



*drones*

# Advances in UAV Detection, Classification and Tracking

---

Edited by

Daobo Wang and Zain Anwar Ali

Printed Edition of the Special Issue Published in *Drones*

# **Advances in UAV Detection, Classification and Tracking**

# **Advances in UAV Detection, Classification and Tracking**

Editors

**Daobo Wang**

**Zain Anwar Ali**

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade • Manchester • Tokyo • Cluj • Tianjin



### *Editors*

Daobo Wang	Zain Anwar Ali
College of Automation	Electronic Engineering
Engineering	Sir Syed University of
Nanjing University of	Engineering & Technology
Aeronautics & Astronautics	Karachi
Nanjing	Pakistan
China	

### *Editorial Office*

MDPI  
St. Alban-Anlage 66  
4052 Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Drones* (ISSN 2504-446X) (available at: [www.mdpi.com/journal/drones/special\\_issues/UAV\\_Detection\\_Classification\\_Tracking](http://www.mdpi.com/journal/drones/special_issues/UAV_Detection_Classification_Tracking)).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

LastName, A.A.; LastName, B.B.; LastName, C.C. Article Title. <i>Journal Name</i> <b>Year</b> , Volume Number, Page Range.
--

**ISBN 978-3-0365-7561-2 (Hbk)**

**ISBN 978-3-0365-7560-5 (PDF)**

© 2023 by the authors. Articles in this book are Open Access and distributed under the Creative Commons Attribution (CC BY) license, which allows users to download, copy and build upon published articles, as long as the author and publisher are properly credited, which ensures maximum dissemination and a wider impact of our publications.

The book as a whole is distributed by MDPI under the terms and conditions of the Creative Commons license CC BY-NC-ND.

# Contents

About the Editors . . . . .	vii	
<b>Daobo Wang and Zain Anwar Ali</b> Editorial of Special Issue “Advances in UAV Detection, Classification and Tracking” Reprinted from: <i>Drones</i> <b>2023</b> , 7, 195, doi:10.3390/drones7030195 . . . . .		1
<b>Qi Zhang, Li Fan and Yulin Zhang</b> Simultaneous Astronaut Accompanying and Visual Navigation in Semi-Structured and Dynamic Intravehicular Environment Reprinted from: <i>Drones</i> <b>2022</b> , 6, 397, doi:10.3390/drones6120397 . . . . .		9
<b>Helu Zhou, Aitong Ma, Yifeng Niu and Zhaowei Ma</b> Small-Object Detection for UAV-Based Images Using a Distance Metric Method Reprinted from: <i>Drones</i> <b>2022</b> , 6, 308, doi:10.3390/drones6100308 . . . . .		29
<b>Zhen Wang and Yanhong Luo</b> Elliptical Multi-Orbit Circumnavigation Control of UAVS in Three-Dimensional Space Depending on Angle Information Only Reprinted from: <i>Drones</i> <b>2022</b> , 6, 296, doi:10.3390/drones6100296 . . . . .		49
<b>Ghulam E. Mustafa Abro, Saiful Azrin B. M. Zulkifli, Rana Javed Masood, Vijanth Sagayan Asirvadam and Anis Laouti</b> Comprehensive Review of UAV Detection, Security, and Communication Advancements to Prevent Threats Reprinted from: <i>Drones</i> <b>2022</b> , 6, 284, doi:10.3390/drones6100284 . . . . .		63
<b>Jiangkun Gong, Jun Yan, Deren Li and Deyong Kong</b> Detection of Micro-Doppler Signals of Drones Using Radar Systems with Different Radar Dwell Times Reprinted from: <i>Drones</i> <b>2022</b> , 6, 262, doi:10.3390/drones6090262 . . . . .		83
<b>Yuzhuang Wan, Yi Zhong, Yan Huang, Yi Han, Yongqiang Cui and Qi Yang et al.</b> ARSD: An Adaptive Region Selection Object Detection Framework for UAV Images Reprinted from: <i>Drones</i> <b>2022</b> , 6, 228, doi:10.3390/drones6090228 . . . . .		99
<b>Jiangkun Gong, Deren Li, Jun Yan, Huiping Hu and Deyong Kong</b> Using Classify-While-Scan (CWS) Technology to Enhance Unmanned Air Traffic Management (UTM) Reprinted from: <i>Drones</i> <b>2022</b> , 6, 224, doi:10.3390/drones6090224 . . . . .		115
<b>Farzaneh Dadrass Javan, Farhad Samadzadegan, Mehrnaz Gholamshahi and Farnaz Ashatari Mahini</b> A Modified YOLOv4 Deep Learning Network for Vision-Based UAV Recognition Reprinted from: <i>Drones</i> <b>2022</b> , 6, 160, doi:10.3390/drones6070160 . . . . .		125
<b>Chuanyun Wang, Zhongrui Shi, Linlin Meng, Jingjing Wang, Tian Wang and Qian Gao et al.</b> Anti-Occlusion UAV Tracking Algorithm with a Low-Altitude Complex Background by Integrating Attention Mechanism Reprinted from: <i>Drones</i> <b>2022</b> , 6, 149, doi:10.3390/drones6060149 . . . . .		147
<b>Amber Israr, Zain Anwar Ali, Eman H. Alkhamash and Jari Juhani Jussila</b> Optimization Methods Applied to Motion Planning of Unmanned Aerial Vehicles: A Review Reprinted from: <i>Drones</i> <b>2022</b> , 6, 126, doi:10.3390/drones6050126 . . . . .		161

<b>Mauro Leonardi, Gianluca Ligresti and Emilio Piracci</b> Drones Classification by the Use of a Multifunctional Radar and Micro-Doppler Analysis Reprinted from: <i>Drones</i> <b>2022</b> , 6, 124, doi:10.3390/drones6050124 . . . . .	<b>183</b>
<b>Xudong Li, Lizhen Wu, Yifeng Niu and Aitong Ma</b> Multi-Target Association for UAVs Based on Triangular Topological Sequence Reprinted from: <i>Drones</i> <b>2022</b> , 6, 119, doi:10.3390/drones6050119 . . . . .	<b>201</b>
<b>Jie-Tong Zou and Xiang-Yin Dai</b> The Development of a Visual Tracking System for a Drone to Follow an Omnidirectional Mobile Robot Reprinted from: <i>Drones</i> <b>2022</b> , 6, 113, doi:10.3390/drones6050113 . . . . .	<b>219</b>
<b>Muhammad Shafiq, Zain Anwar Ali, Amber Israr, Eman H. Alkhamash and Myriam Hadjouni</b> A Multi-Colony Social Learning Approach for the Self-Organization of a Swarm of UAVs Reprinted from: <i>Drones</i> <b>2022</b> , 6, 104, doi:10.3390/drones6050104 . . . . .	<b>231</b>
<b>Hanlin Sheng, Chen Zhang and Yulong Xiang</b> Mathematical Modeling and Stability Analysis of Tiltrotor Aircraft Reprinted from: <i>Drones</i> <b>2022</b> , 6, 92, doi:10.3390/drones6040092 . . . . .	<b>247</b>
<b>Abera Tullu, Mostafa Hassanalian and Ho-Yon Hwang</b> Design and Implementation of Sensor Platform for UAV-Based Target Tracking and Obstacle Avoidance Reprinted from: <i>Drones</i> <b>2022</b> , 6, 89, doi:10.3390/drones6040089 . . . . .	<b>265</b>
<b>Jiacheng Li, Jie Yang and Haibo Zhang</b> Research on Modeling and Fault-Tolerant Control of Distributed Electric Propulsion Aircraft Reprinted from: <i>Drones</i> <b>2022</b> , 6, 78, doi:10.3390/drones6030078 . . . . .	<b>285</b>

# About the Editors

## **Daobo Wang**

Prof. Dr. Wang Dao Bo is a distinguished academic and researcher in the field of automation engineering. He is currently affiliated with the College of Automation Engineering at Nanjing University of Aeronautics and Astronautics (NUAA), located in Nanjing, China.

Prof. Wang has made significant contributions to the development of control theory, system identification, and fault diagnosis of complex systems. He has authored or co-authored over 200 scientific papers and several books on these topics.

In addition to his research work, Prof. Wang has also served in various leadership roles at NUAA, including as the dean of the College of Automation Engineering. He is also an elected member of the Chinese Academy of Engineering, which is a prestigious honor in the academic community.

Overall, Prof. Wang Dao Bo is a respected and accomplished scholar in the field of automation engineering, with a long-standing record of academic excellence and contributions to the field.

## **Zain Anwar Ali**

Engr. Dr. Zain Anwar Ali received a B.S. degree in Electronic Engineering from Sir Syed University of Engineering and Technology, Karachi, Pakistan, in 2009. In the same year, he joined Sir Syed UET, as a Research Assistant in the Electronic Engineering department, and the next year in 2011, he was promoted to the post of junior lecturer due to his hard work and research contributions. He did his master's in Industrial Control and Automation from the Hamdard University of Engineering in 2012 and secured second position in his batch. After that, he was promoted to the post of Lecturer in 2012. Later, in September 2013, he joined Nanjing University of Aeronautics and Astronautics (NUAA) as a Ph.D. research scholar and joined Nanjing Strong flight electronics and machinery LTD to complete his Ph.D. experiment work there. In the year 2017, he completed his Ph.D. in the field of Control Theory and Control Engineering NUAA.

## Editorial

# Editorial of Special Issue “Advances in UAV Detection, Classification and Tracking”

Daobo Wang<sup>1</sup> and Zain Anwar Ali<sup>2,3,\*</sup> 

<sup>1</sup> College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

<sup>2</sup> School of Physics and Electronic Engineering, JiaYing University, Meizhou 514015, China

<sup>3</sup> Department of Electronic Engineering, Sir Syed University of Engineering and Technology, Karachi 75300, Pakistan

\* Correspondence: zainanwar86@hotmail.com

This is an editorial for a Special Issue of *Drones* titled “Advances in UAV Detection, Classification and Tracking”. The main aim of this Special Issue is to promote global collaboration and knowledge transfer between researchers. This Special Issue also includes the top 17 out of 42 selected papers received from academicians, researchers, and students in the field. These papers mainly emphasize recent trends in drone research.

The first paper is titled “Research on Modeling and Fault-Tolerant Control of Distributed Electric Propulsion Aircraft.” This study proposes a very promising distributed electric propulsion (DEP) system with high propulsion efficiency and low fuel consumption. The redundant thrusters of DEP aircraft increase the risk of faults in the propulsion system, so it is necessary to study fault-tolerant controls to ensure flight safety. Little research has been performed on coordinated thrust control, and the research on fault-tolerant controls for DEP systems is also in the preliminary stage. In this study, a mathematical model of a DEP aircraft was built. Aimed at the lateral and longitudinal control of DEP aircraft, a coordinated thrust-control method based on the control of total energy and total heading was designed. Furthermore, a fault-tolerant control strategy and control method were developed for faults in the propulsion system. Simulation results showed that the controller could control the thrust at the pre-fault level. The correctness and effectiveness of the coordinated thrust-control method designed and the fault-tolerant control method for DEP aircraft were theoretically verified. This study provides a theoretical basis for the future engineering application and development of control systems for DEP aircraft [1].

The second paper in this Special Issue is titled, “Design and Implementation of Sensor Platform for UAV-Based Target Tracking and Obstacle Avoidance.” Small-scale unmanned aerial vehicles are currently being deployed in urban areas for missions such as ground target tracking, crime scene monitoring, and traffic management. Aerial vehicles deployed in such cluttered environments are required to demonstrate robust, autonomous navigation and have both target-tracking and obstacle-avoidance capabilities. To this end, this work presents a simply designed but effective steerable sensor platform and implementation techniques for both obstacle avoidance and target tracking. The proposed platform is a two-axis gimbal system capable of roll and pitch/yaw. A mathematical model was developed to govern the dynamics of this platform. The performance of the platform was validated using a software-in-the-loop simulation. The simulation’s results showed that the platform can be effectively steered to all regions of interest except in a backwards direction. Due to its design layout and mount location, the platform can engage sensors for obstacle avoidance and target tracking as per UAV requirements. Moreover, steering the platform in any direction does not induce aerodynamic instability with respect to the unmanned aerial vehicle [2].

The third paper is titled “Mathematical Modeling and Stability Analysis of Tiltrotor Aircraft.” Mathematical modeling is the key problem in developing a tiltrotor. Therefore,

**Citation:** Wang, D.; Ali, Z.A.

Editorial of Special Issue “Advances in UAV Detection, Classification and Tracking”. *Drones* **2023**, *7*, 195.

<https://doi.org/10.3390/drones7030195>

Received: 1 March 2023

Accepted: 10 March 2023

Published: 14 March 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

this paper proposes a dividing modeling method that divides a tiltrotor into five parts (rotor, wing, fuselage, horizontal tail, and vertical fin) and develops aerodynamic models for each of them. In this way, the force and moment generated by each part can be obtained. First, a dynamic model of the rotor and its flapping angle expression was developed using the blade element theory. Using the mature lifting line theory, dynamic models of the wings, fuselage, horizontal tail, and vertical fin were then built. The rotors' dynamic interference on the wings and the nacelle tilt variations against the center of gravity and moment of inertia were taken into account. A non-linear tiltrotor simulation model was built in a MATLAB/Simulink simulation environment, and the Trim command was then applied to trim the tiltrotor. Finally, the XV-15 tiltrotor was used as an example to validate the rationality of the developed model. In the end, the non-linear simulation model was linearized to obtain a state-space matrix, thus performing a stability analysis of the tiltrotor [3].

The fourth paper, which compiled research on motion-planning algorithms for UAVs, is titled "Optimization Methods Applied to Motion Planning of Unmanned Aerial Vehicles: A Review." A flying robot is a system that can fly off and touch down to execute specific tasks. These flying robots are currently capable of flying without human control and can make situation-appropriate decisions with the help of onboard sensors and controllers. Among flying robots, unmanned aerial vehicles (UAVs) are highly attractive and applicable for military and civilian purposes. These UAV applications require motion-planning and collision-avoidance protocols to achieve improved robustness and a faster convergence rate for meeting their targets. Furthermore, optimization algorithms improve the performance of the system and minimize convergence errors. In this survey, diverse scholarly articles were gathered to highlight motion planning for UAVs using bio-inspired algorithms. This study will assist researchers in understanding the latest research performed on UAV motion planning with various optimization techniques. Moreover, this review presents the contributions and limitations of every article to demonstrate the effectiveness of the proposed work [4].

In the fifth paper, the authors present a study titled "The Development of a Visual Tracking System for a Drone to Follow an Omnidirectional Mobile Robot." This research developed a UAV visual tracking system that guides a drone in tracking a mobile robot and accurately landing on it when it stops moving. Two different-color LEDs were installed on the bottom of the drone. The visual tracking system on the mobile robot can detect the heading angle and the distance between the drone and the mobile robot. The heading angle and flight velocity in the pitch and roll direction of the drone were modified by PID controls so that the flying speed and angle are more accurate and the drone can land quickly. The PID tuning parameters were also adjusted according to the height of the drone. The system embedded in the mobile robot, which is equipped with Linux Ubuntu and processes images with OpenCV, can send a control command (SDK 2.0) to the Tello EDU drone via WIFI by using the UDP Protocol. The drone can auto-track the mobile robot. After the mobile robot stops moving, the drone can land on top of the mobile robot. Experimental results indicate that the drone can take off from the top of the mobile robot, visually track the mobile robot, and finally land on the top of the mobile robot accurately [5].

The sixth paper, "A Multi-Colony Social Learning Approach for the Self-Organization of a Swarm of UAVs", offers an improved method for the self-organization of a swarm of UAVs based on a social learning approach. To begin, the authors used three different colonies and three best members, i.e., unmanned aerial vehicles (UAVs), that are randomly placed in the colonies. This study used max-min ant colony optimization (MMACO) in conjunction with a social learning mechanism to plan an optimized path for an individual colony. A multi-agent system (MAS) chooses the most optimal UAV as the leader of each colony and selects the remaining UAVs as agents, which helps organize the randomly positioned UAVs into three different formations. The algorithm then synchronizes and connects the three colonies into a swarm and controls it using dynamic leader selection. The major contribution of this study was to hybridize two different approaches to produce a more

optimized, efficient, and effective strategy. The results verified that the proposed algorithm completed the given objectives. This study also compared the designed method with the non-dominated sorting genetic algorithm II (NSGA-II) to prove that the new method offers better convergence and reaches the target via a shorter route than NSGA-II [6].

The seventh paper, which proposes a triangular topological sequence for UAVs, is titled “Multi-Target Association for UAVs Based on Triangular Topological Sequence.” Thanks to their wide coverage and multi-dimensional perception, multi-UAV cooperative systems are highly regarded in the field of cooperative multi-target localization and tracking. However, due to the similarity of target visual characteristics and the limitations of UAV sensor resolution, it is difficult for UAVs to correctly distinguish visually similar targets. Incorrect correlation matching between targets results in the incorrect localization and tracking of multiple targets by multiple UAVs. In order to solve the association problem of targets with similar visual characteristics and to reduce the localization and tracking errors caused by target association errors based on the relative positions of the targets, the paper proposes a globally consistent target association algorithm for multiple UAV vision sensors based on triangular topological sequences. In contrast to Siamese neural networks and trajectory correlations, this algorithm uses the relative position relationship between targets to distinguish and correlate targets with similar visual features and trajectories. The sequence of neighboring target triangles is constructed using the relative position relationship to produce a specific triangular network. Moreover, this paper proposes a method for calculating the similarity of topological sequences with similar transformation invariances and a two-step optimal association method that considers global objective association consistency. Experimental flight results indicated that the algorithm achieves an association accuracy of 84.63%, and the two-step association is 12.83% more accurate than a single-step association. By conducting this research, the multi-target association problem of similar or even identical visual characteristics can be solved via cooperative surveillance and using multiple UAVs to track suspicious vehicles on the ground [7].

The eighth paper is titled “Drones Classification by the Use of a Multifunctional Radar and Micro-Doppler Analysis.” The use of radars to classify targets has received great interest in recent years, particularly for defense and military applications in which the development of sensor systems for identifying and classifying threatening targets is a mandatory requirement. In the specific case of drones, several classification techniques have already been proposed. Until recently, a micro-Doppler analysis in conjunction with machine learning tools was considered the most effective technique. The micro-Doppler signatures of targets are usually represented in the form of a spectrogram, which is a time–frequency diagram obtained by performing a short-time Fourier transform (STFT) on a radar return signal. Moreover, it is often possible to extract useful information from a target’s spectrogram that can also be used in a classification task. The main aim of this paper is to compare different methods of exploiting a drone’s micro-Doppler analysis on different stages of a multifunctional radar. Three different classification approaches were compared: a classic spectrogram-based classification; spectrum-based classification in which the received signal from the target is picked up after the moving target detector (MTD); and feature-based classification in which the received signal from the target undergoes a detection step after the MTD to extract and use discriminating features as input for the classifier. A theoretical model for the radar return signals of different types of drones and aerial targets was developed to compare the three approaches. This model was validated via a comparison with real recorded data, and it was used to simulate the targets. The results showed that the third approach (feature-based) demonstrated improved performance. Moreover, it also required less modification and less processing power when using a modern, multifunctional radar because it is capable of reusing most of the processing facilities that are already present [8].

The ninth paper is “Anti-Occlusion UAV Tracking Algorithm with a Low-Altitude Complex Background by Integrating Attention Mechanism.” In recent years, the increasing number of unmanned aerial vehicles (UAVs) in low-altitude airspace has introduced not

only convenience to work and life but also great threats and challenges. There are common problems in the process of UAV detection and tracking, such as target deformation, target occlusion, and the submersion of targets by complex background clutter. This paper proposes an anti-occlusion UAV tracking algorithm for low-altitude, complex backgrounds that integrates an attention mechanism to solve the problems of complex backgrounds and occlusions. The algorithm process is as follows: first, extracted features are enhanced using the SeNet attention mechanism. Second, an occlusion-sensing module is used to determine whether the target is occluded. If the target is not occluded, tracking continues. Otherwise, the LSTM trajectory-prediction network is used to predict the UAV position in subsequent frames from the UAV flight trajectory prior to occlusion. This study was verified using the OTB-100, GOT-10k, and integrated UAV datasets. The accuracy and success rates of the integrated UAV datasets were 79% and 50.5%, respectively, 10.6% and 4.9% higher than the rates of the SiamCAM algorithm. Experimental results showed that the algorithm could robustly track a small UAV in a low-altitude, complex background [9].

The tenth paper is titled, “A Modified YOLOv4 Deep Learning Network for Vision-Based UAV Recognition.” The use of drones in various applications has increased, as has their popularity among the general public. As a result, the possibility of drone misuse and their unauthorized intrusion into important places, such as airports and power plants, are increasing. This threatens public safety. Therefore, the accurate and rapid recognition of drone types is important for preventing their misuse and security problems caused by unauthorized drone access. Performing this operation from visible images is always associated with challenges, such as the drone’s small size, confusion with birds, the presence of hidden areas, and crowded backgrounds. In this paper, a novel and accurate technique with a change in the YOLOv4 network is presented to recognize four types of drones (multirotors, fixed-wing, helicopters, and VTOLs) and distinguish them from birds using a set of 26,000 visible images. In this network, more precise and detailed semantic features could be extracted by changing the number of convolutional layers. The performance of the basic YOLOv4 network was also evaluated on the same dataset, and the proposed model performed better than the basic network in solving the challenges. The proposed model also achieved automated, vision-based recognition with a loss of 0.58 in the training phase and an 83% F1-score, 83% accuracy, 83% mean average precision (mAP), and 84% intersection over union (IoU) in the testing phase. These results represent a slight improvement of 4% in these evaluation criteria over the basic YOLOv4 model [10].

The 11th paper is titled “Using Classify-While-Scan (CWS) Technology to Enhance Unmanned Air Traffic Management (UTM).” Drone detection radar systems have been verified to support unmanned air traffic management (UTM). In this paper, the authors propose the use of classify-while-scan (CWS) technology to improve the detection performance of drone-detection radar systems and to enhance UTM applications. The CWS recognizes radar data from each radar cell in the radar beam using an advanced automatic target recognition (ATR) algorithm. It then integrates the recognized results into the tracking unit to obtain real-time situational-awareness results for the entire surveillance area. Real X-band radar data, collected in a coastal environment, demonstrated a significant advancement in a powerful situational-awareness scenario in which birds were chasing a ship to feed on fish. The CWS technology turns drone-detection radars into a sense-and-alert platform that revolutionizes UTM systems by reducing the detection unit’s detection response time (DRT) [11].

