretical framework for energy-efficient data transmission over hybrid BLE/LTE/Wi-Fi/LoRa UAV-assisted wireless network for both free space and free space multipath path loss models. Further, we have derived the analytical expressions for the network energy consumption and

average network delay. Through extensive results, we verified the analytical expressions. We have also analyzed the effect of the velocity of UAVs and the number of packets on the performance of the proposed framework. Finally, it has been shown that the proposed RL algorithms result in better performance in terms of network delay and energy consumption when compared to rule-based algorithm and random hybrid scheme. In the future, we plan to incorporate the dataset obtained from physical layer parameters to evaluate the performance of the proposed algorithms.

References

[1] Dimosthenis C Tsouros, Stamatia Bibi, and Panagiotis G Sarigiannidis. A review on uav-based applications for precision agriculture. *Information*, 10(11):349, Nov. 2019.

[2] Manjula Sharma et al. Survey on unmanned aerial vehicle for Mars exploration: deployment use case. *Drones*, 6(1):4, Dec. 2021.

[3] Aakanksha Chowdhery and Kyle Jamieson. Aerial channel prediction and user scheduling in mobile drone hotspots. *IEEE/ACM Transactions on Networking*, 26(6):2679–2692, Nov. 2018.

[4] Arvind Merwaday and Ismail Guvenc. UAV assisted heterogeneous networks for public safety communications. In *Proc. IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 329–334, New Orleans, LA, USA, 2015.

[5] Yuntao Wang et al. Task offloading for post-disaster rescue in unmanned aerial vehicles networks. *IEEE/ACM Transactions on Networking*, pages 1–15, Jan. 2022.

[6] Sky Magic - Drone Light Shows. [Online]. Available: https://skymagic.show.

[7] Eyuel D. Ayele, Kallol Das, Nirvana Meratnia, and Paul J.M. Havinga. Leveraging BLE and LoRa in IoT network for wildlife monitoring system (WMS). In *Proc. IEEE World Forum on Internet of Things (WF-IoT)*, pages 342–348, Singapore, 2018.

[8] Yeduri Sreenivasa Reddy et al. Optimisation of indoor hybrid PLC/VLC/RF communication systems. *IET Communications*, 14(1):117–126, Jan. 2020.

[9] Mohamed Kashef, Mohamed Abdallah, and Naofal Al-Dhahir. Transmit power optimization for a hybrid PLC/VLC/RF communication system. *IEEE Transactions on Green Communications and Networking*, 2(1):234–245, Nov. 2018.

[10] Mengjie Yi et al. Deep reinforcement learning for fresh data collection in UAV-assisted IoT networks. In *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 716–721, Toronto, ON, Canada, 2020.

[11] Shu Fu et al. Towards energy-efficient UAV-assisted wireless networks using an artificial intelligence approach. *IEEE Wireless Communications*, pages 1–11, May 2022.

[12] Sabarish Krishna Moorthy, Maxwell Mcmanus, and Zhangyu Guan. ESN reinforcement learning for spectrum and flight control in THz-enabled drone networks. *IEEE/ACM Transactions on Networking*, 30(2):782–795, Nov. 2022.

[13] Yirga Yayeh Munaye, Rong-Terng Juang, Hsin-Piao Lin, and Getaneh Berie Tarekegn. Resource allocation for multi-UAV assisted IoT networks: A deep reinforcement learning approach. In *Proc. International Conference on Pervasive Artificial Intelligence (ICPAI)*, pages 15–22, Taipei, Taiwan, 2020.

[14] Xijian Zhong, Yan Guo, Ning Li, and Yancheng Chen. Joint optimization of relay deployment, channel allocation, and relay assignment for UAVs-aided D2D networks. *IEEE/ACM Transactions on Networking*, 28(2):804–817, Feb. 2020.

[15] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[16] S. M. Mahdi H. Daneshvar, Pardis Alikhah Ahari Mohajer, and Sayyed Majid Mazinani. Energy-efficient routing in WSN: A centralized cluster-based approach via grey wolf optimizer. *IEEE Access*, 7:170019–170031, Nov. 2019.

[17] Om Jee Pandey and Rajesh M. Hegde. Low-latency and energy-balanced data transmission over cognitive small world WSN. *IEEE Transactions on Vehicular Technology*, 67(8):7719–7733, May 2018.

[18] Bluetooth Wireless Technology. [Online]. Available: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/.

[19] A. N. Wilson, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Embedded sensors, communication technologies, computing platforms and machine learning for UAVs: A review. *IEEE Sensors Journal*, 22(3):1807–1826, Feb. 2022.

[20] 3gpp Standards for the IoT. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1805-iot_r14.

[21] 3gpp Release 13 Overview. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/1628-rel13.

[22] Evgeny Khorov, Anton Kiryanov, Andrey Lyakhov, and Giuseppe Bianchi. A tutorial on IEEE 802.11ax high efficiency WLANs. *IEEE Communications Surveys & Tutorials*, 21(1):197–216, Sep. 2019.

[23] Pycom OEM module. [Online]. Available: https://pycom.io/product/l04-oem-module/.

[24] Semtech SX1276 - Datasheet. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276.

[25] Xianjin Xia, Yuanqing Zheng, and Tao Gu. LiteNap: Downclocking LoRa reception. *IEEE/ACM Transactions on Networking*, 29(6):2632–2645, Jul. 2021.

[26] Tom M Apostol. *Calculus, Volume 1*. John Wiley & Sons, 1991.

[27] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*. MIT press Cambridge, 1998.

[28] Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[29] Kai-Chu Tsai, Zirui Zhuang, Ricardo Lent, Jingyu Wang, Qi Qi, Li-Chun Wang, and Zhu Han. Tensor-based reinforcement learning for network routing. *IEEE Journal of Selected Topics in Signal Processing*, 15(3):617–629, 2021.

[30] Fan Jiang, Zesheng Gu, Changyin Sun, and Rongxin Ma. Dynamic User Pairing and Power Allocation for NOMA with Deep Reinforcement Learning. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2021.

[31] Yejian Zhao, Yanhong Wang, Yuanyuan Tan, Jun Zhang, and Hongxia Yu. Dynamic Jobshop Scheduling Algorithm Based on Deep Q Network. *IEEE Access*, 9:122995–123011, 2021.

[32] Sifat Rezwan and Wooyeol Choi. Priority-Based Joint Resource Allocation With Deep Q-Learning for Heterogeneous NOMA Systems. *IEEE Access*, 9:41468–41481, 2021.

[33] Wei Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, Jun. 2004.

[34] Juan Zhang, James F. Campbell, Donald C. Sweeney II, and Andrea C. Hupman. Energy consumption models for delivery drones: A comparison and assessment. *Transportation Research Part D: Transport and Environment*, 90:102668, 2021.

[35] H Chtourou and Mohamed Haouari. A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. *Computers & industrial engineering*, 55(1):183–194, 2008.