The 12th paper, “ARSD: An Adaptive Region Selection Object Detection Framework for UAV Images”, proposes an object detection framework for UAVs. The performance of object detection has greatly improved due to the rapid development of deep learning. However, object detection in high-resolution images from unmanned aerial vehicles images remains a challenging problem for three main reasons: (1) the objects in aerial images have different scales and are usually small; (2) the images are high-resolution, but state-of-the-art object-detection networks are of a fixed size; (3) the objects are not evenly distributed in aerial images. To solve these problems, the authors proposed a two-stage adaptive

region selection detection framework. An overall region detection network was first applied to coarsely localize the object. A fixed-point, density-based target-clustering algorithm and an adaptive selection algorithm were then designed to select object-dense sub-regions. The object-dense sub-regions were sent to a key region detection network where the results were fused with the results from the first stage. Extensive experiments and comprehensive evaluations on the VisDrone2021-DET benchmark datasets demonstrated the effectiveness and adaptiveness of the proposed framework. Experimental results showed that the proposed framework outperformed the existing baseline methods by 2.1% without additional time consumption in terms of the mean average precision (mAP) [12].

The 13th paper is titled “Comprehensive Review of UAV Detection, Security, and Communication Advancements to Prevent Threats.” Over time, unmanned aerial vehicles (UAVs), also known as drones, have been used in very different ways. Advancements in key UAV areas include detection (including radio frequency and radar), classification (including micro-, mini-, close-, short-, and medium-range; medium-range endurance; low-altitude, deep-penetration; low-altitude, long-endurance; and medium-altitude, long-endurance), tracking (including lateral tracking, vertical tracking, a moving aerial pan with a moving target, and a moving aerial tilt with a moving target), and so forth. Even with these improvements and advantages, security and privacy can still be ensured by researching a number of key aspects of unmanned aerial vehicles, such as jamming a UAV’s control signals and redirecting them for a high-assault activity. This review article examined the privacy issues related to drone standards and regulations. The manuscript provides a comprehensive answer to these limitations. In addition to updated information on current legislation and the many classes that can be used to establish communication between a ground control room and an unmanned aerial vehicle, this article provides a basic overview of unmanned aerial vehicles. This review provides readers with an understanding of UAV shortcomings, recent advancements, and strategies for addressing security issues, assaults, and limitations. The open research areas described in this manuscript can be utilized to create novel methods for strengthening the security and privacy of unmanned aerial vehicles [13].

The 14th paper is “Detection of Micro-Doppler Signals of Drones Using Radar Systems with Different Radar Dwell Times.” Not all drone radar dwell times are suitable for detecting the micro-Doppler signals (or jet engine modulation, JEM) produced by rotating blades in the radar signals of drones. Theoretically, any X-band drone radar system can detect the micro-Doppler effects of blades due to the micro-Doppler effect and the partial resonance effect. However, the authors of this paper analyzed radar data from three radar systems with different radar dwell times and similar resolutions for frequency and velocity. These systems, Radar- $\alpha$ , Radar- $\beta$ , and Radar- $\gamma$ , had radar dwell times of 2.7 ms, 20 ms, and 89 ms, respectively. The results indicated that Radar- $\beta$  is the best radar for detecting the micro-Doppler signals (i.e., JEM signals) produced by the rotating blades of DJI Phantom 4, a quadrotor drone. Radar- $\beta$  achieved the best results because the detection probability for JEM signals was almost 100% with approximately two peaks that had similar magnitudes to the body Doppler. In contrast, Radar- $\alpha$  could barely detect any micro-Doppler effects, and Radar- $\gamma$  detected only weak micro-Doppler signals at a magnitude of only 10% of the body Doppler’s magnitude. A proper radar dwell time is the key to micro-Doppler detection. This research provides an idea for designing a cognitive micro-Doppler radar by changing the radar dwell time to detect and track the micro-Doppler signals of drones [14].

The 15th paper is titled, “Elliptical Multi-Orbit Circumnavigation Control of UAVS in Three-Dimensional Space Depending on Angle Information Only.” In order to analyze the circumnavigation tracking problem in a complex, three-dimensional space, this paper proposes a UAV group circumnavigation control strategy in which the UAV circumnavigation orbit is an ellipse for which its size can be adjusted arbitrarily. At the same time, the UAV group can be assigned to multiple orbits for tracking. The UAVs only have the target’s angle information, and the target’s position information can be obtained by using the angle information and the proposed three-dimensional estimator, thereby establishing an ideal

relative velocity equation. By constructing the error dynamic equation between the actual relative velocity and the ideal relative velocity, the three-dimensional circumnavigation problem is transformed into a velocity-tracking problem. Since UAVs are easily disturbed by external factors during flight, a sliding mode control was used to improve the system's robustness. Finally, the effectiveness of the control law and its robustness to unexpected situations were verified via simulations [15].

The 16th paper is titled, "Small-Object Detection for UAV-Based Images Using a Distance Metric Method." Object detection is important in unmanned aerial vehicle (UAV) reconnaissance missions. However, since UAVs fly at a high altitude to obtain a large reconnaissance view, the objects captured often have a small pixel size and their categories have high uncertainties. Given the limited computing capability of UAVs, large detectors based on convolutional neural networks (CNNs) have difficulty achieving real-time detection performance. To address these problems, the authors designed a small-object detector for UAV-based images in this paper. They modified the backbone of YOLOv4 according to the characteristics of small-object detection and improved the performance of small-object positioning by modifying the positioning loss function. Using the distance metric method, the proposed detector can classify trained and untrained objects by using the objects' features. Furthermore, the authors designed two data-augmentation strategies to enhance the diversity of the training set. The method was evaluated on a small-object dataset and obtained 61.00% on trained objects and 41.00% on untrained objects with 77 frames per second (FPS). Flight experiments confirmed the utility of the approach on small UAVs, which demonstrated a satisfying detection performance and real-time inference speed [16].

The 17th paper is "Simultaneous Astronaut Accompanying and Visual Navigation in Semi-Structured and Dynamic Intravehicular Environment." The application of intravehicular robotic assistants (IRA) can save valuable working hours for astronauts in space stations. There are various types of IRAs, such as those that accompany drones working in microgravity and a dexterous humanoid robot for collaborative operations. In either case, the ability to navigate and work alongside human astronauts lays the foundation for IRA deployment. To address this problem, this paper proposes a framework of simultaneous astronaut accompaniment and visual navigation. The framework contains a customized astronaut detector, an intravehicular navigation system, and a probabilistic model for astronaut visual tracking and motion prediction. The customized detector was designed to be lightweight. It achieved superior performance (AP@0.5 of 99.36%) for astronaut detection in diverse postures and orientations during intravehicular activities. A map-based visual navigation method was proposed for accurate localization with 6 DoF (1~2 cm, 0.5°) in semi-structured environments. To ensure navigation robustness in dynamic scenes, the feature points within the detected bounding boxes were filtered out. A probabilistic model was formulated based on the map-based navigation system and the customized astronaut detector. Both trajectory correlation and geometric similarity clues were incorporated into the model for stable visual tracking and the trajectory estimation of the astronaut. The overall framework enables the robotic assistant to track and identify the astronaut efficiently during intravehicular activities and can provide foresighted services while moving. The overall performance and superiority of the proposed framework were verified by conducting extensive ground experiments in a space station mockup [17].

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References



1. Li, J.; Yang, J.; Zhang, H. Research on Modeling and Fault-Tolerant Control of Distributed Electric Propulsion Aircraft. *Drones* **2022**, *6*, 78. [CrossRef]
2. Tullu, A.; Hassanalian, M.; Hwang, H.-Y. Design and Implementation of Sensor Platform for UAV-Based Target Tracking and Obstacle Avoidance. *Drones* **2022**, *6*, 89. [CrossRef]

3. Sheng, H.; Zhang, C.; Xiang, Y. Mathematical Modeling and Stability Analysis of Tiltrotor Aircraft. *Drones* **2022**, *6*, 92. [CrossRef]
4. Israr, A.; Ali, Z.A.; Alkhamash, E.H.; Jussila, J.J. Optimization Methods Applied to Motion Planning of Unmanned Aerial Vehicles: A Review. *Drones* **2022**, *6*, 126. [CrossRef]
5. Zou, J.-T.; Dai, X.-Y. The Development of a Visual Tracking System for a Drone to Follow an Omnidirectional Mobile Robot. *Drones* **2022**, *6*, 113. [CrossRef]
6. Shafiq, M.; Ali, Z.A.; Israr, A.; Alkhamash, E.H.; Hadjouni, M. A Multi-Colony Social Learning Approach for the Self-Organization of a Swarm of UAVs. *Drones* **2022**, *6*, 104. [CrossRef]
7. Li, X.; Wu, L.; Niu, Y.; Ma, A. Multi-Target Association for UAVs Based on Triangular Topological Sequence. *Drones* **2022**, *6*, 119. [CrossRef]
8. Leonardi, M.; Ligresti, G.; Piracci, E. Drones Classification by the Use of a Multifunctional Radar and Micro-Doppler Analysis. *Drones* **2022**, *6*, 124. [CrossRef]
9. Wang, C.; Shi, Z.; Meng, L.; Wang, J.; Wang, T.; Gao, Q.; Wang, E. Anti-Occlusion UAV Tracking Algorithm with a Low-Altitude Complex Background by Integrating Attention Mechanism. *Drones* **2022**, *6*, 149. [CrossRef]
10. Dadrass Javan, F.; Samadzadegan, F.; Gholamshahi, M.; Ashatari Mahini, F. A Modified YOLOv4 Deep Learning Network for Vision-Based UAV Recognition. *Drones* **2022**, *6*, 160. [CrossRef]
11. Gong, J.; Li, D.; Yan, J.; Hu, H.; Kong, D. Using Classify-While-Scan (CWS) Technology to Enhance Unmanned Air Traffic Management (UTM). *Drones* **2022**, *6*, 224. [CrossRef]
12. Wan, Y.; Zhong, Y.; Huang, Y.; Han, Y.; Cui, Y.; Yang, Q.; Li, Z.; Yuan, Z.; Li, Q. ARSD: An Adaptive Region Selection Object Detection Framework for UAV Images. *Drones* **2022**, *6*, 228. [CrossRef]
13. Abro, G.E.M.; Zulkifli, S.A.B.M.; Masood, R.J.; Asirvadam, V.S.; Laouti, A. Comprehensive Review of UAV Detection, Security, and Communication Advancements to Prevent Threats. *Drones* **2022**, *6*, 284. [CrossRef]
14. Gong, J.; Yan, J.; Li, D.; Kong, D. Detection of Micro-Doppler Signals of Drones Using Radar Systems with Different Radar Dwell Times. *Drones* **2022**, *6*, 262. [CrossRef]
15. Wang, Z.; Luo, Y. Elliptical Multi-Orbit Circumnavigation Control of UAVS in Three-Dimensional Space Depending on Angle Information Only. *Drones* **2022**, *6*, 296. [CrossRef]
16. Zhou, H.; Ma, A.; Niu, Y.; Ma, Z. Small-Object Detection for UAV-Based Images Using a Distance Metric Method. *Drones* **2022**, *6*, 308. [CrossRef]
17. Zhang, Q.; Fan, L.; Zhang, Y. Simultaneous Astronaut Accompanying and Visual Navigation in Semi-Structured and Dynamic Intravehicular Environment. *Drones* **2022**, *6*, 397. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

## Article

# Simultaneous Astronaut Accompanying and Visual Navigation in Semi-Structured and Dynamic Intravehicular Environment

Qi Zhang <sup>1,\*</sup> , Li Fan <sup>2,3</sup> and Yulin Zhang <sup>2,3</sup> 

<sup>1</sup> College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China

<sup>2</sup> College of Control Science and Engineering, Zhejiang University, Hangzhou 310000, China; fanli77@zju.edu.cn (L.F.); zhangyulin@zju.edu.cn (Y.Z.)

<sup>3</sup> Huzhou Institute, Zhejiang University, Huzhou 313000, China

\* Correspondence: zhangqi9241@alumni.nudt.edu.cn

**Abstract:** The application of intravehicular robotic assistants (IRA) can save valuable working hours for astronauts in space stations. There are various types of IRA, such as an accompanying drone working in microgravity and a dexterous humanoid robot for collaborative operations. In either case, the ability to navigate and work along with human astronauts lays the foundation for their deployment. To address this problem, this paper proposes the framework of simultaneous astronaut accompanying and visual navigation. The framework contains a customized astronaut detector, an intravehicular navigation system, and a probabilistic model for astronaut visual tracking and motion prediction. The customized detector is designed to be lightweight and has achieved superior performance (AP@0.5 of 99.36%) for astronaut detection in diverse postures and orientations during intravehicular activities. A map-based visual navigation method is proposed for accurate and 6DoF localization (1~2 cm, 0.5°) in semi-structured environments. To ensure the robustness of navigation in dynamic scenes, feature points within the detected bounding boxes are filtered out. The probabilistic model is formulated based on the map-based navigation system and the customized astronaut detector. Both trajectory correlation and geometric similarity clues are incorporated into the model for stable visual tracking and trajectory estimation of the astronaut. The overall framework enables the robotic assistant to track and distinguish the served astronaut efficiently during intravehicular activities and to provide foresighted service while in locomotion. The overall performance and superiority of the proposed framework are verified through extensive ground experiments in a space-station mockup.

**Citation:** Zhang, Q.; Fan, L.; Zhang, Y. Simultaneous Astronaut Accompanying and Visual Navigation in Semi-Structured and Dynamic Intravehicular Environment. *Drones* **2022**, *6*, 397. <https://doi.org/10.3390/drones6120397>

Academic Editors: Daobo Wang and Zain Anwar Ali

Received: 17 October 2022

Accepted: 3 December 2022

Published: 6 December 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** astronaut detection; astronaut accompanying; intravehicular visual navigation; semi-structured environment; dynamic scenes

## 1. Introduction

Human resources in space are scarce and expensive due to launch costs and risks. There is evidence that astronauts will become increasingly physically and cognitively challenged as missions become longer and more varied [1]. The application of artificial intelligence and the use of robotic assistants allow astronauts to focus on more valuable and challenging tasks during both intravehicular and extravehicular activities [2–4]. Up to now, several robotic assistants of various types and functionalities have been developed to improve astronauts' onboard efficiency and help perform regular maintenance tasks such as thermal inspection [5] and on-orbit assembly [6]. These robots include free-flying drones designed to operate in microgravity such as Astrobe [7], Int-Ball [8], CIMON [9], IFPS [10], BIT [11], and more powerful humanoid assistants such as Robonaut2 [12] from NASA and Skybot F-850 [13] proposed by Roscosmos. Although different designs and principles are adopted, robust intravehicular navigation and the ability to work along with human astronauts constitutes the basis for their onboard deployment.

Firstly, to provide immediate service, the robotic assistant should be able to detect and track the served astronaut with high accuracy and efficiency. The recent advances in deep learning have made it possible to solve this problem. Extensive research has been carried out in terms of pedestrian detection [14] and object detection [15,16] by predecessors based on computer-vision techniques. However, the problem of astronaut detection and tracking has some distinctive characteristics due to the particular onboard working environment. On one hand, astronauts can wear similar uniforms and present diverse postures and orientations during intravehicular activities. This can cause problems for general-purpose detectors that are designed and trained for daily life scenes. On the other hand, the relatively fixed and stable background, and the limited range of motion in the space station are beneficial to customizing the astronaut detector. In terms of astronaut motion tracking and prediction, the problem cannot be simply resolved by calibrating intrinsic parameters as with a fixed surveillance camera. The robotic assistant can move and rotate at all times in the space station. Both the movement of the robot and the motion of the served astronauts will change the projected trajectories on the image plane. The trajectories must be decoupled so that the robot can distinguish the actual movement of the served astronauts. Our previous works [17,18] have mainly focused on the astronaut detection and tracking problem from a simplified fixed point of view.

An effective way to decouple motion is to incorporate the robot's 6DoF localization result so that the measured 3D positions of the astronauts can be transformed into the inertial world frame of the space station. Many approaches can be applied to achieve 6DoF localization in the space station. SPHERES [19] is a free-flying research platform propelled by cold-gas thrusters in the International Space Station (ISS). A set of ultrasonic beacons are mounted around the experimental area to provide localization with high efficiency, which resembles a regional satellite navigation system. However, the system can only provide the positioning service within a cubic area of 2 m and may suffer from the problem of signal occlusion and multi-path artifact [20]. The beacon-based approach is more suitable for experiments than service-robot applications. Int-Ball [8] is a spherical camera drone that can record HD videos under remote control currently deployed in the Japanese Experiment Module (JEM). It aims to realize zero photographing time by onboard crew members. Two stereoscopic markers are mounted on the airlock port and the entrance side for in-cabin localization. The accuracy of the marker-based method depends heavily on observation distance. When the robot is far away from the marker, the localization accuracy drops sharply. Moreover, if the robot conducts large-attitude maneuvering, the markers will move out of the robot's vision, and an auxiliary localization system has to take over.

From our perspective, the robotic assistant should not rely on any marker or auxiliary device other than its proprietary sensors for intravehicular navigation. This ideology aims to make the robot's navigation system an independent module and to enhance its adaptability to environmental changes. The space station is an artificial facility with abundant visual clues, which can provide ample references for localization. Astrobees [7] is a new generation of robotic assistants propelled by electric fans in the ISS. It adopts a map-based visual navigation system which does not rely on any external device. An intravehicular map of the ISS is constructed to assist the 6DoF localization of the robot [21]. The team has also studied the impact of light-intensity variations on the map-based navigation system [22]. However, they did not consider the coexistence of human astronauts and the problem of dynamic scenes introduced by various intravehicular activities. These problems are crucial for IRA to work in the manned space station and to provide satisfactory assistance.

To resolve the problem, this paper proposes the framework of simultaneous astronaut-accompanying and in-cabin visual navigation. The semi-structured environment of the space station is utilized to build various registered maps to assist intravehicular localization. Astronauts are detected and tracked in real time with a customized astronaut detector. To enhance the robustness of navigation in dynamic scenes, map matches within the bounding boxes of astronauts are filtered out. The computational workload is evenly distributed within a multi-thread computing architecture so that real-time performance

can be achieved. Based on the robust localization and the customized astronaut detector, a probabilistic model is proposed for astronaut visual tracking and short-term motion prediction, which is crucial for the robot to accompany the served astronaut in the space station and to provide immediate assistance. Table 1 compares our proposed approach and existing methods in the literature. The incorporation of the intravehicular navigation system enables astronaut visual tracking and trajectory prediction from a moving point of view, which is one of the unique contributions of this paper.

**Table 1.** Comparison between our proposed approach and existing methods in the literature.

Robotic Assistant	Navigation Method	Accuracy	Additional Devices	Dynamic Scene	Drawbacks
SPHERES [19]	radio-based	0.5 cm/2.5°	ultrasonic beacons	yes	limited workspace
Astrobee [7]	map-based	5~12 cm/1°~6°	not required	no	for static scene
Int-Ball [8]	marker-based	2 cm/3°	marker	yes	limited field of view
CIMON [9]	vision-based	/	/	/	/
IFPS [10]	map-based	1~2 cm/0.5°	not required	no	for static scene
Robonaut2 [12]	/	/	/	/	/
Skybot F-850 [13]	/	/	/	/	/
Proposed	map-based	1~2 cm/0.5°	not required	yes	/

The rest of this paper is organized as follows. In Section 2, the problem of astronaut detection in diverse postures and orientations is discussed. In Section 3, we focus on the problem of map-based intravehicular navigation in both static and dynamic environments. In Section 4, the astronaut visual tracking and short-term motion prediction model is presented. Experiments to evaluate the overall design and comparative analyses are discussed in Section 5. Finally, we summarize in Section 6.

## 2. Astronaut Detection in Diverse Postures and Orientations

In this section, we address the problem of astronaut detection during intravehicular activities, which is an important component of the overall framework. A lightweight and customized network is designed for astronaut detection in diverse postures and orientations, which achieved superior performance after fine-tuning with a homemade dataset.

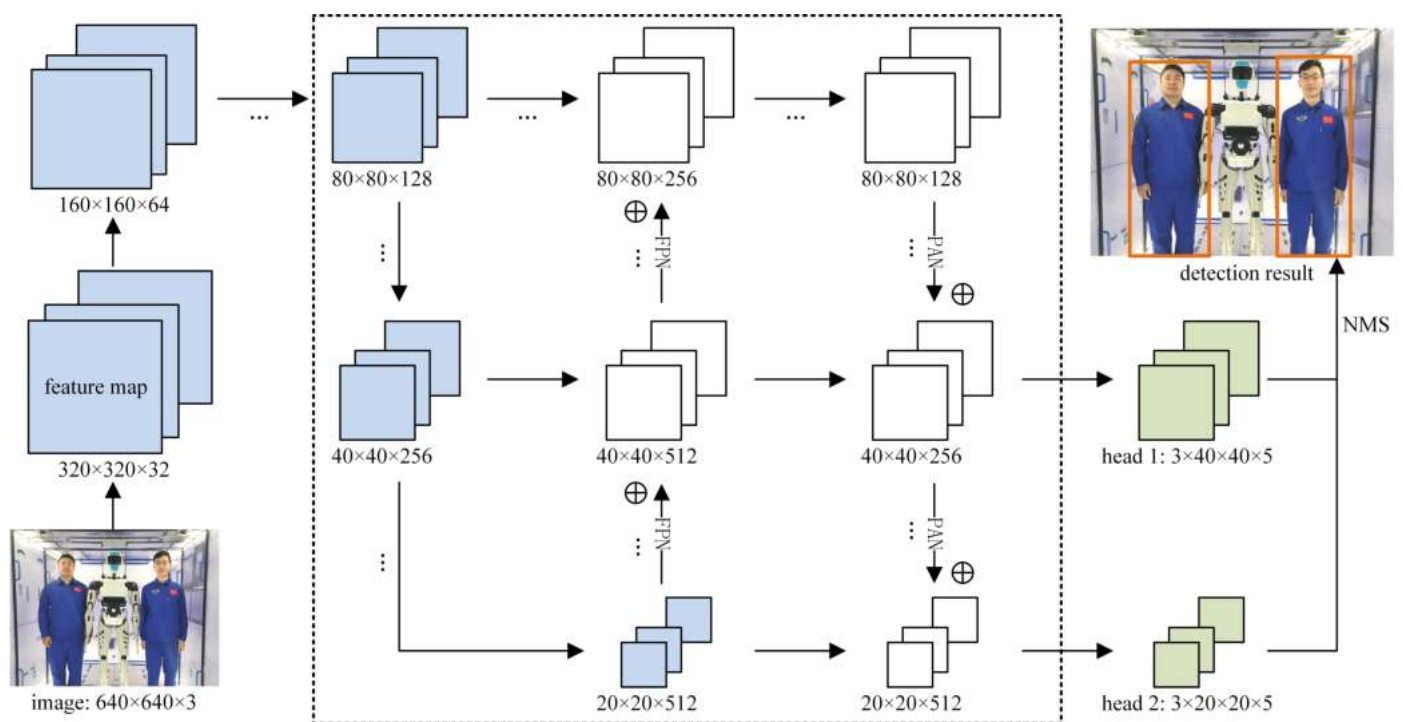
### 2.1. Design of the Customized Astronaut-Detection Network

The special intravehicular working environment has introduced some new features to the astronaut-detection problem, which can be summarized as

- (1) Astronauts can present diverse postures and orientations during intravehicular activities, such as standing upside down and climbing with handrails.
- (2) Astronauts may wear similar uniforms, which are hard to distinguish.
- (3) Images can be taken from any position or orientation by IRA in microgravity.
- (4) It is possible to simplify the astronaut detector while maintaining satisfactory performance by utilizing the relatively fixed and stable background and the limited range of motion in the space station.
- (5) There is a limited number of crew members onboard the space station at the same time.

To achieve satisfactory performance, the astronaut-detection network should be equipped to cope with the above features and be lightweight enough to provide real-time detections. Anchor-based and one-shot object-detection methods [15,23], such as the Yolo network, are widely used in pedestrian detection for their balance between accuracy and efficiency. However, these networks do not perform well in the astronaut-detection task. Many false and missed detections can be found in their results. This poor performance is due to the fact that the structures of those networks are designed for general-purpose applications and the parameters are trained with daily life examples. There lies a gap between the networks' expertise and the actual application scenarios.

To fill the gap, we proposed a lightweight and customized astronaut-detection network based on the anchor-based technique. The main structure of the network is illustrated in Figure 1, where some repetitive layers are collapsed for better understanding. Input to the network is the color image taken by the robot with a resolution of  $640 \times 480$ . Layers in blue are feature-extraction modules characterized by abundant residual blocks [24], which can mitigate the notorious issue of vanishing and exploding gradients. The raw pixels are gradually compressed to the feature maps of  $80 \times 80$ ,  $40 \times 40$ , and  $20 \times 20$ , respectively. Layers in the dashed box apply structures of feature pyramid network (FPN) and path aggregation network (PAN) [25] to accelerate feature fusion in different scales. The residual blocks, FPN, and PAN structures have introduced abundant cross-layer connections, which improves the network's overall fitting capacity. Green layers on the right-hand side are the anchor-based detection heads that output the final detection results after non-maximum suppression.

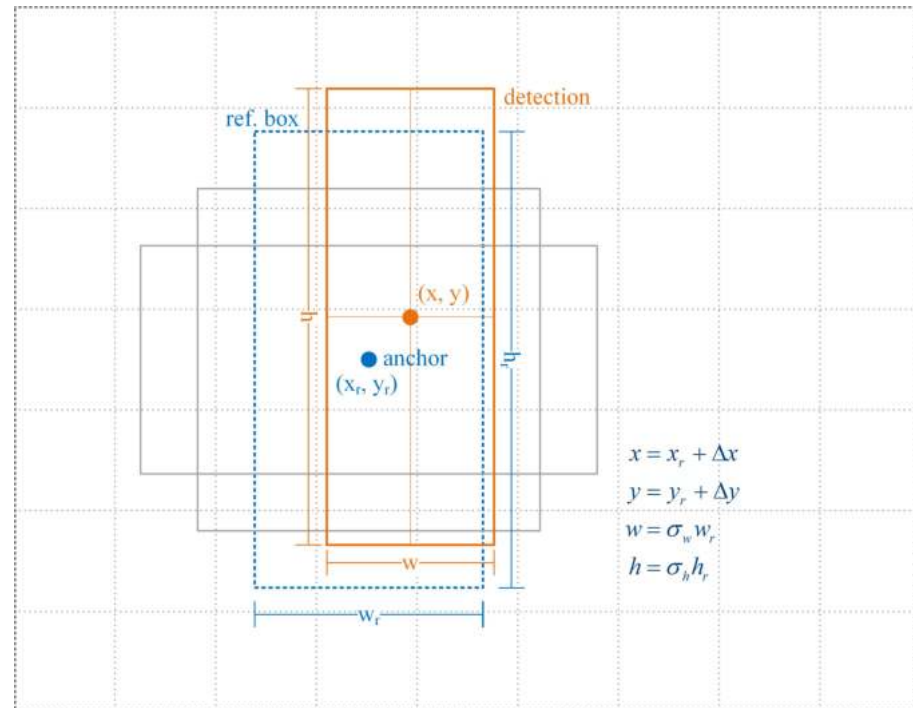


**Figure 1.** Architecture of the lightweight and customized astronaut-detection network. Layers in blue are the feature-extraction modules. Layers in the dashed box are characterized by abundant cross-layer connections for feature fusion. Layers in green are the two anchor-based detection heads.

Considering the limited number of served astronauts and their possible scales on the images, only two detection heads are designed, which also reduces the parameters and improves the network's real-time performance. The detection head with a  $20 \times 20$  grid system is mainly responsible for astronaut detection in proximity, while the other head with a  $40 \times 40$  grid system mainly provides smaller scale detections when astronauts are far away. As shown in Table 2, three reference bounding boxes of different sizes and shapes are designed for each anchor to adapt to the diverse postures and orientations of astronauts during intravehicular activities. A set of correction parameters ( $\Delta x$ ,  $\Delta y$ ,  $\sigma_w$ , and  $\sigma_h$ ) are estimated with respect to the most similar reference boxes to characterize the final detection, as shown in Figure 2. Each reference box also outputs the confidence  $p$  of the detection. The two detection heads provide a total of 6000 reference boxes, which is sufficient to cover all possible scenarios in the space station. To summarize, the astronaut-detection problem is modeled as a regression problem fitted by a lightweight and customized convolutional network with 7.02 million trainable parameters.

**Table 2.** Detection-head specifications of the lightweight and customized astronaut detector.

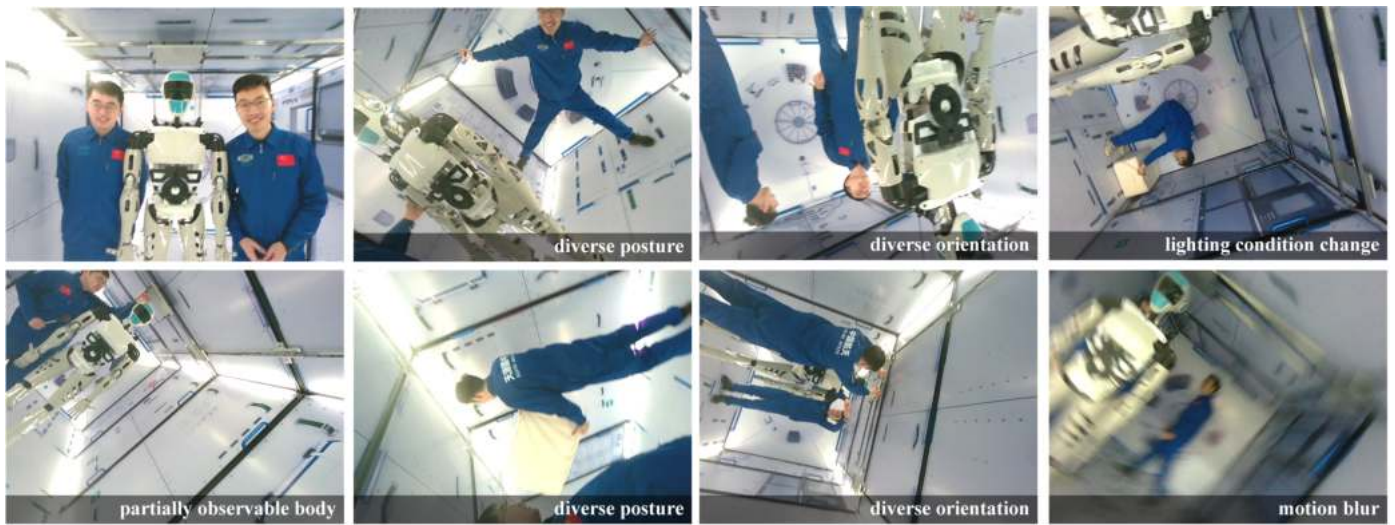
Detection Head	Grid System	Prior Bounding Boxes	Ratio	Predictions for Each Anchor
1	40 × 40	[100, 200]	1/2	$[\Delta x, \Delta y, \sigma_w, \sigma_h, p_i] \times 3$
		[200, 100]	2/1	
		[150, 150]	1/1	
2	20 × 20	[200, 400]	1/2	$[\Delta x, \Delta y, \sigma_w, \sigma_h, p_i] \times 3$
		[400, 200]	2/1	
		[300, 300]	1/1	

**Figure 2.** A set of correction parameters is estimated with respect to the most similar reference boxes to characterize the final detection.

## 2.2. Astronaut-Detection Dataset for Network Fine Tuning

The proposed network cannot maximize its performance before training with an appropriate dataset. General-purpose datasets such as the COCO [26] and the CrowdHuman [27] dataset mismatch the requirements. The incompatibility may be found in the crowdedness of people, the diversity of people's postures and orientations, and the scale of the projections, etc. Even though various data-augmentation techniques can be employed in the training process, it is difficult to mitigate the mismatches between the daily life scenes and the actual working scenarios in the space station.

To address the problem, we built a space-station mockup of high fidelity on the ground, and created a customized dataset for astronaut detection and visual tracking. Volunteers are invited to imitate the intravehicular activities of astronauts in the space-station mockup. During data collection, we constantly moved and rotated the camera so that bodies in the captured images show diverse perspectives. As shown in Figure 3, the proposed dataset incorporated a variety of scenes such as diverse postures and orientations of astronauts, partially observable human bodies, illumination variations, and motion blur. In total, 17,824 labeled images were collected, where 12,000 were used as the training dataset while the remaining 5824 were used as the testing dataset.



**Figure 3.** Examples in the customized astronaut-detection dataset.

### 2.3. Network Pre-Training and Fine Tuning

The astronaut detector is trained in two steps. In the pre-training phase, the network is fed with a cropped COCO dataset for 300 epochs. The cropped dataset is made by discarding crowd labels and labels that are too small or not human from the COCO 2017 dataset. The pre-training process will improve the detector's generalization ability and reduce the risk of over fitting by incorporating large numbers of samples. In the second step, the pretrained model is fine-tuned with the customized astronaut-detection dataset for 100 epochs to obtain the final detector with superior accuracy. The objective function is kept the same in both steps and is formulated as a weighted sum of the confidence loss and the bounding-box regression loss.

$$Loss = \frac{1}{A} \sum_{i=1}^A \sum_{j=1}^G \left( L_{conf}(p_i, \hat{p}_{ij}) + \lambda \hat{c}_{ij} L_{loc}(l_i, \hat{l}_{ij}) \right) \quad (1)$$

where  $L_{conf}(\cdot)$  is the cross-entropy confidence loss,  $L_{loc}(\cdot)$  is the bounding-box regression loss related to the prediction  $l_i$  and the matched target  $\hat{l}_{ij}$  where the CIOU [28] criterion is adopted,  $\hat{c}_{ij}$  is 1 if the match exists,  $\lambda$  is the weight parameter set to 1,  $G$  is the number of ground truth label, and  $A$  is the total number (6000) of reference bounding boxes.

After the two-step training, the proposed detector achieved superior detection accuracy (better than 99%) and recall rate (better than 99%) in the testing dataset, which outperforms the general-purpose detector and the pre-trained detector. Detailed analyses will be discussed in Section 5.

The proposed astronaut detector will play an important role in the robust intravehicular visual navigation in the manned space station to be discussed in Section 3, and support the astronaut visual tracking and motion prediction to be discussed in Section 4.

### 3. Visual Navigation in Semi-Structured and Dynamic Environments

In this section, we focus on the problem of robust visual navigation in the semi-structured and dynamic intravehicular environment, which is the other component of the overall framework. The problem is addressed using a map-based visual navigation technique that does not rely on any marker or additional device. The semi-structured environment makes it unnecessary to use a SLAM-like approach to explore unknown areas, and a map-based method is more practical and reliable. Moreover, compared with possible long-term environmental changes, the ability to cope with instant dynamic factors introduced by various intravehicular activities is more important.

### 3.1. Map-Based Navigation in Semi-Structured Environments

A proprietary RGB-D camera is used as the only sensor for mapping and intravehicular navigation. The RGB-D camera can not only provide color images with rich semantic information, but also the depth value of each pixel, which can improve the perception of distance and eliminate scale uncertainties.

#### (A) Construction of the visual navigation map

In the mapping phase, the RGB-D camera is used to collect a video stream inside the space-station mockup from various positions and orientations. The collected data covered the entire space so that few blind areas are introduced. Based on the video stream, three main steps are utilized to build the final maps for intravehicular navigation.

- (1) Build initial map using standard visual SLAM technique.
- (2) Maps are optimized to minimize the distortion and the overall reprojection error.
- (3) The optimized maps are registered to the space station with a set of known points.

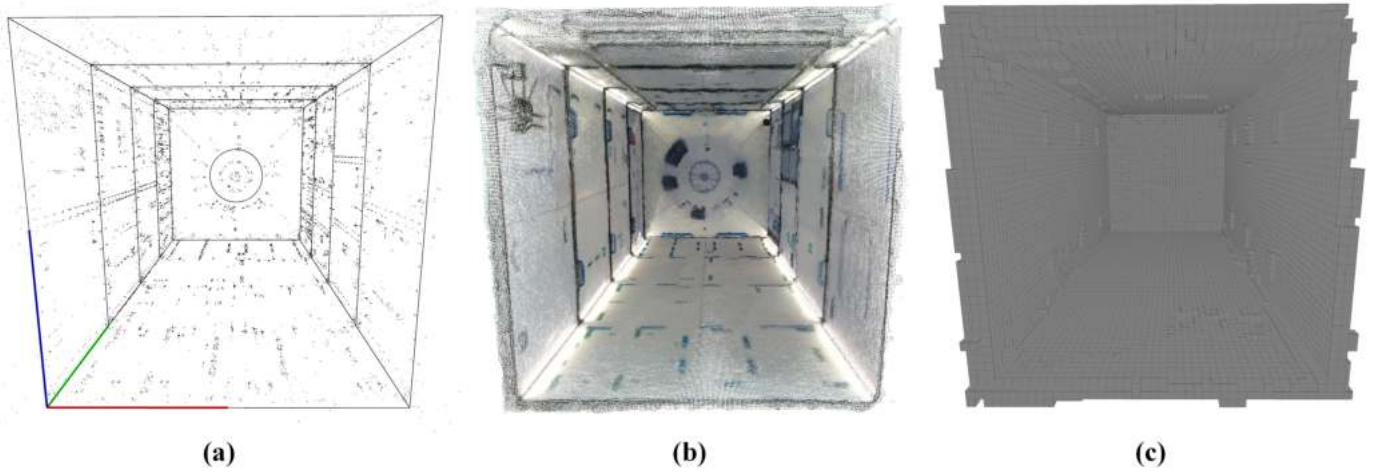
The initial map can be constructed using the Structure from Motion (SfM) technique [29] or standard visual SLAM technique. In our case, a widely used keyframe-based SLAM method [30] is adapted to build the initial point cloud map of the space station. The very first image frame is set as the map's origin temporarily. The point-cloud map contains plenty of distinguishable map points for localization and keyframes to reduce redundancy and assist feature matching. By searching enough associated map points in the current image, the robot can obtain its 6DoF pose with respect to the map.

In the second step, the map is optimized several times to minimize the overall measurement error, so that the map's distortion can be reduced as much as possible, and higher navigation accuracy can be achieved. The optimization problem is summarized as the minimization of reprojection error of associated map points in all keyframes.

$$\{\mathbf{X}^j, \mathbf{R}_k, \mathbf{t}_k\} = \arg \min \sum_{k=1}^K \sum_{j=1}^M \rho \left( c_k^j \left\| \mathbf{x}_k^j - \pi(\mathbf{R}_k \mathbf{X}^j + \mathbf{t}_k) \right\|^2 \right) \quad (2)$$

where  $\mathbf{X}^j$  is the coordinate of the  $j$ th map point,  $\mathbf{R}_k$  and  $\mathbf{t}_k$  are the rotational matrix and translational vector of the  $k$ th keyframe,  $\pi(\cdot)$  is the camera projection function with known intrinsic parameters,  $\mathbf{x}_k^j$  is the pixel coordinate of the matched feature point in the  $k$ th keyframe with respect to the  $j$ th map point, and  $c_k^j$  is 1 if the match exists.  $\rho(\cdot)$  is the robust Huber cost function to reduce the impact of error matches.

The constrained space in the space station allows for minimal distortion of the maps after global optimization as compared with applications in large-scale scenes. In the third step, a set of points with known coordinates are utilized to transform the optimized map to the world frame of the space station. Various types of maps can be constructed accordingly for different purposes such as localization, obstacle avoidance and communication [31]. Figure 4 presents three typical maps registered to the space-station mockup. All maps have an internal dimension of  $2 \times 4 \times 2$  m. The point-cloud map shown in Figure 4a contains, in total, 12,064 map points with distinctive features, and 209 keyframes which are used to accelerate feature matching for pose initialization and re-localization. Figure 4b,c illustrates the dense point-cloud map and the octomap [32] constructed concurrently with the sparse point-cloud map. The clear definition and the straight contours of the mockup in the dense point cloud and the distinguishable handrails in the octomap proved the high accuracy of the maps after global optimization (2), which guarantees the accuracy of the map-based navigation system.



**Figure 4.** Various maps constructed and registered to the space-station mockup. (a) The (sparse) point-cloud map. (b) The dense point-cloud map. (c) Octomap for obstacle avoidance and motion planning.

#### (B) Map-based localization and orientation

With prebuilt maps, two steps are carried out for intravehicular localization and orientation. In the first step, the robot tries to obtain an initial estimate of its 6DoF pose from scratch. This is achieved by comparing the current image with each similar keyframe in the sparse point-cloud map. Initial pose will be recovered using a PnP solver when enough 2D–3D matches are associated. With an initial pose estimation, local map points are then projected to the current image to search more 2D–3D matches for pose-only optimization, which will provide a more accurate localization result. The pose-only optimization problem can be summarized as the minimization of reprojection error with a static map.

$$\{\mathbf{R}, \mathbf{t}\} = \arg \min \sum_{j=1}^M \rho \left( c^j \left\| \mathbf{x}^j - \pi(\mathbf{R}\mathbf{X}^j + \mathbf{t}) \right\|^2 \right) \quad (3)$$

where  $\mathbf{R}$  and  $\mathbf{t}$  are the rotational matrix and translational vector of the robot with respect to the world frame,  $\mathbf{x}^j$  is the pixel coordinate of the matched feature point with respect to the  $j$ th map point, and  $c^j$  is 1 if the match exists.

When the robot succeeds to localize itself for several consecutive frames after initialization or re-localization, frame-to-frame velocity is utilized to provide the initial guess to search map points, which saves time and helps improve computational efficiency.

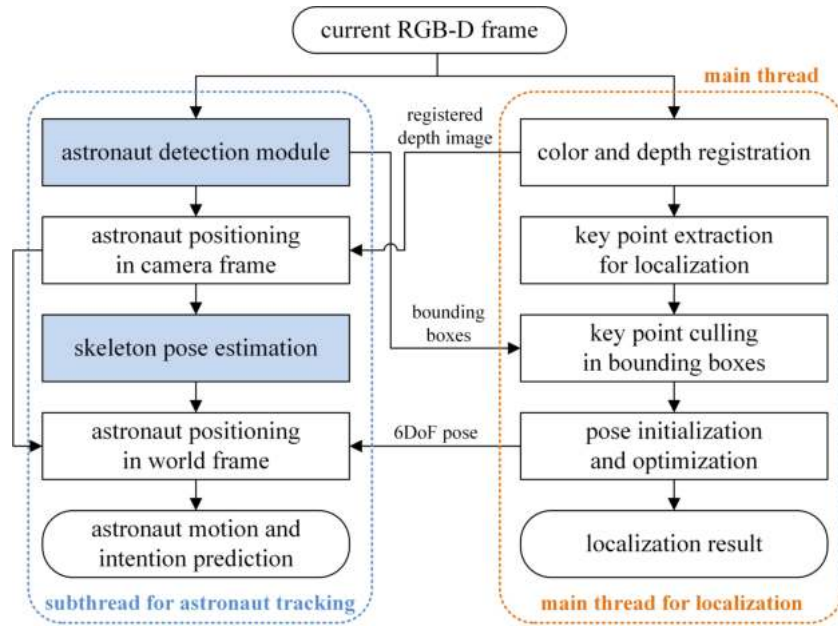
#### 3.2. Robust Navigation during Human–Robot Collaboration

The robotic assistant usually works side by side with human astronauts to provide immediate service. In the constrained intravehicular environment, astronauts can take a field of vision in front of the robot. Astronauts' various intravehicular activities can also occlude the map points and introduce dynamic disturbance to the navigation system. In such a condition, the robot may be confused to search enough map points for stable in-cabin localization. The poor localization will, in turn, create uncertainties for the robot to accomplish various onboard tasks.

To address the problem, we proposed the integrated framework of simultaneous astronaut accompanying and visual navigation in the dynamic and semi-structured intravehicular environment. The framework can not only solve the problem of robust navigation in dynamic scenes during human–robot collaboration, but also assist in tracking and predicting the short-motion of the served astronaut to provide more satisfactory and foresighted assistance.

As shown in Figure 5, the framework adopts a multi-thread computing architecture to ensure real-time performance. The main thread in the red dashed box is mainly responsible

for image pre-processing and in-cabin visual navigation, whereas the sub thread in the blue dashed box is mainly responsible for astronaut detection, visual tracking, and trajectory estimation. Specifically, while the main thread is working on frame registration and feature extraction, the sub thread tries to detect astronauts in the meantime. The first-round information exchange between the two threads is carried out at this point. Then, feature points within the detected bounding boxes are filtered out to avoid large areas of disturbances to the visual navigation system.



**Figure 5.** Computational work flow of the integrated framework of simultaneously astronaut accompanying and visual navigation in the semi-structured and dynamic intravehicular environment.

While the main thread is working on 6DoF pose initialization and optimization, the sub thread is idle and can perform some computations such as astronaut skeleton extraction. The second-round information exchange is carried out once the main thread has obtained the localization result. The optimized 6DoF pose together with the detected bounding boxes are utilized for astronaut visual tracking and motion prediction in the sub thread, which will be discussed in Section 4.

The computational burden of the proposed framework is evenly distributed where the main thread uses mainly CPU resources and the sub thread consumes mainly GPU resources. The overall algorithm is tested to run at over 30 Hz with a GS66 laptop (low-power i9@2.4GHz processor and RTX 2080 GPU for notebook).

#### 4. Astronaut Visual Tracking and Motion Prediction

Astronaut visual tracking and motion prediction help the robot track and identify the served astronaut and provide immediate assistance when required. The solution to the problem is based on the research into astronaut detection in Section 2 and the research into robust intravehicular navigation in Section 3.

Specifically, the astronaut visual-tracking problem is to detect and track the movement of a certain target astronaut in a sequence of images, which is formulated as a maximum a-posteriori (MAP) estimation problem as

$$i = \arg \max P^k(p | \beta_i^k, \beta_i^{k-1}), i = 1, 2, \dots, M \quad (4)$$

where  $M$  is the number of detected astronauts in the current (or  $k$ th) frame;  $\beta_i^k$  and  $\beta_i^{k-1}$  are the  $i$ th bounding box in the current frame and the target to be matched in the previous

frame, respectively; and  $P^k(\cdot | \beta_i^k, \beta_t^{k-1})$  defines the probability of the match. We seek to find the bounding box with the largest posterior probability. If no bounding box is matched for a long time or the wrong bounding box is selected, the tracking task fails.

The posterior probability in Equation (4) is determined by a variety of factors. For example, when the 3D position of  $\beta_i^k$  is close to the predicted trajectory of the served astronaut, or the bounding boxes overlap, there is a high match probability. On the contrary, when the 3D position of  $\beta_i^k$  deviates from the predicted trajectory or the geometry mismatches, the probability will be small. According to the above factors, the overall posterior probability is decomposed into the trajectory correlation probability  $P_{\text{predicton}}^k$ , and the geometric correlation probability  $P_{\text{geometry}}^k$  and other clues  $P_{\text{others}}^k$  include identity identification probability.

$$\begin{aligned} i &= \arg \max P^k(p | \beta_i^k, \beta_t^{k-1}) \\ &= \arg \max P_{\text{predicton}}^k(p | \beta_i^k) P_{\text{geometry}}^k(p | \beta_i^k, \beta_t^{k-1}) P_{\text{others}}^k(p | \beta_i^k, \beta_t^{k-1}) \end{aligned} \quad (5)$$

#### (A) Matching with predicted trajectory

The served astronaut's trajectory can be estimated and predicted using the astronaut detection result in the image flow and the robot's 6DoF localization information.

Firstly, the 3D position of the astronaut in the robot body frame  $[x_b, y_b, z_b]^T$  is obtained using the camera's intrinsic parameters. The coordinates are averaged over a set of points within a small central area in the bounding box to reduce the measurement error. Then, by incorporating the 6DoF pose  $\{\mathbf{R}, \mathbf{t}\}$  of the robot (3), the astronaut's 3D position can be transformed to be represented in the world frame of the space station  $[x_w, y_w, z_w]^T$ .

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \mathbf{R}^T \left( \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} - \mathbf{t} \right) \quad (6)$$

The space station usually keeps three axes stabilized to the earth and orbits every 1.5 h. We assume the space station to be an inertial system when modeling the instant motion of astronauts. The motion is formulated as a constant acceleration model for simplicity. For example, when the astronaut moves freely in microgravity, a constant speed can be estimated and the acceleration is zero. When the astronaut contacts with the surroundings, a time-varying acceleration can be estimated by introducing a relatively large acceleration noise in the model. The above motion model and corresponding measurement model are defined as

$$\begin{aligned} \mathbf{x}_w^k &= A\mathbf{x}_w^{k-1} + \mathbf{w} \\ \mathbf{z}^k &= H\mathbf{x}_w^k + \mathbf{v} \end{aligned} \quad (7)$$

where  $A \in \mathbb{R}^{9 \times 9}$  is the state transition matrix that determines the relationship between the current state  $\mathbf{x}_w^k \in \mathbb{R}^9$  and the previous state  $\mathbf{x}_w^{k-1}$ , vector  $\mathbf{z}^k$  is the measured 3D position of the astronaut represented in the world frame,  $H \in \mathbb{R}^{3 \times 9}$  is the measurement matrix,  $\mathbf{w}$  is the time-invariant process noise to characterize the error of the simplified motion model, and  $\mathbf{v}$  is the time-invariant measurement noise determined by the positioning accuracy of the served astronaut. The process and measurement noise are assumed to be white Gaussian with zero means and covariance matrices of  $Q$  and  $R$ , respectively.

The nine-dimensional state vector contains the estimated position, velocity, and acceleration of the served astronaut represented in the world frame as

$$\mathbf{x}_w^k = \begin{bmatrix} x_w^k & y_w^k & z_w^k & v_{x,w}^k & v_{y,w}^k & v_{z,w}^k & a_{x,w}^k & a_{y,w}^k & a_{z,w}^k \end{bmatrix} \in \mathbb{R}^9 \quad (8)$$

The trajectory of the served astronaut can be predicted with the above constant acceleration model. The update interval is kept the same as the frequency of the overall astronaut-detection and visual-navigation framework at 30 Hz.

$$\begin{aligned} \mathbf{x}_w^{k-} &= A\mathbf{x}_w^{k-1} \\ P^{k-} &= AP^{k-1}A^T + Q \end{aligned} \quad (9)$$

where  $P$  is the state covariance matrix. We propagate Equation (9) for a few seconds to predict the short-term motion of the served astronaut.

With estimated trajectories, a comparison is made between the prediction and each bounding box in the current image frame. There would be a high correlation probability if the 3D position of a certain bounding box is close to the predicted trajectory of the target astronaut. The trajectory correlation probability is defined as

$$P_{\text{predicton}}^k \left( p \mid \beta_i^k \right) = e^{-\alpha_0 \|z_i^k - \mathbf{x}_w^{k-}(1:3)\|} \quad (10)$$

where  $z_i^k$  is the measured position of the astronaut in the  $i$ th bounding box and  $\alpha_0$  is a non-negative parameter.

Once the match is verified together with other criteria, the measurement will be used to correct the motion model of the target astronaut.

$$\begin{aligned} K^k &= P^{k-} H^T (H P^{k-} H^T + R)^{-1} \\ \mathbf{x}_w^k &= \mathbf{x}_w^{k-} + K^k (z^k - H \mathbf{x}_w^{k-}) \\ P^k &= (I - K^k H) P^{k-} \end{aligned} \quad (11)$$

#### (B) Matching with geometric similarity

Besides trajectory correlation, the geometric similarity of the bounding boxes can also provide valuable information for visual tracking. The overall algorithm runs at 30 Hz, and, thus, we assume few changes between consecutive frames to be introduced. Many criteria can be used to characterize the similarity between bounding boxes. We selected the most straightforward IOU (intersection over union) criterion. When a certain bounding box in the current image frame overlaps heavily with the target in the previous frame, there would be a high matching probability. The geometric correlation probability is defined as

$$P_{\text{geometry}}^k \left( p \mid \beta_i^k, \beta_i^{k-1} \right) = e^{-\alpha_1 (1-\text{IOU})} \quad (12)$$

where  $\alpha_1$  is a non-negative parameter, and a larger  $\alpha_1$  will give more weight to the IOU criterion.

#### (C) Matching with other clues

Some other clues can also be incorporated to assist astronaut visual tracking. For example, face recognition is helpful for initial identity confirmation and tracking recovery after long-time loss. The corresponding posterior probability is formulated as

$$P_{\text{others}}^k \left( p \mid \beta_i^k, \beta_i^{k-1} \right) = \begin{cases} 1.0, & \text{matched} \\ 0.5, & \text{not sure} \\ 0.0, & \text{not matched} \end{cases} \quad (13)$$

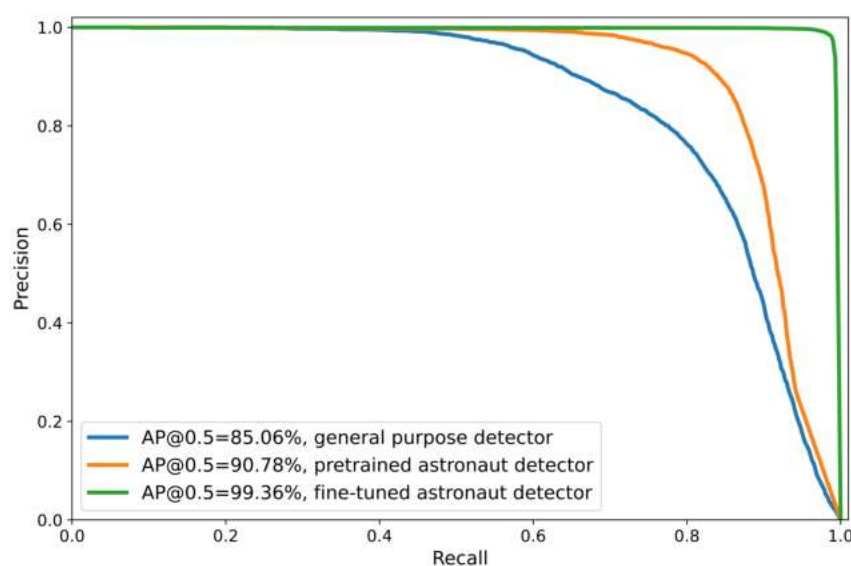
During the experiments, we only applied the trajectory and geometric correlation probabilities into the framework. The face-recognition part is out of the scope of this paper, and can be referenced from our previous work [18].

## 5. Experimental Results and Discussion

Experiments were carried out to evaluate each component of the proposed framework in Section 5.1 (astronaut detection) and Section 5.2 (visual navigation) respectively. The overall performance is verified and discussed in Section 5.3.

### 5.1. Evaluation of the Customized Astronaut Detector

The performance of the proposed astronaut-detection network is evaluated in the testing dataset (5824 images) collected in the space-station mockup. As shown in Figure 6, the fine-tuned detector shows an AP@0.5 of 99.36%, which outperforms the general-purpose detection network (85.06%) and the pretrained detector (90.78%). The superior performance of the astronaut detector benefited from the customized network structure designed for in-travehicular applications and the proposed astronaut-detection dataset to mitigate possible domain inconsistency.



**Figure 6.** Comparison of the precision-recall curves of three detectors in the task of astronaut detection in the space-station mockup.

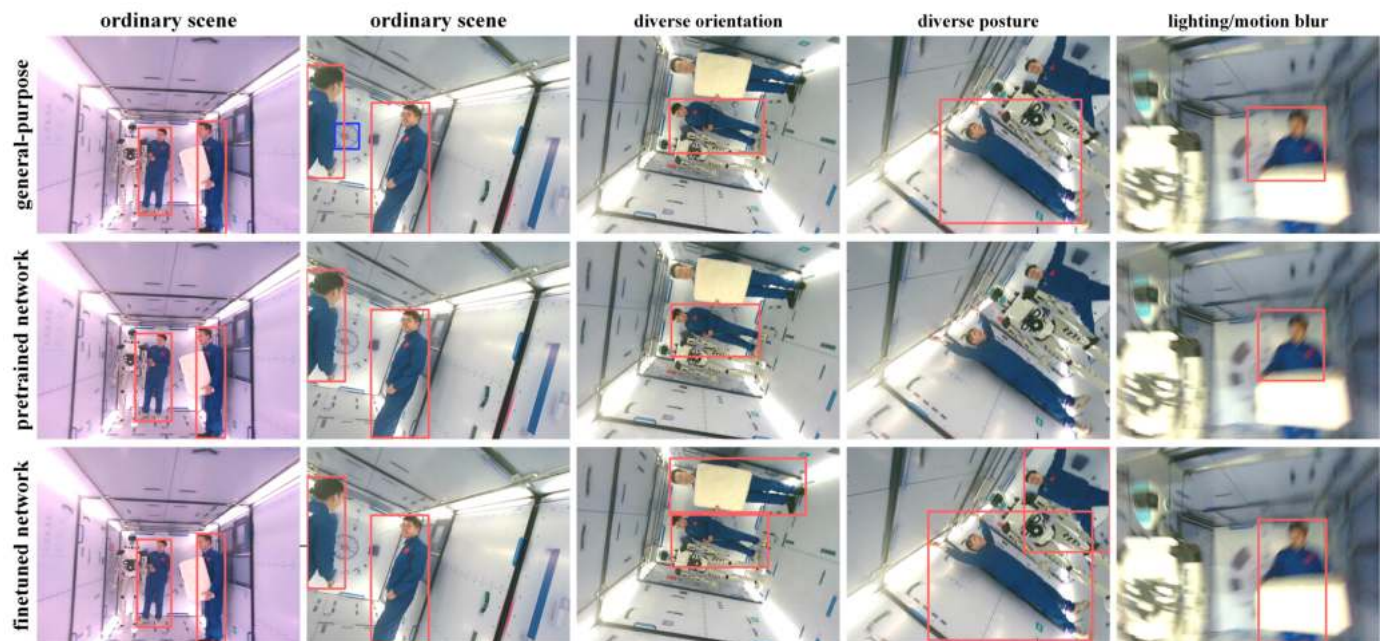
Figure 7 presents some typical results for comparative studies. All three networks achieved satisfactory detection when volunteers are close to the upright posture. The general-purpose network may give some false bounding boxes that do not exist in the mockup, such as a clock. When astronauts' body postures or orientations are significantly different from daily life scenes on the ground, both the general-purpose detector and the pretrained detector degrade. A large number of missed detections and poor detections can be found. On the other hand, the fine-tuned astronaut detector still guarantees its performance when dealing with the challenging task. It is worth mentioning that all networks showed satisfactory performance to cope with illumination variation and motion blur without implementing image-enhancement algorithms [33].

The proposed astronaut detector showed superior performance to cope with the rich body postures and orientations. The estimated pixel coordinates of the bounding boxes are also more accurate than its competitors. The proposed detector runs at over 80 Hz on a GS66 laptop, proving the sufficiency for real-time performance.

### 5.2. Evaluation of Map-Based Navigation in Semi-Structured and Dynamic Environments

Experiments were conducted in the mockup to test the accuracy and robustness of the proposed map-based navigation system in both static and dynamic scenarios. As shown in Figure 8a, the mockup has an internal dimension of  $2 \times 4 \times 2$  m and has high fidelity to a

real space station. The handrails, buttons, experiment cabinets, and airlock provided stable visual references for visual navigation.



**Figure 7.** Astronaut-detection performance of the general-purpose network, pretrained network and the fine-tuned astronaut detector on the testing dataset.

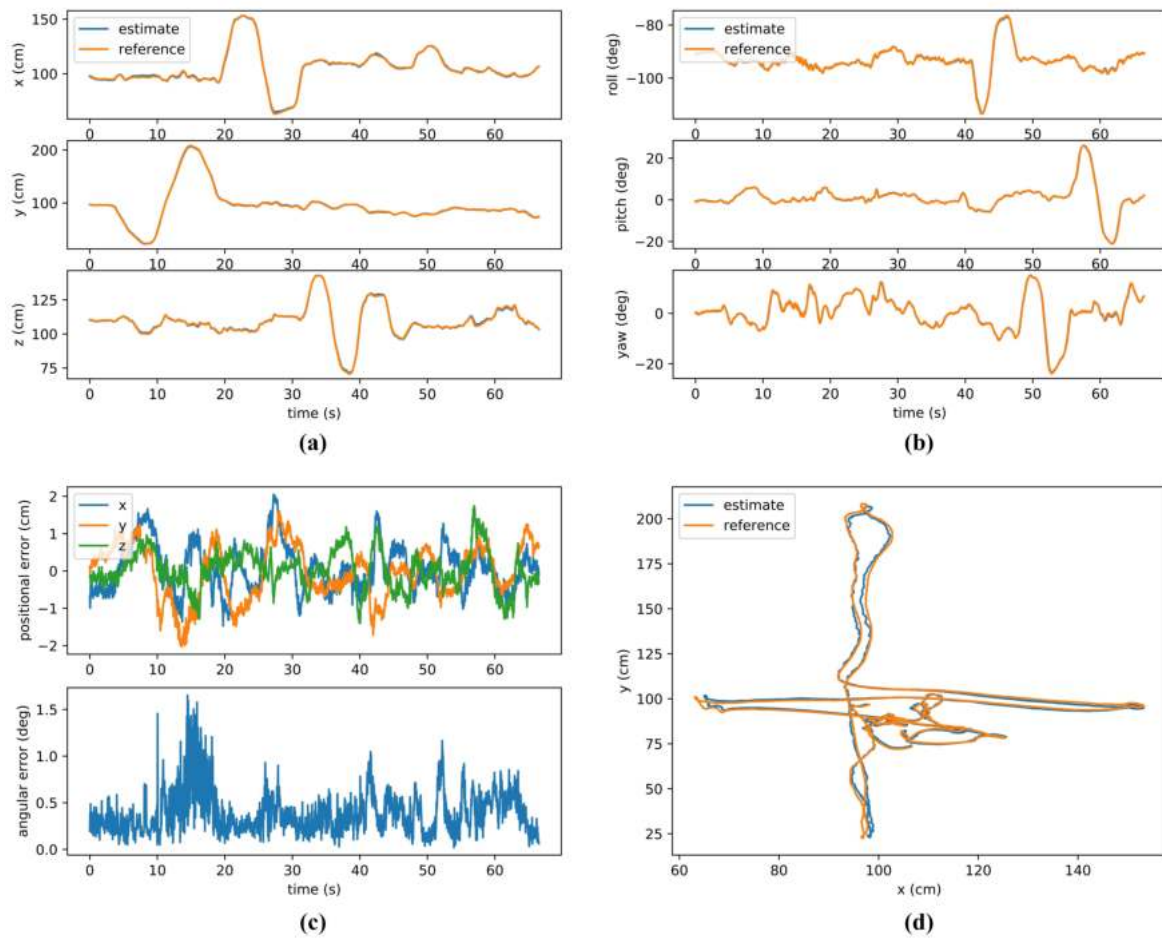


**Figure 8.** The ground experimental environment. (a) The space-station mockup of high fidelity. (b) The humanoid robotic assistant Taikobot used in the experiment.

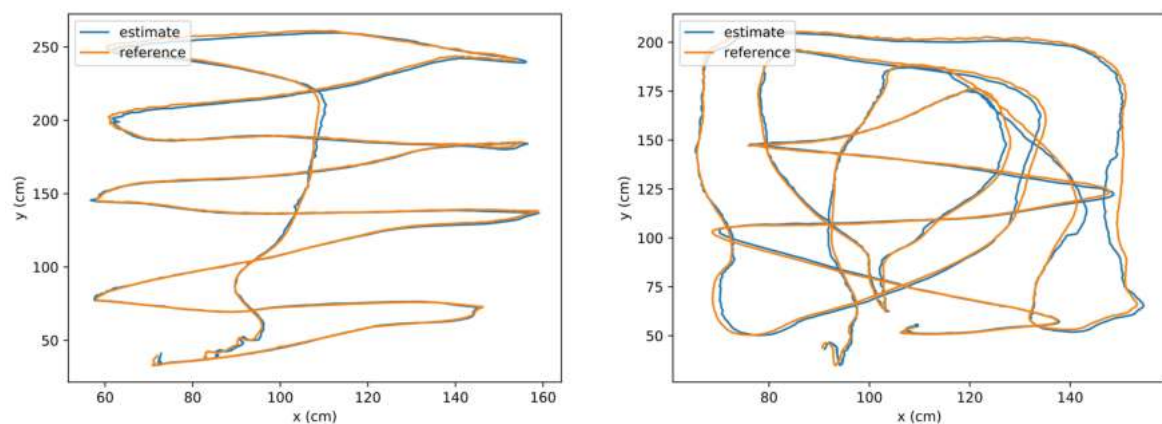
#### (A) Performance in static environment

During the experiment, the RGB-D camera is moved and rotated constantly to collect video streams in the mockup. Four large ( $60 \times 60$  cm) Aruco markers [34] are fixed to the back of the camera to provide reference trajectories for comparison. Figure 9 presents the results in a static environment. As shown in Figure 9a,b, we performed a large range of motion in all six translational and rotational directions consecutively. The estimated 6DoF pose almost coincides with the reference trajectories. Figure 9c presents the corresponding error curves. The average positional error is less than 1cm (the maximum error does not exceed 2 cm) and the average three-axis angular error is less than  $0.5^\circ$ . The camera's overall trajectory during the experiment is shown in Figure 9d. Two other random trajectories

were also collected and analyzed as shown in Figure 10 where identical performance was achieved, proving the feasibility of the proposed navigation method.



**Figure 9.** Localization and orientation performance of the proposed map-based navigation system in static environment. (a) Position curves in world frame. (b) Euler angle curves with respect to the world frame. (c) Positional error and three-axis angular error. (d) The estimated trajectories in the XY plane of the space-station mockup.

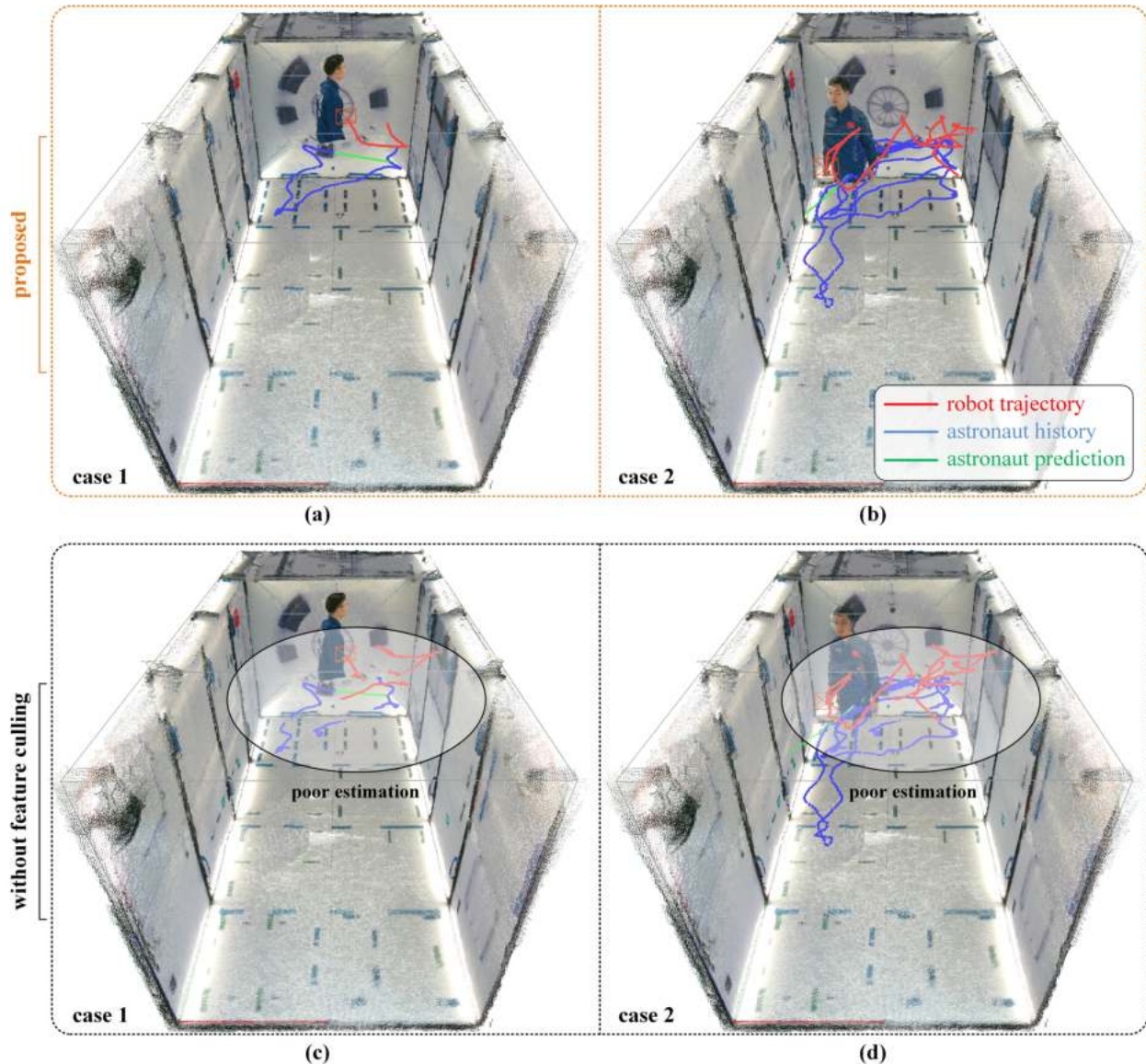


**Figure 10.** Two random trajectories tested in the space-station mockup (static environment).

#### (B) Performance in dynamic environment

Next, we evaluate the map-based navigation in dynamic scenes when the robot works along with human astronauts. As shown in Figure 8b, the humanoid robotic assistant

Taikobot [35] we developed previously is used this time. The RGB-D camera mounted in the head of Taikobot is used both for astronaut detection and intravehicular navigation. During the experiment, the robot moves along with a volunteer astronaut in the mockup to provide immediate assistance. The astronaut can occasionally require a large field of vision in front of the robot during intravehicular activities. As shown in Figure 11a,b, the robot navigates robustly and smoothly in the dynamic environment with the proposed framework. Based on the stable localization result of the robot, the trajectories of the served astronaut are also estimated and predicted in the meantime, which will be discussed in Section 5.3.



**Figure 11.** Localization performance of the map-based navigation system in dynamic environment. Red lines are the estimated trajectories of the robotic assistant. Blue and green lines are the estimated and predicted trajectories of the served astronaut, respectively. (a,b) performance of the proposed framework. (c,d) performance without feature culling.

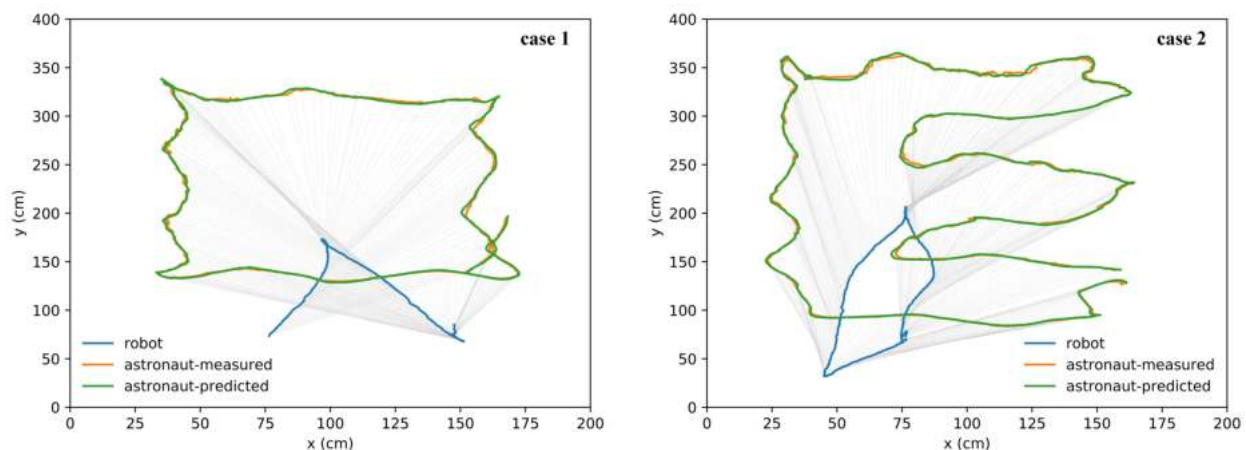
By comparison, when we remove the feature-culling module in the framework, the robot becomes lost several times with the same data input as shown in Figure 11c,d. The degradation in the navigation system is caused by the dynamic feature points detected on the served astronauts, which makes it difficult for the robot to locate sufficient references

for stable in-cabin navigation. As we can see, the poor localization result also led to poor trajectory estimation of the astronaut.

### 5.3. Verification of Simultaneous Astronaut Accompanying and Visual Navigation

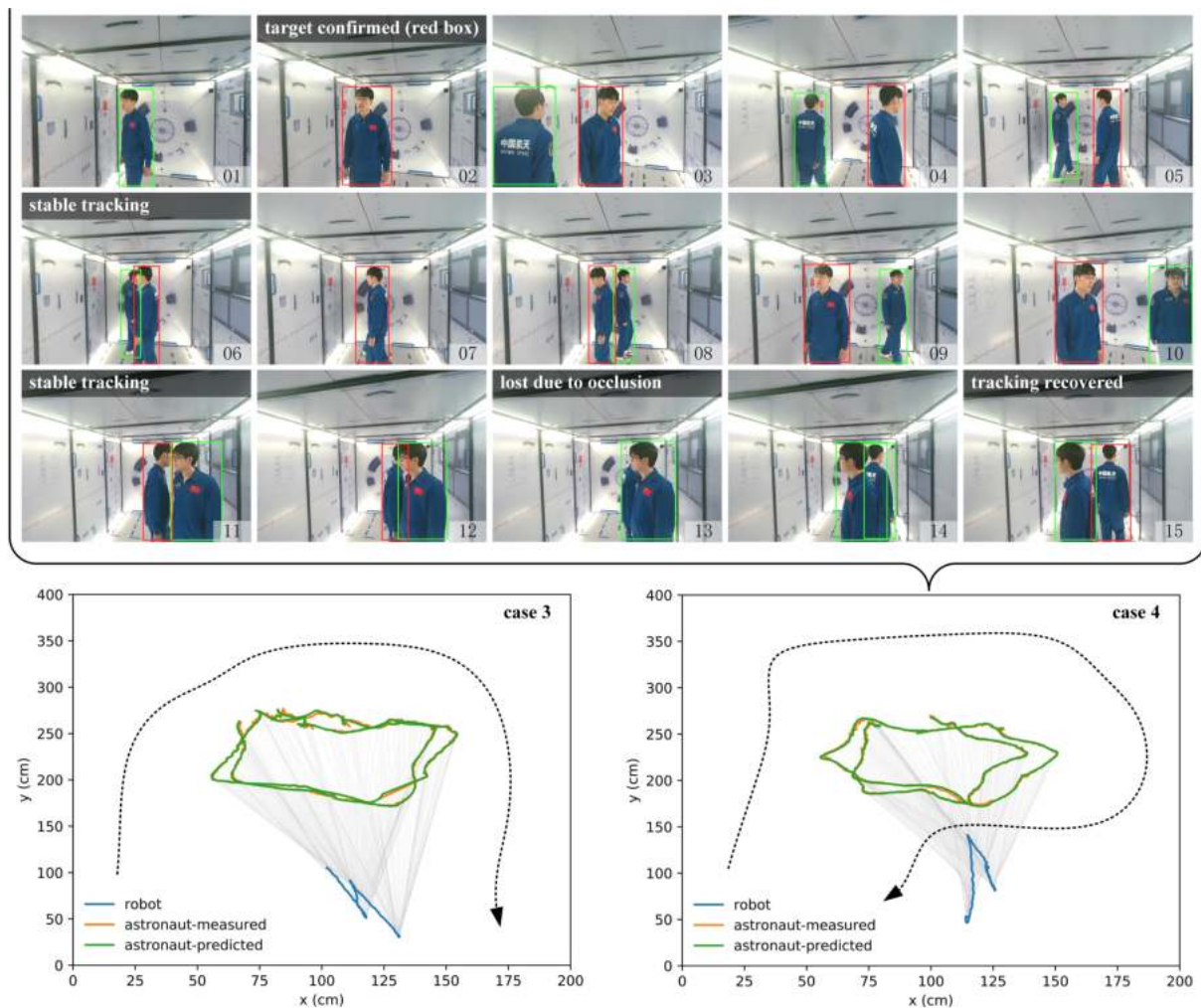
Based on the robust intravehicular navigation system and the customized astronaut detector, the trajectory of the served astronaut can be identified, estimated and predicted efficiently.

Firstly, we present the results when only one astronaut is served. Figure 12 gives two typical scenarios where the robot moves along with one astronaut in the mockup. The red and green curves are the measured and predicted trajectories of the served astronaut, respectively. The blue curves are the estimated trajectories of the robot by (3). During the experiments, the astronaut was kept within the robot's perspective. In both scenarios, the robot can navigate smoothly in the dynamic scenes, and the astronaut is tracked stably in the image flow at all times. The predicted trajectories of the astronaut are identical to the measurements. By applying the proposed motion model, the predictions are also smoothed compared with the raw measurements.



**Figure 12.** Experimental results of simultaneous astronaut tracking and visual navigation when the robotic assistant accompanies one astronaut.

When multiple astronauts coexist, the robot is able to track a certain astronaut to provide a customized service. The task is more challenging compared with the previous examples. Figure 13 presents the results of two typical scenarios where the robot works along with two astronauts at the same time. We take case 4 to discuss in detail. As shown in the picture series in Figure 13, when the robot has confirmed (red bounding box) the target astronaut (astronaut A), the other astronaut (astronaut B) enters in the field of view of the robot. The robot can distinguish the target astronaut from astronaut B by utilizing the trajectory correlation and geometric similarity criteria (5). The robot tracks the target astronaut robustly even though the two astronauts move closely and overlap. The most challenging part occurs when astronaut B moves in between the robot and the target astronaut. When astronaut A is completely obscured from the robot, tracking loss is inevitable. However, when astronaut A reappears in the image, the robot recovers the tracking immediately. It is worth mentioning that only the trajectory and geometry criteria are used in the tracking process, which have minimal computing burden. Other criteria such as face recognition can also be incorporated into the framework for tracking recovery after long-time loss.



**Figure 13.** Experimental results of simultaneous astronaut tracking and visual navigation when multiple astronauts coexist in the space-station mockup. The red bounding boxes in the sequentially numbered pictures denote the target astronaut. The dotted curves in the two sketches denote the routes of astronaut B.

## 6. Conclusions

This paper proposed the framework of simultaneous astronaut accompanying and visual navigation in the semi-structured and dynamic intravehicular environment. In terms of the intravehicular navigation problem of IRA, the proposed map-based visual-navigation framework is able to provide real-time and accurate 6DoF localization results even in dynamic scenes during human–robot interaction. Moreover, compared with the other map-based localization methods of IRA in the literature, we achieved superior accuracy (1–2 cm, 0.5°). In terms of the astronaut visual tracking and short-term motion-prediction problem, the proposed MAP model with geometric similarity and trajectory correlation hints enables IRA to distinguish and accompany the served astronaut with minimal calculation from a moving point of view. The overall framework provided a feasible solution to address the problem of intravehicular robotic navigation and astronaut–robot coordination in the manned and constrained space station.

**Author Contributions:** Conceptualization, Q.Z. and Y.Z.; methodology, Q.Z. and Y.Z.; software, Q.Z.; validation, Q.Z.; formal analysis, Q.Z.; investigation, Q.Z.; resources, L.F. and Y.Z.; data curation, Q.Z. and L.F.; writing—original draft preparation, Q.Z.; writing—review and editing, L.F. and Y.Z.; visualization, Q.Z. and L.F.; supervision, L.F. and Y.Z.; project administration, L.F. and Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by Huzhou Institute of Zhejiang University under the Huzhou Distinguished Scholar Program (ZJHI—KY0016).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to intellectual-property protection.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

IRA	Intravehicular robotic assistants
CIMON	Crew interactive mobile companion
IFPS	Intelligent formation personal satellite
SPHERES	Synchronized position hold engage and reorient experimental satellite
ISS	International Space Station
JEM	Japanese experiment module
FPN	Feature pyramid network
PAN	Path aggregation network
IOU	Intersection over union
CIOU	Complete intersection over union
COCO	Common object in context
RGB-D	Red green blue-depth
SFM	Structure from motion
SLAM	Simultaneous localization and mapping
PnP	Perspective-n-point
AP	Average precision
MAP	Maximum a posteriori

## References

1. Sgobba, T.; Kanki, B.; Clervoy, J.F. *Space Safety and Human Performance*, 1st ed.; Butterworth-Heinemann: Oxford, UK, 2018; pp. 357–376. Available online: <https://www.elsevier.com/books/space-safety-and-human-performance/sgobba/978-0-08-101869-9> (accessed on 10 October 2022).
2. Russo, A.; Lax, G. Using artificial intelligence for space challenges: A survey. *Appl. Sci.* **2022**, *12*, 5106. [CrossRef]
3. Miller, M.J.; McGuire, K.M.; Feigh, K.M. Information flow model of human extravehicular activity operations. In Proceedings of the 2015 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2015.
4. Miller, M.J. Decision support system development for human extravehicular activity. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2017.
5. Akbulut, M.; Ertas, A.H. Establishing reduced thermal mathematical model (RTMM) for a space equipment: An integrative review. *Aircr. Eng. Aerosp. Technol.* **2022**, *94*, 1009–1018. [CrossRef]
6. Li, D.; Zhong, L.; Zhu, W.; Xu, Z.; Tang, Q.; Zhan, W. A survey of space robotic technologies for on-Orbit assembly. *Space Sci. Technol.* **2022**, *2022*, 9849170. [CrossRef]
7. Smith, T.; Barlow, J.; Bualat, M. Astrobe: A new platform for free-flying robotics on the international space station. In Proceedings of the 13th International Symposium on Artificial Intelligence, Robotics, and Automation in Space, Beijing, China, 20–22 June 2016.
8. Mitani, S.; Goto, M.; Konomura, R. Int-ball: Crew-supportive autonomous mobile camera robot on ISS/JEM. In Proceedings of the 2019 IEEE Aerospace Conference, Yellowstone Conference Center, Big Sky, MT, USA, 2–9 March 2019.
9. Experiment CIMON—Astronaut Assistance System. Available online: <https://www.dlr.de/content/en/articles/missions-projects/horizons/experimente-horizons-cimon.html> (accessed on 10 October 2022).
10. Zhang, R.; Wang, Z.K.; Zhang, Y.L. A person-following nanosatellite for in-cabin astronaut assistance: System design and deep-learning-based astronaut visual tracking implementation. *Acta Astronaut.* **2019**, *162*, 121–134. [CrossRef]
11. Liu, Y.Q.; Li, L.; Ceccarelli, M.; Li, H.; Huang, Q.; Wang, X. Design and testing of BIT flying robot. In Proceedings of the 23rd CISM IFToMM Symposium, Online, 20–24 September 2020. Available online: [http://doi.org/10.1007/978-3-030-58380-4\\_9](http://doi.org/10.1007/978-3-030-58380-4_9) (accessed on 10 October 2022).
12. NASA Facts Robonaut 2, Technical Report. Available online: [https://www.nasa.gov/sites/default/files/files/Robonaut2\\_508.pdf](https://www.nasa.gov/sites/default/files/files/Robonaut2_508.pdf) (accessed on 10 October 2022).
13. Meet Skybot F-850, the Humanoid Robot Russia Is Launching into Space. Available online: <https://www.space.com/russia-launching-humanoid-robot-into-space.html> (accessed on 10 October 2022).

14. Chen, L.; Lin, S.; Lu, X.; Cao, D.; Wu, H.; Guo, C.; Liu, C.; Wang, F. Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3234–3246. [CrossRef]
15. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
16. Avdelidis, N.P.; Tsourdos, A.; Lafiosca, P.; Plaster, R.; Plaster, A.; Droznika, M. Defects recognition algorithm development from visual UAV inspections. *Sensors* **2022**, *22*, 4682. [CrossRef] [PubMed]
17. Zhang, R.; Wang, Z.K.; Zhang, Y.L. Astronaut visual tracking of flying assistant robot in space station based on deep learning and probabilistic model. *Int. J. Aerosp. Eng.* **2018**, *2018*, 6357185. [CrossRef]
18. Zhang, R.; Zhang, Y.L.; Zhang, X.Y. Tracking in-cabin astronauts Using deep learning and head motion clues. *Int. J. Aerosp. Eng.* **2018**, *9*, 2680–2693. [CrossRef]
19. Saenz-Otero, A.; Miller, D.W. Initial SPHERES operations aboard the International Space Station. In Proceedings of the 6th IAA Symposium on Small Satellites for Earth Observation, Berlin, Germany, 23–26 April 2008.
20. Prochniewicz, D.; Grzymala, M. Analysis of the impact of multipath on Galileo system measurements. *Remote Sens.* **2021**, *13*, 2295. [CrossRef]
21. Coltin, B.; Fusco, J.; Moratto, Z.; Alexandrov, O.; Nakamura, R. Localization from visual landmarks on a free-flying robot. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, Seoul, Republic of Korea, 8–9 October 2016.
22. Kim, P.; Coltin, B.; Alexandrov, O. Robust visual localization in changing lighting conditions. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation, Marina Bay Sands, Singapore, 29 May–3 June 2017.
23. Xiao, Z.; Wang, K.; Wan, Q.; Tan, X.; Xu, C.; Xia, F. A2S-Det: Efficiency anchor matching in aerial image oriented object detection. *Remote Sens.* **2021**, *13*, 73. [CrossRef]
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition, In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
25. Liu, S.; Qi, L.; Qin, H. Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018.
26. COCO Common Objects in Context. Available online: <https://cocodataset.org/> (accessed on 10 October 2022).
27. Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; Sun, J. CrowdHuman: A benchmark for detecting human in a crowd. *arXiv* **2018**, arXiv:1805.00123.
28. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2020**, *52*, 8574–8586. [CrossRef] [PubMed]
29. Jiang, S.; Jiang, C.; Jiang, W. Efficient structure from motion for large-scale UAV images: A review and a comparison of SfM tools. *ISPRS J. Photogramm. Remote. Sens.* **2020**, *167*, 230–251. [CrossRef]
30. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
31. Koletsis, E.; Cartwright, W.; Chrisman, N. Identifying approaches to usability evaluation. In Proceedings of the 2014 Geospatial Science Research Symposium, Melbourne, Australia, 2–3 December 2014.
32. Hornung, A.; Wurm, K.M.; Bennenwitz, M.; Stachniss, C.; Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **2013**, *34*, 189–206. [CrossRef]
33. Irmak, E.; Ertas, A.H. A review of robust image enhancement algorithms and their applications. In Proceedings of the 2016 IEEE Smart Energy Grid Engineering Conference, Oshawa, ON, Canada, 21–24 August 2016.
34. Romero-Ramirez, F.J.; Muñoz-Salinas, R.; Medina-Carnicer, R. Speeded up detection of squared fiducial markers. *Image Vis. Comput.* **2018**, *76*, 38–47. [CrossRef]
35. Zhang, Q.; Zhao, C.; Fan, F.; Zhang, Y. Taikobot: A full-size and free-flying humanoid robot for intravehicular astronaut assistance and spacecraft housekeeping. *Machines* **2022**, *10*, 933. [CrossRef]

## Article

# Small-Object Detection for UAV-Based Images Using a Distance Metric Method

Helu Zhou <sup>1,2,†</sup>, Aitong Ma <sup>1,†</sup>, Yifeng Niu <sup>1</sup> and Zhaowei Ma <sup>1,\*</sup>

<sup>1</sup> College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410078, China

<sup>2</sup> Aircraft Technology Branch of Hunan Aerospace Co., Ltd., Changsha 410200, China

\* Correspondence: mazhaowei1989@126.com

† These authors contributed equally to this work.

**Abstract:** Object detection is important in unmanned aerial vehicle (UAV) reconnaissance missions. However, since a UAV flies at a high altitude to gain a large reconnaissance view, the captured objects often have small pixel sizes and their categories have high uncertainty. Given the limited computing capability on UAVs, large detectors based on convolutional neural networks (CNNs) have difficulty obtaining real-time detection performance. To address these problems, we designed a small-object detector for UAV-based images in this paper. We modified the backbone of YOLOv4 according to the characteristics of small-object detection. We improved the performance of small-object positioning by modifying the positioning loss function. Using the distance metric method, the proposed detector can classify trained and untrained objects through object features. Furthermore, we designed two data augmentation strategies to enhance the diversity of the training set. We evaluated our method on a collected small-object dataset; the proposed method obtained 61.00%  $mAP_{50}$  on trained objects and 41.00%  $mAP_{50}$  on untrained objects with 77 frames per second (FPS). Flight experiments confirmed the utility of our approach on small UAVs, with satisfying detection performance and real-time inference speed.

**Keywords:** small-object detection; backbone design; object positioning; object classification; UAV flight experiment

**Citation:** Zhou, H.; Ma, A.; Niu, Y.; Ma, Z. Small-Object Detection for UAV-Based Images Using a Distance Metric Method. *Drones* **2022**, *6*, 308. <https://doi.org/10.3390/drones6100308>

Academic Editor: Diego González-Aguilera

Received: 11 September 2022

Accepted: 17 October 2022

Published: 20 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Nowadays, Unmanned aerial vehicles (UAVs) play an important role in civil and military fields, such as system mapping [1], low-attitude remote sensing [2], collaborative reconnaissance [3], and others. In many applications, reconnaissance tasks are mostly based on UAV airborne vision. In this case, the detection and recognition of ground targets is an important demand. However, when the UAV flies at high altitudes, the captured object occupies a relatively small pixel scale in UAV airborne images. It is a challenge to detect such small objects in complex large scenes. Additionally, due to the limited computing resources in UAVs, many large-scale detection models based on server and cloud computing are not suitable for online real-time detection of small unmanned aerial vehicles. In this case, achieving fast and accurate small-object detection using the onboard computer becomes challenging. This paper mainly focuses on the detection of small objects in UAV reconnaissance images.

Combining with the flight characteristics of small UAVs and the computing capability of onboard processors, this paper selects a neural-network-based model as the basic detection model. To the best of our knowledge, most of the current detection algorithms for UAVs use one-stage detectors [4]. One of the state-of-the-art detectors among the one-stage detectors is YOLOv4 [5]. The YOLOv4 object detector integrates various classic ideas [6–9] in the field of object detection and works at a faster speed and higher accuracy than other alternative detectors. We choose YOLOv4 as the benchmark detector. The YOLOv4 detector

was proposed and trained using a common dataset [10] covering various objects. However, the objects in the field of UAV reconnaissance that we are concerned with are limited in category, such as cars, aircraft, and ships. There are many instances of subdividing these limited categories of targets, but current object detection training sets rarely care about all types of objects. Therefore, objects that have not appeared in the training set are difficult to recognize in the UAV reconnaissance image during the inference stage, which is also a major challenge for UAV object detection. Generally, these objects are small in the vision of UAVs. When the flight altitude of the UAV is different, the image pixels of the same object are also different. Since YOLOv4 is a multi-scale detector, we improve YOLOv4 to make it more suitable for small-object detection in UAV reconnaissance images. Furthermore, the images of the same scene obtained by the UAV are different under different flight weather conditions.

To solve the above challenges, we proposed a small-object detection method that is applied to UAV reconnaissance. Our contributions are described as follows:

1. We propose two data augmentation methods to improve the generalization of the algorithm on the scene;
2. We design a backbone network that is suitable for small-object detection and modify the positioning loss function of the one-stage detector to improve detection accuracy;
3. We design a metric-based object classification method to classify objects into subclasses and detect objects that do not appear in the training phase, in other words, untrained objects.

The remainder of this manuscript is structured as follows. Section 2 introduces some related works of object detection algorithms. Section 3 formulates the detector structure for UAV untrained small-object detection and introduces the improved algorithm. Experimental results are presented in Section 4 to validate the effectiveness of our proposed method. Section 5 concludes this paper and envisages some future work.

## 2. Related Works

### 2.1. Small-Object Detection Algorithm

Most of the state-of-the-art detectors are based on deep-learning methods [11]. These methods mainly include two-stage detectors, one-stage detectors, and anchor-free detectors. Two-stage detectors first extract possible object locations and then perform classification and relocation. The classic two-stage detectors include spatial pyramid pooling networks (SPPNet) [9], faster region-CNN (RCNN) [12], etc. The one-stage detectors perform classification and positioning at the same time. Some effective one-stage detectors mainly include the single shot multi-box detector (SSD) [13], You Only Look Once (YOLO) series [5,8,14,15], etc. Anchor-free detectors include CenterNet [16], ExtremeNet [17], etc. These methods do not rely on predefined anchors to detect objects. In addition, some scholars have introduced transformers into the object detection field, such as detection with transformers (DETR) [18] and vision transformer faster RCNN (ViT FRCNN) [19], which have also achieved good results. However, the detection objects of the general object detectors are multi-scale. They are not designed for small-object detection specifically.

Small-object detection algorithms can be mainly divided into two kinds. One is to improve the detection performance of small objects with multiple scales in a video or image sequence. The other is to improve the detection performance of small objects with only a scale in an image. The improved detection methods of small objects with multiple scales mainly include feature pyramids, data augmentation, and changing training strategies. In 2017, Lin et al. proposed feature pyramid networks (FPN) [20], which improves the detection performance effect of small objects by fusing high-level and low-level features to generate multi-scale feature layers. In 2019, M. Kisantal et al. proposed two data augmentation methods [21] for small objects to increase the frequency of small objects in training images. B. Singh et al. designed scale normalization for image pyramids (SNIP) [22]. SNIP selectively backpropagates the gradients of objects of different sizes, and trains and tests images of different resolutions, respectively. The research object of

these methods is multi-scale objects, which cannot make full use of the characteristics of small objects. The approaches that only detect small objects with a scale are mainly of three kinds: designing networks, using context information, and generating super-resolution images. L. Sommer et al. proposed a very shallow network for detecting objects in aerial images [23]. Small-object detection based on network design is few and immature. J. Li et al. proposed a new perceptual GAN network [24] to improve the resolution of small objects to improve the detection performance. In this paper, we focus on algorithms that are suitable for small-object detection in UAV reconnaissance images.

## 2.2. Object Detection Algorithm for UAV

The object detection algorithms used in UAVs are mainly designed based on the requirements of the task scenarios. M. Liu et al. proposed an improved detection algorithm based on YOLOv3 [8]. The algorithm first optimizes the Res-Block and then improves the darknet structure by increasing the convolution operation. Y. Li et al. proposed a multi-block SSD (MBSSD) mechanism [25] for railway scene monitored by UAVs. MBSSD uses transfer learning to solve the problem of insufficient training samples and improve accuracy. Y. Liu et al. proposed multi-branch parallel feature pyramid networks (MPFPN) [26] and used a supervised spatial attention module (SSAM) to focus the model on object information. MPFPN conducted experiments on a UAV public dataset named VisDrone-DET [27] to prove its effectiveness. These algorithms are combined with UAV application scenarios, but are all based on classic methods. They cannot work well when inferencing against untrained small objects.

## 3. Proposed Method

In this paper, we focus on small objects in images from the aerial view of UAVs. The proportion of object pixels in the image is less than 0.1% and the objects to be detected include objects that have not appeared in the training set. Current deep-learning-based object detection algorithms depend on a large amount of data. Therefore, we design an approach to expand the dataset in the proposed detection framework. In addition to the classic methods such as rotation, cropping, and color conversion, we propose two data augmentation methods—background replacement and noise adding—to improve the generalization of the detection algorithm. Background replacement gives the training images more negative samples to make the training set have a richer background. Noise adding can prevent the training model from overfitting the object.

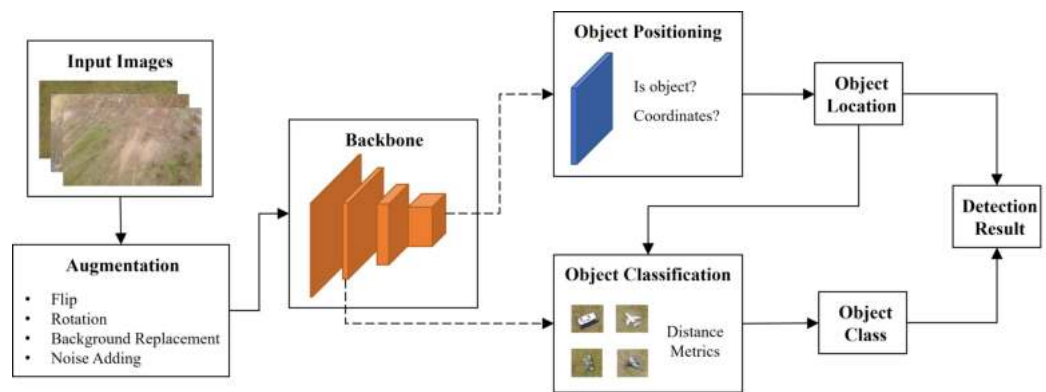
After preprocessing the training images, the image features need to be obtained through the backbone network. We choose YOLOv4 [5], which has a good trade-off between speed and accuracy, as the benchmark algorithm for research. Common backbones in object detection algorithms are used to detect multi-scale objects. The receptive field of the backbone module is extremely large. In YOLOv4, the input of the backbone called CSPDarknet53 [7] is  $725 \times 725$ . However, the number of pixels of the small object in the image generally does not exceed  $32 \times 32$ . For small-object detection, the backbone does not need such a large receptive field. Therefore, the common backbone needs to be modified to apply to small object detection.

The YOLO series algorithms have three outputs, whether it is an object, which category it belongs to, and the bounding box coordinates of the object. These outputs are calculated by the same feature map. However, the convergence direction of the object positioning and object classification is not consistent. For example, the same object may have different coordinate positions, but the same coordinate position may have different objects. From this perspective, object positioning and object classification cannot use the same features. One of the research ideas of the proposed algorithm is to use different methods to deal with object positioning and object classification separately. This can avoid influence between positioning and classification.

As the input and output sizes of convolution neural networks (CNNs) are determined, the YOLO series algorithms can only detect a fixed number of objects. In order to recognize

untrained categories of objects, we extract the object features from feature maps and design a metric-based method to classify objects. When the algorithm is trained, the classification loss and the positioning loss are backpropagated at the same time.

The overall structure of the proposed detector is shown in Figure 1. The data augmentation module uses methods such as background replacement and noise adding to change the training data to obtain better training performance. The backbone network extracts the multi-layer features in the image for object positioning and object classification. The object positioning module uses high-level features to obtain the center point coordinates, width and height of the object. The object classification module uses the positioning results to extract object features and then judges the category of the object by distance measurement. Finally, the detection result are obtained by combining the positioning and classification calculation.



**Figure 1.** Structure of the proposed small-object detection method, including data augmentation, backbone network and object positioning and object classification modules.

### 3.1. Image Augmentation

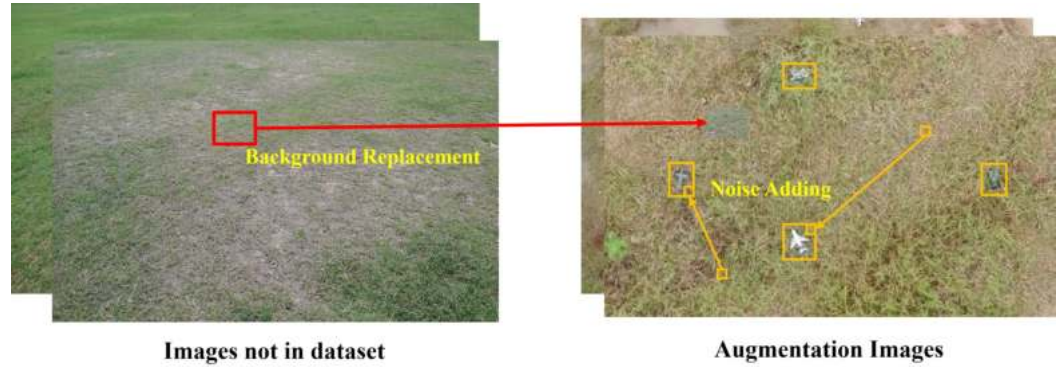
We analyze two main reasons for insufficient datasets. One is that the background in the image is relatively monotonous and the other is that the object state is relatively monotonous. Aimed at these two reasons, we propose two methods to increase the image number of the database, as shown in Figure 2.

The purpose of background replacement is to reduce the impact of background singularity in UAV-based images. We randomly crop some areas in images that are not in the training set to cover areas in training images that do not contain the object. This can increase the diversity of negative samples, making the model eliminate the interference of irrelevant factors.

The output result of the object detection is a rectangular box surrounding the object. However, the object is generally not a standard rectangle, which means that the detected rectangular box will contain some information that does not belong to the object. If the object location does not change much, it is very likely to overfit the background information near the object, which is not conducive to the generalization of the detector. Generally speaking, invalid information will appear at the edge of the rectangular box. Therefore, we design a noise-adding augmentation strategy. We randomly select pixels in the image to cover the pixels near the edge of the rectangular box containing the object. Since we cannot accurately determine whether a pixel belongs to the object or background, we fill the pixels along the bounding box and the pixel block used as noise contains no more than 10 pixels, considering that the object is in the center of the detection box. The pixels near the object are changed by randomly adding noise to improve the generalization of the detector. The formula of background replacement is expressed as follows:

$$\begin{aligned}\tilde{x} &= M \odot x_A + (1 - M) \odot z_B \\ \tilde{y} &= y_A\end{aligned}\quad (1)$$

where  $M \in \{0,1\}^{W \times H}$  represents the part of the image that needs to be filled and  $\odot$  means pixel by pixel multiplication.  $x_A$  and  $x_B$  denote two samples in the training set.  $y_A$  and  $y_B$  represent the label corresponding to the training samples, and  $z_B$  is an image in the background image set.



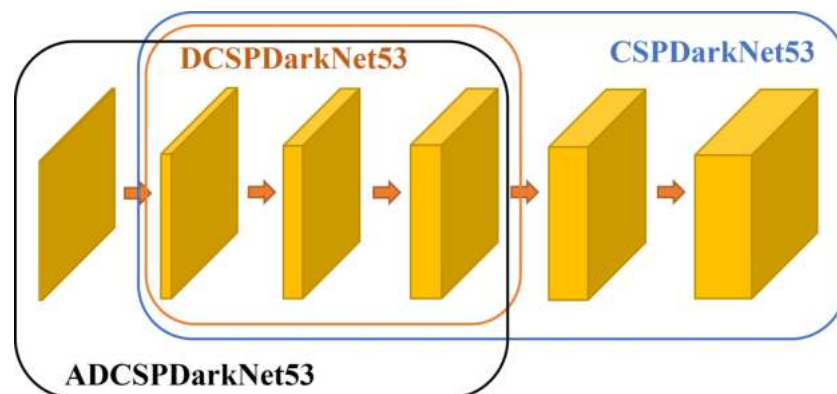
**Figure 2.** Two data augmentation methods. The red line represents the background replacement and the yellow lines represent noise adding.

### 3.2. Backbone Design

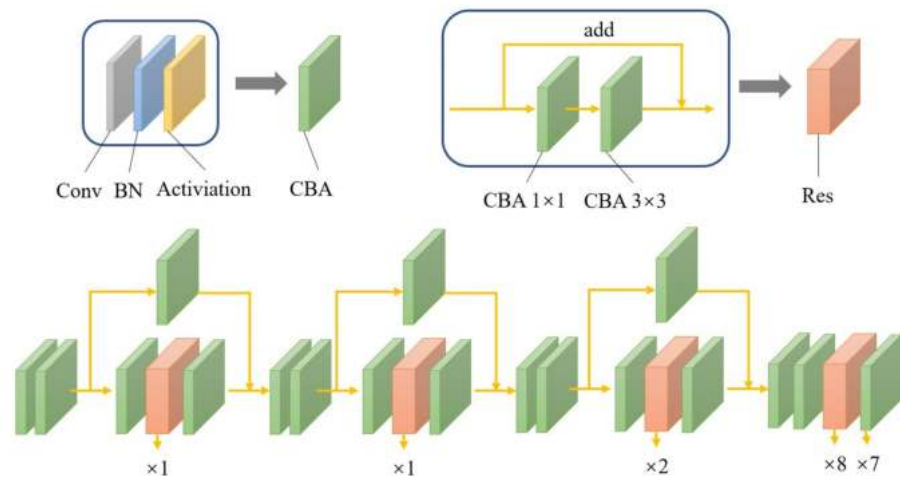
YOLOv4 proves that CSPDarknet53 is the relatively optimal model. We modify CSPDarknet53 to make it suitable for small-object detection. The comparison between the modified backbone in this paper and the original backbone is shown in the Figure 3.

Compared with the original network, the modified network reduces the receptive field and improves the input image resolution without increasing the computational complexity. Since the research object focuses on small-scale objects, there is no need to consider large-scale objects. The modified backbone deletes the network layers used to predict large-scale objects, which reduces the network depth by half. We call it DCSPDarknet53.

In order to reduce the computational complexity of deep learning, the resolution of the input image is usually downsampled. However, low image resolution will make it difficult to correctly classify and locate small objects. Therefore, a convolutional layer is added in the front of the network to calculate higher-resolution images. We call it as ADCSPDarknet53. At the cost of a small amount of calculation speed, the detection accuracy is improved on a large scale. The specific network structure of our proposed backbone network ADCSPDarknet53 is shown in Figure 4.



**Figure 3.** Comparison of backbones. CSPDarknet53 is the backbone of YOLOv4. DCSPDarknet53 is a backbone for small objects. ADCSPDarknet53 is a backbone that increases the downsampling network layer.



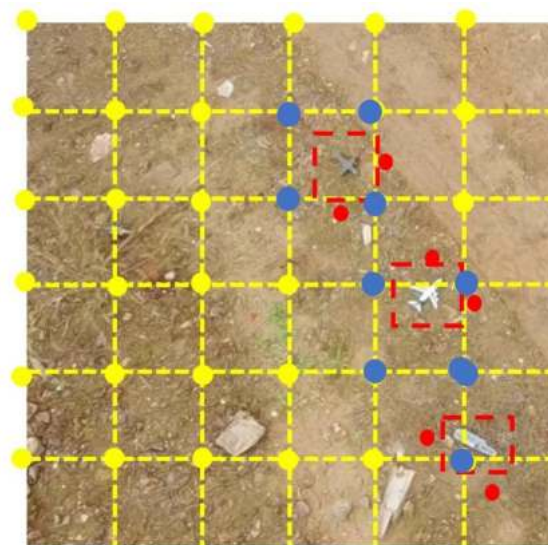
**Figure 4.** The structure of backbone network ADCSPDarknet53. Conv means convolution layer. BN denotes batch normalization. Activation layer uses ReLU function.

### 3.3. Object Positioning

The object positioning algorithm is improved based on the YOLO series. YOLOv5 [28] uses a positive sample expansion method. In addition to the original positive sample, two anchor points close to the object center are also selected as positive samples. The calculation formula is expressed as follow:

$$P = \left\{ \begin{array}{l} p, \\ \text{if } (p \cdot x - \lfloor p \cdot x \rfloor \leq 0.5) : p + (-1, 0) \\ \text{if } (p \cdot x - \lfloor p \cdot x \rfloor > 0.5) : p + (1, 0) \\ \text{if } (p \cdot y - \lfloor p \cdot y \rfloor \leq 0.5) : p + (0, -1) \\ \text{if } (p \cdot y - \lfloor p \cdot y \rfloor > 0.5) : p + (0, 1) \end{array} \right\} \quad (2)$$

where  $P$  represents the expanded positive sample coordinate set and  $p$  means the original positive sample coordinate. For example, in Figure 5, the gray plane is predicted by the grid where the gray plane's center point is located. After the expansion, the gray plane is also predicted by the grid where the red dots are located.



**Figure 5.** Selection of positive sample. The yellow dots are anchor points. The red dots are positive samples expanded by YOLOv5. The blue dots are positive samples proposed in this article.

The distance between each anchor point is not far in the last feature layer, which means that the object may contain multiple anchor points. It is not appropriate to select the closest anchor point as a positive sample and define other anchor points as negative samples. Therefore, we revise the selection method of positive samples. We calculate the four anchor points around the object center point as positive samples, as shown by the blue dots in Figure 5.

The YOLO series algorithms define the probability that the anchor contains the object as 1. Since each anchor point has a different distance from the object, the strategy of defining all as 1 cannot reflect the difference between different anchor points. Therefore, we use the Euclidean distance between the anchor point and the object center point as the metric for the probability that the anchor contains the object. As the positive sample must contain the object, the probability of containing objects of the positive sample anchor point cannot be 0. According to the design principle, the function of calculating the probability is shown as follows:

$$p_{obj} = 1 - \left( (x_a - x_t)^2 + (y_a - y_t)^2 \right) / 4 \quad (3)$$

### 3.4. Untrained Sub-Class Object Detection

Generally speaking, top-level features are beneficial to object positioning and bottom-level features are beneficial to object classification. In order to separate the process of object localization and object classification to reduce the distractions between these two and make better use of the features extracted from the backbone network, we select features from the middle layers of the backbone for object classification. Through object positioning, we can obtain the coordinates of the object. Using these coordinates, the feature vector of the object can be extracted from feature maps and then can be used to classify the object.

In UAV airborne images, it is common to think of objects in terms of large classes, such as aircraft, cars, buildings, pedestrians, etc. However, specific objects such as black cars and white cars are difficult to determine. To address this sub-class classification problem, we divide the object classification process into the rough classification process and the fine classification process. Rough classification mainly distinguishes objects with large differences in appearance, such as aircraft and cars. Fine classification mainly distinguishes objects that have similar appearance characteristics, but belong to different classes, such as black cars, white cars, etc.

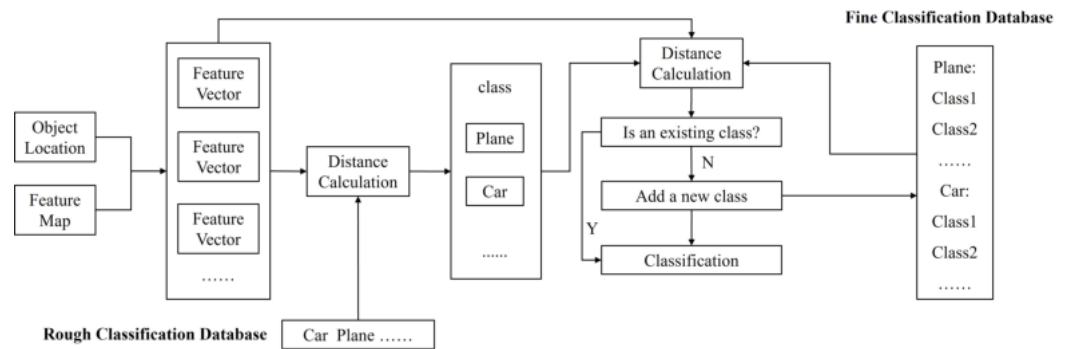
In this paper, a measurement method based on Euclidean distance is used to classify the object. The advantage is that it does not need to fix the object class. By using this metric learning, the algorithm can identify the potential objects in the scene that do not appear in training process, in other words, untrained object. The training goal of object classification is to make the object features of the same class as close as possible and to make the object features of different classes as far as possible. After extracting the object features, three objects are randomly selected from all objects, in which two classes are the same. We use the triple loss [29] as the loss function of object classification. The loss function is defined as:

$$\text{loss}_{cls} = \max(d(a_1, a_2) - d(a_1, b) + \text{thre}, 0) \quad (4)$$

where  $a_1$  and  $a_2$  represent objects that belong to the same class.  $b$  means the object that is different from  $a_1$  and  $a_2$  and  $\text{thre}$  is the expected distance between objects of different classes. The rough classification calculates the loss value between the object classes, while the fine classification calculates the loss value between the object sub-classes. The  $\text{thre}$  of the fine classification process is lower than the  $\text{thre}$  of the rough classification.

In the testing process, we input the labeled images with all objects into the trained model to obtain image features and then extract the feature vector of the object according to the object position to construct a classification database. The classification database is used to classify the object in the test image. The flowchart of object classification is shown in Figure 6. First, the rough classification database is used to determine the object class and then the fine classification database is used to determine the object sub-class.

The principle of object classification is to classify the object into the class closest to the object. If the distance between the object and any category in the database is greater than the threshold, the object is considered to belong to an unknown class that has not appeared in the database. If the distance between the object and the class closest to it is less than the threshold, the object is considered to belong to that class.



**Figure 6.** The flowchart of object classification for fine and rough classification.

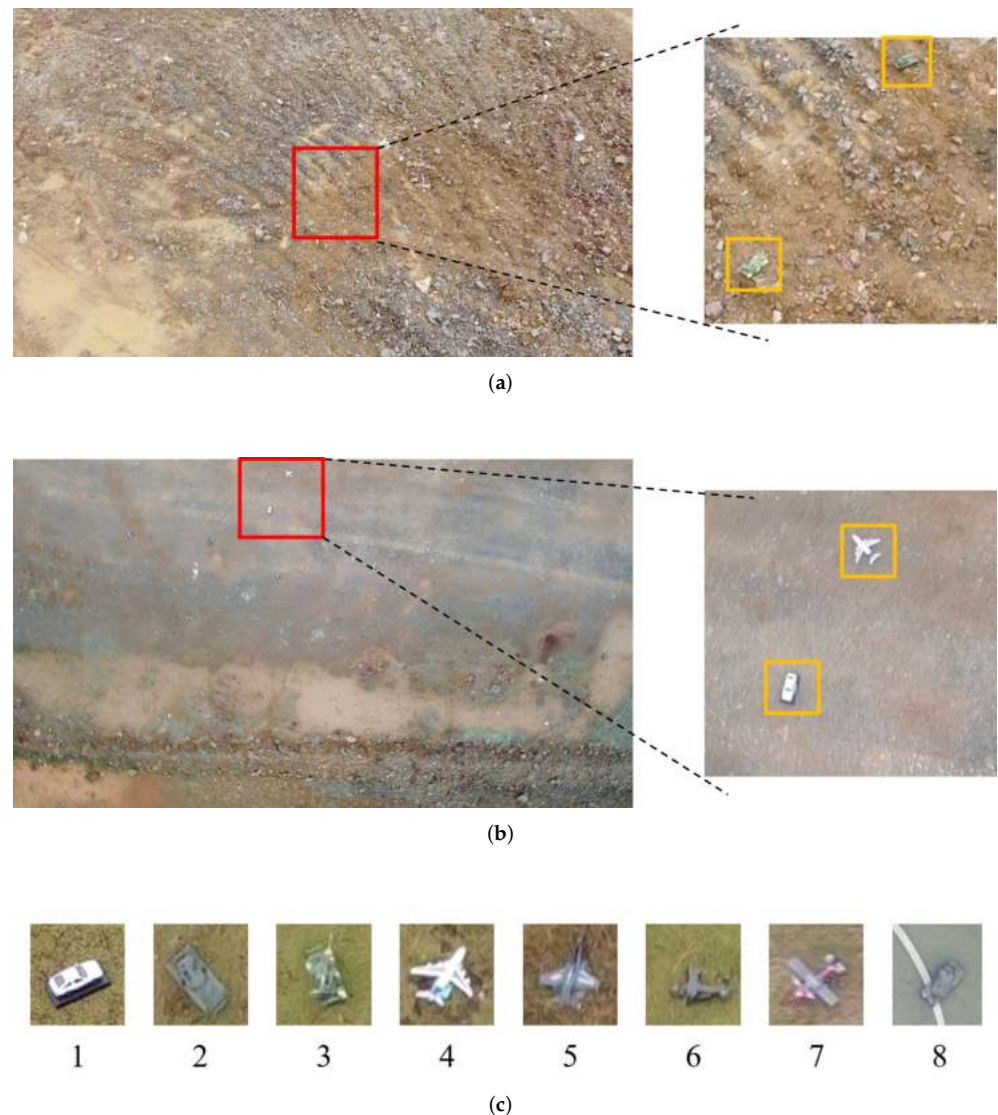
In summary, we designed two data augmentations—background replacement and noise adding—to increase the background diversity of the dataset. Based on the information flow of small objects through convolution layers, we modified the detector backbone CSPDarkNet53 to ADCSPDarknet53 to obtain a larger feature map for small-object detection as well as to reduce the computation cost. For object positioning, we selected the four anchor points around the object center point as positive samples and modified the function for calculating the objectness probability, which can increase the positioning accuracy of small objects. For object classification, we combined information from shallow feature maps and positioning results to perform rough and fine classification processes to obtain more accurate classification results and identity untrained sub-class small objects.

#### 4. Experiments

To evaluate the small-object detection and classification algorithm proposed in this paper, we first constructed a dataset consists of small objects. Then, we performed experiments on trained and untrained small objects to compare localization and classification performance. Finally, we conducted flight experiments to test the detection performance and real-time inference of the proposed algorithm on small UAVs.

##### 4.1. Dataset of Small Objects and Evaluation Metrics

We choose to detect small objects in the visual field of UAVs to evaluate our algorithm. In order to obtain as much target data as possible, we built a scaled-down experimental environment to collect our dataset. The UAV we used to collect the dataset was a DJI Mini2. To obtain various data, we used some small target models as objects to be detected. The target models were between 15 cm–25 cm in length and 5 cm–20 cm in width. When taking the image data, the flight altitude of the UAV was controlled between 8–10 m to simulate the dataset captured at high altitudes. As the resolution of captured images is  $1920 \times 1080$  pixels, the pixel ratio of the object to be detected in the image is less than 0.1% as shown in Figure 7a. Eight types of objects were selected to construct the dataset. There are two object classes in the dataset, car and plane. Each class has four sub-classes, which are listed in Figure 7b.



**Figure 7.** (a,b) are example images in the collected small-object dataset. The images on the left show the pixel proportion of the object to be detected in the whole image, and the patches on the right are a zoomed-in version of the red box on the left images. (c) shows eight objects in the dataset. 1, 2 and 3 are three trained cars belonging to different sub-classes. 4, 5 and 6 are three trained planes belonging to different sub-classes. 7 is an untrained plane and 8 is an untrained car. 7 and 8 are objects that did not appear in the training set and validation set.

The collected dataset are split into training set, validation set and testing set, with 977 images, 100 images and 195 images, respectively. For the object-detection task, the label file contains four object positioning coordinates and an object class label. For the sub-class object classification task, the label file contains two object class labels, in which the first label denotes the object class and the second one is the sub-class. In order to evaluate the detection performance of the proposed algorithm on untrained small objects, the designs of the testing set are slightly different from the training set and validation set. The training set and the validation set contain six types of objects, including three types of cars and three types of planes. In addition to these six types of objects, one type of car and one type of plane are added to the testing set.

For evaluation, we use the general indicators in the field of object detection [30] to evaluate the performance of the proposed algorithm, including mean average precision (mAP), mean average precision at  $IoU = 0.50$  ( $mAP_{50}$ ) and frames per second (FPS). Intersection over union (IoU) evaluates the overlap between the ground truth bounding

boxes and the predicted bounding box. Based on IoU and the predicted object category confidence scores, mAP is applied to measure the detection performance with classification and localization.  $mAP_{50}$  is calculated with  $IoU \geq 0.50$ . These two metrics are performed under the COCO criteria [10]. FPS is used to evaluate the running speed of the algorithm on certain computation platforms. The FPS calculation method is the same as in YOLOv4.

#### 4.2. Implementation Details

We chose YOLOv4 as our baseline model since the proposed network is improved based on YOLOv4. For comparison of object detection, we implemented several object-detection models on the collected dataset, including Faster RCNN (with VGG16 [31] as backbone), SSD [13], FCOS [31], PPYOLO [32], PPYOLOv2 [33], PPYOLOE [34] and PicoDet [35]. Among them, Faster RCNN is a two-stage object detector and the rest are one-stage detectors. FCOS, PPYOLOE and PicoDet are anchor-free detectors, and PicoDet is designed for mobile devices. All the detectors have the same size of input image ( $608 \times 608$ ) and were trained from the pretrained model on the COCO dataset [10] to obtain faster and better convergence. The training epochs were set to 300 with initial learning rate 0.001 and Adam optimizer. Then, learning rate decayed by 0.1 times at the 150th epoch and 250th epoch. The batch size was set to be 16. For the special sub-class classification part of our method, the *thre* of the rough classification was 10 and the *thre* of the fine classification was 2. All the training and testing experiments were conducted on one NVIDIA Quadro GV100 GPU.

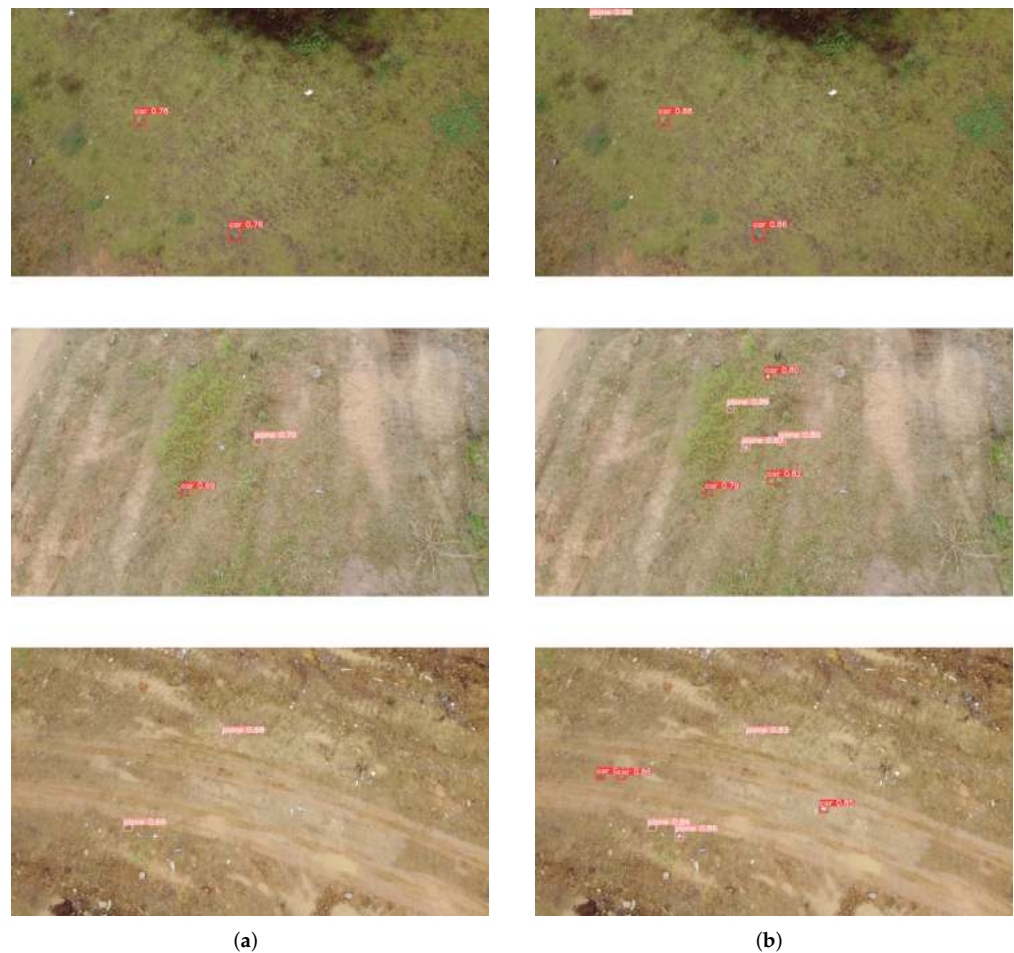
#### 4.3. Experiment Results

##### 4.3.1. Small Object Detection

We compare the detection performance on small objects with several existing object detectors. The results are listed in Table 1, from which we can see that our proposed method gives the highest average precision (33.80%) compared to the others, as well as running at the highest speed (77 FPS). It also achieves the third highest  $mAP_{50}$ , with 61.00% among ten models. These improvements can be attributed to the following aspects: (1) the modified backbone network focuses more on the small objects and discards the deep layers, which have little effect on detecting small objects. Meanwhile, the interference of extra parameters on network learning is reduced. (2) Metric-based learning improves the ability of the network to classify objects. (3) The proposed object positioning method increase the number of positive samples and thus improves small-object localization abilities. (4) The modified backbone network reduces computation significantly, which makes the network run at a faster speed and achieves the performance of real-time detection. Figure 8 shows some detection results of the collected dataset using YOLOv4 and our algorithm. Our algorithm has stronger ability to detect small objects.

**Table 1.** Experiment results of small object detection.

Method	FPS	$mAP_{50}$	mAP
PPYOLOE_s	56	33.40%	12.70%
PPYOLOE_l	34	52.50%	22.40%
PicoDet_s	68	21.70%	7.50%
FasterRCNN	18	29.40%	10.80%
SSD	51	34.80%	10.80%
PPYOLO	54	63.70%	27.50%
PPYOLOv2	36	62.00%	25.60%
FCOS	26	21.20%	8.80%
YOLOv4	71	39.60%	21.70%
Ours(ADCSPDarknet53)	<b>77</b>	61.00%	<b>33.80%</b>



**Figure 8.** Small-object detection experiment with other algorithms. (a) YOLOv4 algorithm detection results; (b) our algorithm detection results.

#### 4.3.2. Untrained Sub-Class Object Classification

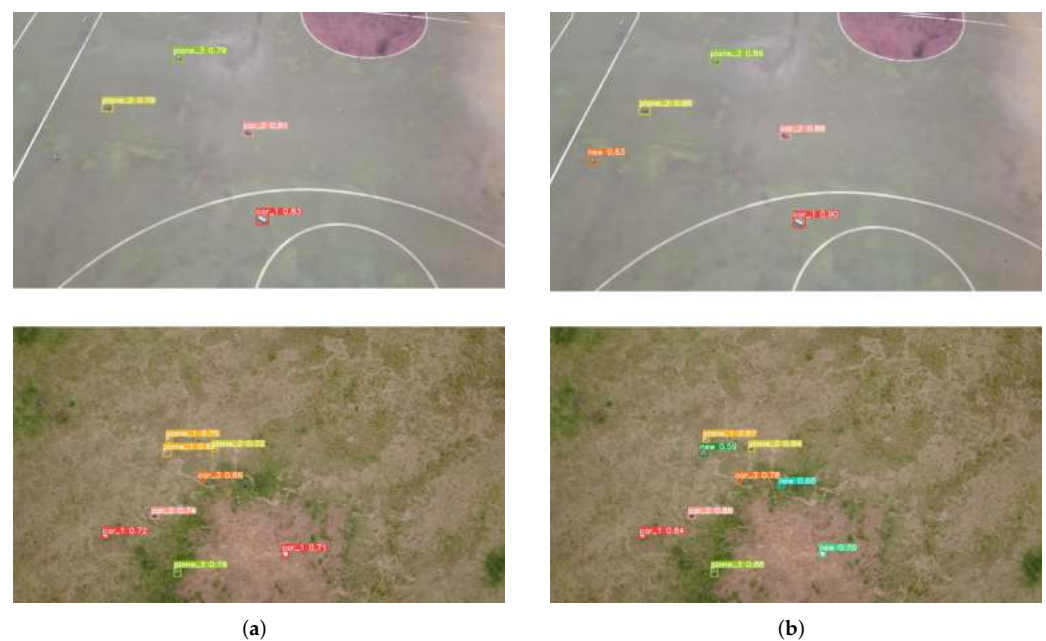
In our proposed method, object classification includes the process of rough object classification and fine object classification. The designed classification method has two advantages. One is that it can classify untrained objects, and the other is that it avoids the mutual influence of classification and positioning. To illustrate both points, we conduct multiple comparative experiments. The experimental results of object classification are shown in Table 2. Experiments 1, 2 and 3 represent detection results of YOLOv4 under different conditions. Experiments 4 and 5 are the detection results of the same rough classification model under different test categories. Experiments 6, 7 and 8 are the detection results of the same fine classification model under different test categories.

Comparing Experiments 1 and 3, the accuracy of object detection decreases from 88.30% to 65.60% when objects are classified during training, which shows the interference between object classification and localization. Through Experiments 3 and 4, it can be found that metric-based learning is beneficial to improve the result of object detection, as the  $mAP_{50}$  increases from 65.60% to 88.30%. It can be demonstrated by Experiments 1, 3 and 6 that the detection performance of the object is worse for more categories. In Experiment 8, we can find that the untrained car and the untrained plane can be detected with 49.4%  $mAP_{50}$  and 34.1%  $mAP_{50}$ , respectively. Although the metric-based method is not very accurate in detecting untrained objects, it can still locate untrained objects and distinguish them from trained objects.

**Table 2.** Experiment results of untrained object classification.

Experiment	Model	Number of Training Classes	Number of Testing Classes	$mAP_{50}$ (Trained)	$mAP_{50}$ (Untrained)
1	YOLOv4	1	1	88.30%	-
2	YOLOv4	2	2	51.10%	-
3	YOLOv4	2	1	65.60%	-
4	rough classification	2	1	83.80%	-
5	rough classification	2	2	64.70%	-
6	fine classification	9	1	78.30%	-
7	fine classification	9	2	57.70%	-
					aircraft: 34.10%
8	fine classification	9	9	41.900%	car: 49.40%

Figure 9 shows the visualization results of untrained object classification using YOLOv4 and our algorithm. For the untrained sub-class objects, YOLOv4 will give incorrect classification results, but the proposed algorithm will add these untrained objects to new sub-classes using metric-based learning.



**Figure 9.** Untrained object classification experiment with other algorithms. (a) YOLOv4 algorithm detection results; (b) our algorithm detection results. Different colors of bounding boxes mean different sub-classes, and the recognized untrained sub-class objects are labeled with ‘new’.

#### 4.3.3. Ablation Study

To analyze the effectiveness of our proposed method, we conducted an ablation study on data augmentation, backbone design and object positioning.

**Data Augmentation.** Based on YOLOv4, we analyze the results of two data augmentation methods. It can be seen from Table 3 that the two types of data augmentation can improve detection performance. The results show that replacing part of the image background can lead the network to learn more combinations of patterns and effectively increase the diversity of the dataset. Adding noise around the object reduces the overfitting to special backgrounds. However, the effect of running the two methods at the same time

is not as good as the effect of running the two methods separately. This is mainly because the data augmentation method we propose introduces significant noise while increasing the diversity of the dataset. Applying both methods at the same time may cause too much noise and cannot obtain better performance.

**Table 3.** Experiment results of data augmentation.

Image Input Size	Replace Background	Add Noise	$mAP_{50}$	mAP
$608 \times 608$	N	N	36.50%	19.10%
$608 \times 608$	Y	N	44.60%	24.90%
$608 \times 608$	N	Y	41.70%	20.70%
$608 \times 608$	Y	Y	39.60%	22.40%

**Backbone Design.** The backbone design consists of two steps. First, the DCSPDarknet53 deletes the deep network layers used to detect large-scale objects in the original detection backbone CSPDarknet53 to reduce the influence of the deep network on the detection of small objects. Then, ADCSPDarknet53 adds a network layer for downsampling at the front end of the network, so as to obtain better detection results while increasing the computational complexity as little as possible. The experiment results are shown in Table 4. Compared to the CSPDarknet53-based detector, there is little increase in  $mAP_{50}$  and mAP of the DCSPDarknet53-based detector in small-object detection, but the calculation speed is more than double, which proves that small-object detection can use a high-resolution network with fewer layers. As for the ADCSPDarknet53-based detector, the  $mAP_{50}$  is increased by 18% and the mAP is increased by 11% compared to the original detector. Although the FPS drops, it still runs faster than the CSPDarknet53-based detector. In the actual scene, the image size and backbone can be adjusted as needed.

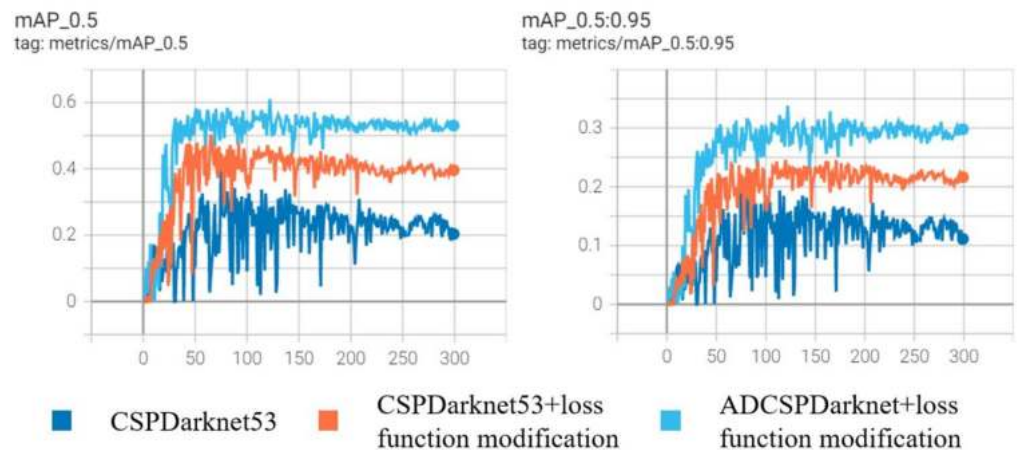
**Table 4.** Experiment results of backbone design.

Backbone	Image Input Size	FPS	$mAP_{50}$	mAP
CSPDarknet53	$608 \times 608$	71	39.60%	21.70%
DCSPDarknet53	$608 \times 608$	166	42.90%	23.90%
ADCSPDarknet53	$1216 \times 1216$	77	57.60%	32.70%

**Object Positioning.** We modified the loss function of object positioning; Table 5 shows the detection evaluation of detectors with and without loss function modification. After modifying the loss function of object positioning, the  $mAP_{50}$  of the CSPDarknet53-based detector is increased by 10.7% and the mAP is increased by 2.9%. For the ADCSPDarknet53-based detector, the  $mAP_{50}$  increased by 3.4% and the mAP increased by 1.1%. This proves the effectiveness of the modified loss function, which can select positive samples that can represent the ground truth more accurately from many candidate samples. The training process of object positioning is shown in Figure 10. With the modified object positioning loss function, the training process is more stable.

**Table 5.** Experiment results of object positioning.

Backbone	Loss Function Modification	$mAP_{50}$	mAP
CSPDarknet53	N	39.60%	21.70%
CSPDarknet53	Y	50.30%	24.60%
ADCSPDarknet53	N	57.60%	32.70%
ADCSPDarknet53	Y	61.00%	33.80%



**Figure 10.** Training process of object positioning.

#### 4.4. Flight Experiments

In order to verify the effectiveness of the proposed algorithm in the actual UAV application scenario, we built a UAV flight experiment system and deployed the proposed algorithm on a small drone.

##### 4.4.1. Experiment Settings

The drone we used as the experiment platform was a Matrice M210v2 drone manufactured by DJI Innovations. Its overall dimensions are  $883 \times 886 \times 398$  mm with a maximum takeoff weight 6.14 kg. The onboard optical camera was a DJI innovation company Chanshi X5s camera equipped with a DJI MFT 15 mm/1.7 ASPH lens. It was fixed to the drone body through a 3-DOF gimbal and its posture can be controlled with a remote controller. The resolution of the images taken by the camera was set to be  $1920 \times 1080$  pixels. To implement our proposed algorithm on the drone, we deployed a Nvidia Jetson Xavier NX processor on the drone for real-time processing. In order to ensure the safe outdoors flight of the drone, we also set the real-time kinematic (RTK) global navigation satellite system on the drone for the positioning. The flight experiment system is shown in Figure 11.

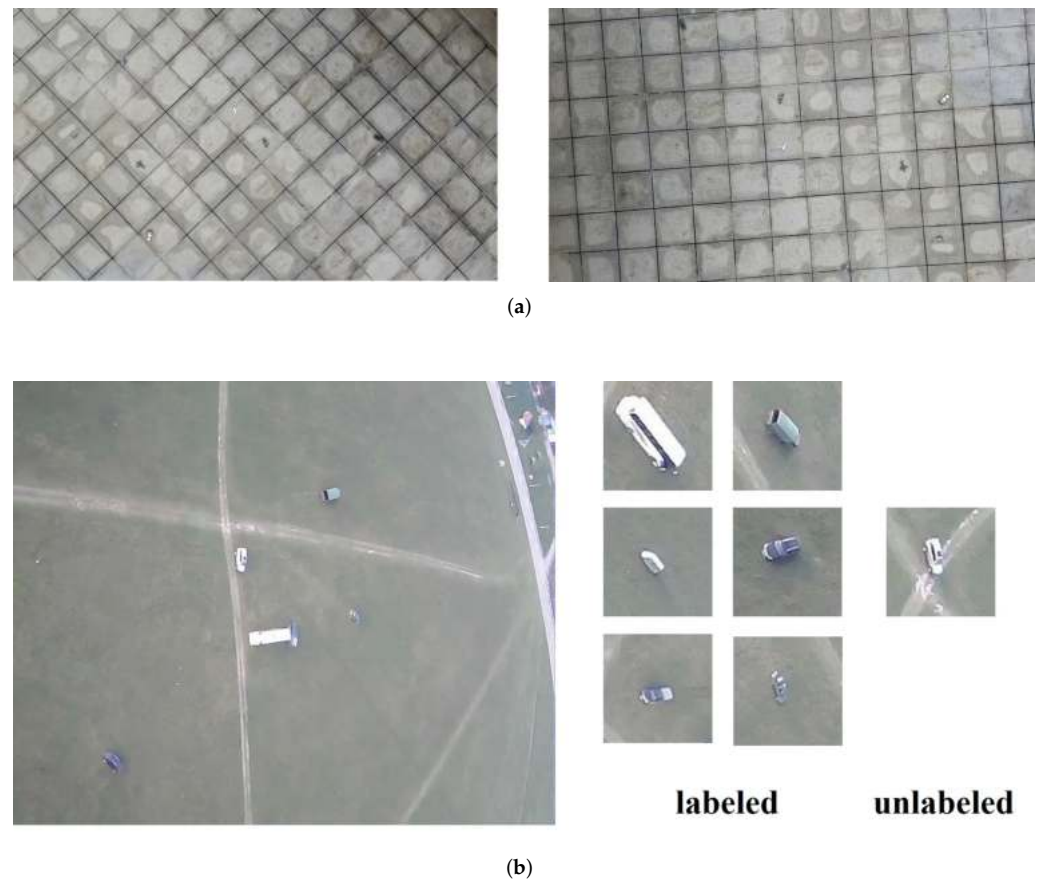


**Figure 11.** Hardware system for flight experiments. (a) DJI Matrice M210v2 drone; (b) DJI camera; (c) Nvidia Jetson Xavier NX.

The proposed algorithm is implemented by PyTorch 1.10 on Nvidia Jetson Xavier NX's GPU with a computational capacity of 7.2. All programs run on Robot Operating System (ROS) systems. While the drone is flying, we use the Rosbag tool to record the on-board processing data, such as the real-time detection image results. Once the drone is back on the ground, we can use the Rosbag's playback function to check how well the algorithm works.

We set up two flight scenarios to validate our algorithm with trained and untrained objects. In one detection scenario, the objects to be detected were the small models used for creating the above dataset, but with new backgrounds. In this case, the flight altitude of the drone was set to 10 m to stay consistent with the dataset. The purpose of this detection

scenario is to verify the generalization performance of the learned model in practical application scenarios. In another detection scenario, the model detected real vehicles with flight altitude of 95 m. We used seven different types of vehicles to test the classification and localization ability of the model for object classes with high inter-class similarity. In this case, we collected new data but only six types were labeled and appear in the training set. Then the model is retrained and tested for detection performance and speed. The two scenarios settings are shown in Figure 12.

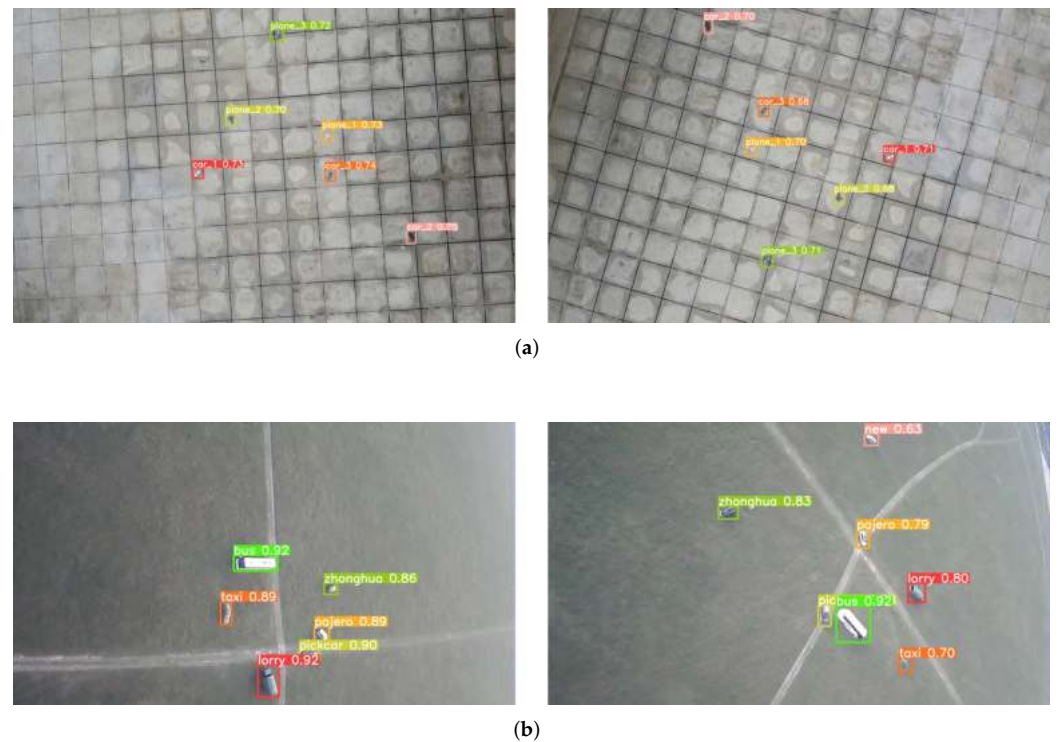


**Figure 12.** (a) The first detection scenario is set to have the same objects as the collected dataset, but with different background. (b) The second detection scenario uses real vehicles as objects. The figure on the left shows part of the drone’s field of view, and the right images show different types of vehicles, with six labeled types and one unlabeled.

#### 4.4.2. Results

Some qualitative detection results are shown in Figure 13. In Figure 13a, our proposed algorithm can detect small objects in the visual field without being influenced by the changing background. This is because the data enhancement method we used effectively prevents the model from overfitting to the background during the training process. In Figure 13b, the learned objects are detected by the proposed detector. In addition, the potential target (the unlabeled one) is also identified by the network and classified into a new class according to metric learning.

It is worth noting that small-object detection during flight faces additional challenges, such as camera vibration and motion caused by flight. Camera motion in the imaging process leads to the blurring of objects and damages the features. In this case, our algorithm can still detect small objects in the airborne visual field accurately. Our proposed algorithm not only extracts the features of small objects with CNNs, but also distinguishes the inter-class and intra-class differences of objects by measuring the distance metric. This more powerful feature extraction method helps reduce the effect of motion blur.



**Figure 13.** (a) Detection results on small model objects with different backgrounds. (b) Detection results on seven types of vehicles.

We also checked the real-time performance of our proposed method. We computed the runtime of the algorithm on the edge GPU (Nvidia Xavier NX) using different input image resolutions, including the pre-processing phase, model inference phase, and post-processing phase. The results are listed in Table 6. For images with input sizes of  $416 \times 416$ ,  $608 \times 608$  and  $640 \times 640$ , our algorithm reaches an average speed of 22 FPS. The small runtime difference mainly comes from the pre-processing phase and post-processing phase, since these two phases are run on the CPU. The parallel computing capability of the GPU makes the inference time of the model almost the same. However, for images with input size of  $1216 \times 1216$ , it takes more than twice as long to process a single frame. As there are four times as many pixels in the image, more time is needed to perform normalization for each pixel and permute the image channels in the pre-processing stage. For model inference, the larger input image size makes the feature map in the network larger, with more activation needs to compute, which accounts for the increase of time usage [36]. During the post-processing phase, more candidate detection boxes need to be computationally suppressed.

In our flight test, we used an image input size of  $608 \times 608$ . Without any acceleration library, our algorithm can achieve real-time performance, which is sufficient for reconnaissance missions on UAVs. Some runtime optimizations can be made, for example, using the TensorRT [37] library to accelerate the model inference, or by improving the code efficiency in pre-processing and post-processing stages.

**Table 6.** Experiment results of real-time performance.

Image Input Size	Average Pre-Processing Time (ms)	Average Inference Time (ms)	Average Post-Processing Time (ms)	Average Total Process Time (ms)	FPS
416 × 416	1.6	40.8	1.3	43.7	22.88
608 × 608	2.1	41.0	1.3	44.4	22.52
640 × 640	2.2	41.3	1.4	44.9	22.27
1216 × 1216	3.1	51.9	49.3	104.3	9.59

## 5. Conclusions

Aimed at challenges such as small-scale objects, untrained objects during inference, and real-time performance requirements, we designed a detector to detect small objects in UAV reconnaissance images. To conduct our research, we collected a small object dataset from the perspective of a high-flying UAV. We proposed two data augmentation methods, background replacement and noise adding, which improve the background diversity of the collected dataset. For the backbone design, we designed ADCSPDarkent53 based on the characteristics of small objects and evaluated the improved backbone on accuracy and speed. For object positioning, we modified the positioning loss function, which greatly improved detection accuracy. For object classification, a metric-based classification method was proposed to solve the problem of untrained sub-class object classification. Experiments on UAV-captured images and flight tests show the effectiveness and applicable scope of the proposed small-object detector. In the next step, improvements can be made in terms of dataset construction, feature selection and metric function design.

**Author Contributions:** Conceptualization, H.Z.; methodology, H.Z.; software, H.Z. and A.M.; validation, A.M.; formal analysis, H.Z. and A.M.; investigation, H.Z.; resources, Y.N.; data curation, H.Z. and A.M.; writing—original draft preparation, H.Z.; writing—review and editing, A.M.; visualization, H.Z. and A.M.; supervision, Y.N.; project administration, Z.M.; funding acquisition, Y.N. and Z.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper was supported by National Natural Science Foundation of China (No. 61876187) and Natural Science Foundation of Hunan Province (No. S2022JJQNJJ2084 and No. 2021JJ20054).

**Acknowledgments:** Thanks to the Unmanned Aerial Vehicles Teaching and Research Department, Institute of Unmanned Systems, College of Intelligence Science and Technology, National University of Defense Technology for providing the experimental platform.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Belmonte, L.M.; Morales, R.; Fernández-Caballero, A. Computer Vision in Autonomous Unmanned Aerial Vehicles—A Systematic Mapping Study. *Appl. Sci.* **2019**, *9*, 3196. [CrossRef]
2. Zhang, H.; Wang, L.; Tian, T.; Yin, J. A Review of Unmanned Aerial Vehicle Low-Altitude Remote Sensing (UAV-LARS) Use in Agricultural Monitoring in China. *Remote Sens.* **2021**, *13*, 1221. [CrossRef]
3. Zheng, Y.J.; Du, Y.C.; Ling, H.F.; Sheng, W.G.; Chen, S.Y. Evolutionary Collaborative Human-UAV Search for Escaped Criminals. *IEEE Trans. Evol. Comput.* **2020**, *24*, 217–231. [CrossRef]
4. Wu, X.; Li, W.; Hong, D.; Tao, R.; Du, Q. Deep Learning for Unmanned Aerial Vehicle-Based Object Detection and Tracking: A survey. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 91–124. [CrossRef]
5. Alexey, B.; Wang, C.; Mark Liao, H. Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
6. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
7. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 14–19 June 2020.
8. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.

9. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
10. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Proceedings of the Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.
11. Tong, K.; Wu, Y.; Zhou, F. Recent advances in small object detection based on deep learning: A review. *Image Vis. Comput.* **2020**, *97*, 103910. [CrossRef]
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
13. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
14. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
15. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
16. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
17. Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-up object detection by grouping extreme and center points. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Long Beach, CA, USA, 15–20 June 2019; pp. 850–859.
18. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 213–229.
19. Beal, J.; Kim, E.; Tzeng, E.; Park, D.H.; Zhai, A.; Kislyuk, D. Toward transformer-based object detection. *arXiv* **2020**, arXiv:2012.09958.
20. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
21. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Augmentation for small object detection. *arXiv* **2019**, arXiv:1902.07296.
22. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3578–3587.
23. Schumann, A.; Sommer, L.; Klatte, J.; Schuchert, T.; Beyerer, J. Deep cross-domain flying object classification for robust UAV detection. In *Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy, 29 August–1 September 2017; pp. 1–6.
24. Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; Yan, S. Perceptual generative adversarial networks for small object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 21–26 July 2017; pp. 1222–1230.
25. Yundong, L.; Han, D.; Hongguang, L.; Zhang, X.; Zhang, B.; Zhifeng, X. Multi-block SSD based on small object detection for UAV railway scene surveillance. *Chin. J. Aeronaut.* **2020**, *33*, 1747–1755.
26. Liu, Y.; Yang, F.; Hu, P. Small-object detection in UAV-captured images via multi-branch parallel feature pyramid networks. *IEEE Access* **2020**, *8*, 145740–145750. [CrossRef]
27. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, Seoul, Korea, 27–28 October 2019.
28. Glenn, J.; Ayush, C.; Jirka, B.; Alex, S.; Yonghye, K.; Jiacong, F.; Tao, X.; Kalen, M.; Yifu, Z.; Colin, W.; et al. Ultralytics/Yolov5. Available online: <https://github.com/ultralytics/yolov5> (accessed on 1 April 2021).
29. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
30. Padilla, R.; Netto, S.L.; Da Silva, E.A. A survey on performance metrics for object-detection algorithms. In *Proceedings of the 2020 international conference on systems, signals and image processing (IWSSIP)*, Niteroi, Brazil, 1–3 July 2020; pp. 237–242.
31. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea, 27–28 October 2019; pp. 9627–9636.
32. Long, X.; Deng, K.; Wang, G.; Zhang, Y.; Dang, Q.; Gao, Y.; Shen, H.; Ren, J.; Han, S.; Ding, E.; et al. PP-YOLO: An effective and efficient implementation of object detector. *arXiv* **2020**, arXiv:2007.12099.
33. Huang, X.; Wang, X.; Lv, W.; Bai, X.; Long, X.; Deng, K.; Dang, Q.; Han, S.; Liu, Q.; Hu, X.; et al. PP-YOLOv2: A practical object detector. *arXiv* **2021**, arXiv:2104.10419.
34. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. PP-YOLOE: An evolved version of YOLO. *arXiv* **2022**, arXiv:2203.16250.
35. Yu, G.; Chang, Q.; Lv, W.; Xu, C.; Cui, C.; Ji, W.; Dang, Q.; Deng, K.; Wang, G.; Du, Y.; et al. PP-PicoDet: A Better Real-Time Object Detector on Mobile Devices. *arXiv* **2021**, arXiv:2111.00902.

36. Dollar, P.; Singh, M.; Girshick, R. Fast and Accurate Model Scaling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 924–932.
37. Rajeev, R.; Kevin, C.; Xiaodong, H.; Samurdhi, K.; Shuyue, L.; Ryan, M.; Gwena, C.; Vinh, N.; Boris, F.; Paul, B.; et al. TensorRT Open Source Software. Available online: <https://github.com/NVIDIA/TensorRT> (accessed on 2 October 2022).

## Article

# Elliptical Multi-Orbit Circumnavigation Control of UAVS in Three-Dimensional Space Depending on Angle Information Only

Zhen Wang <sup>1,2</sup> and Yanhong Luo <sup>1,2,\*</sup> 

<sup>1</sup> The State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110004, China

<sup>2</sup> School of Information Science and Engineering, Northeastern University, Shenyang 110004, China

\* Correspondence: luoyanhong@ise.neu.edu.cn; Tel.: +86-024-8368-6470

**Abstract:** In order to analyze the circumnavigation tracking problem in complex three-dimensional space, in this paper, we propose a UAV group circumnavigation control strategy, in which the UAV circumnavigation orbit is an ellipse whose size can be adjusted arbitrarily; at the same time, the UAV group can be assigned to multiple orbits for tracking. The UAVs only have the angle information of the target, and the position information of the target can be obtained by using the angle information and the proposed three-dimensional estimator, thereby establishing an ideal relative velocity equation. By constructing the error dynamic equation between the actual relative velocity and the ideal relative velocity, the circumnavigation problem in three-dimensional space is transformed into a velocity tracking problem. Since the UAVs are easily disturbed by external factors during flight, the sliding mode control is used to improve the robustness of the system. Finally, the effectiveness of the control law and its robustness to unexpected situations are verified by simulation.

**Keywords:** three-dimensional circumnavigation control; elliptical multi-orbit; UAV group

**Citation:** Wang, Z.; Luo, Y. Elliptical Multi-Orbit Circumnavigation Control of UAVS in Three-Dimensional Space Depending on Angle Information Only. *Drones* **2022**, *6*, 296. <https://doi.org/10.3390/drones6100296>

Academic Editors: Daobo Wang and Zain Anwar Ali

Received: 30 August 2022

Accepted: 6 October 2022

Published: 10 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the past decade, the problem of UAV control has attracted more and more attention [1–3]. Since the multi-agent control problem has received continuous attention [4,5], multi-UAV control is also a hot area of research [6–8]. The multi-UAV circumnavigation control problem is a field of application for multi-UAV control, which refers to a group of UAVs surrounding the target in a prescribed formation to perform tasks such as monitoring [9], tracking and rounding up [10].

The circumnavigation problem has gradually developed from a single-agent circumnavigation static target [11–13] to a single-agent circumnavigation dynamic target [14–16], and up to the current multi-agent circumnavigation dynamic target [4,17,18]. Ref. [19] analyzed the circumnavigation problem in a two-dimensional plane, assuming that each agent has a fixed velocity. Ref. [20] used the self-propelled particle system to analyze the circumnavigation problem, but it required the circumnavigation target to be a static target. Most of the initial research assumed that each agent can accurately know the position information of the target [17,21], but this is difficult to achieve in practice. In order to overcome this drawback, researchers have proposed the use of distance measurement to obtain the position of the target. In [22], by measuring the distance between the agent and the target, the backstepping control is used to analyze the circumnavigation problem. In [23], a sliding mode control method based on distance measurement is designed to enable the UAV to achieve circumnavigation without using GPS. In practice, achieving distance measurement requires high-accuracy and expensive sensors, but UAVs are small and have limited payload capacity, so this approach is not realistic for UAVs. In contrast, angle measurement is relatively easy to implement. UAVs only need to carry cameras or other small sensors to measure angles.

In [24], the kinematic model of the underwater vehicle is combined with the dynamic model, and a cascade-based distributed circumnavigation control system is proposed, which can realize the circumnavigation of static and dynamic targets. In [25], the navigation control is realized by using the angle information between adjacent UAVs and the angle information between the UAV and the target. According to the local information of the target, ref. [26] uses the estimation method to estimate the location information of the target, so as to achieve circumnavigation, and this method does not need to consider the initial state of the agent. Refs. [27,28] propose a wireless speed measurement method to keep the multi-agent formation in the desired formation. Ref. [29] proposes a swarm control strategy for fixed-wing UAVs using multi-objective reinforcement learning. Taking the underwater robot as the research object, ref. [30] uses fast non-singular terminal sliding technology to provide faster convergence for full-drive underwater robot trajectory tracking control, while using adaptive technology to remove the restrictions that require system parameters.

However, the above studies were all conducted in the two-dimensional plane, but the UAV is active in the complex three-dimensional space; it is obviously more challenging to analyze the circumnavigation problem in three-dimensional space. At the same time, it should be noted that the formations proposed in these works are generally circular formations, which limits their practical application. In this paper, we set the formation to be elliptical and allowed the UAVs to be deployed in different orbits. The main contributions of this paper are summarized as follows:

- (1) A circumnavigation control law in three-dimensional space using only angle information is proposed, and an estimation method is used to obtain the position information of the target. In this way, the limitation of requiring knowledge of both target position information and angle information at the same time is eliminated.
- (2) The circumnavigation trajectory is set as an ellipse instead of being limited to a circular trajectory, and the major and minor semi-axes of the ellipse can be arbitrarily set. At the same time, UAVs can be deployed on multiple orbits by setting different coefficients.
- (3) Using the dynamic equation of the UAV, the three-dimensional position estimator and the adjustable elliptical orbit, the relative ideal velocity equation is designed, and by constructing the dynamic error between the ideal relative velocity and the actual velocity, the circumnavigation control problem is transformed into the tracking problem of relative velocity. At the same time, by adopting sliding mode control, the robustness of the system is greatly improved, and the stability of the system is proved by the Lyapunov method.

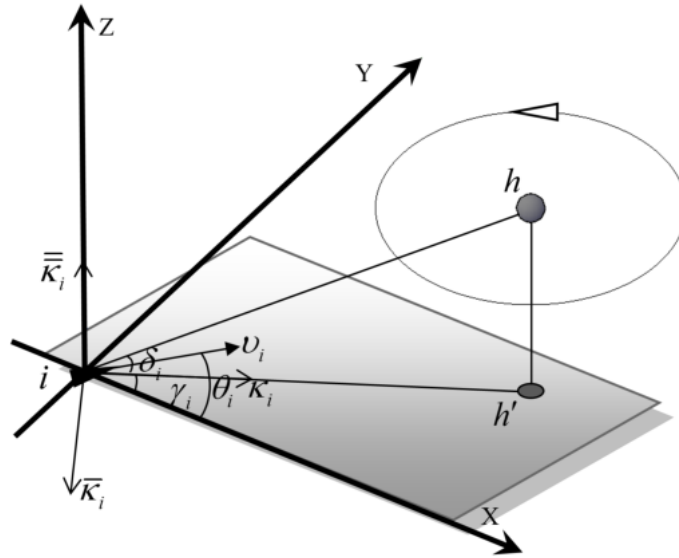
## 2. Problem Formulation

### 2.1. Define the Desired Angle

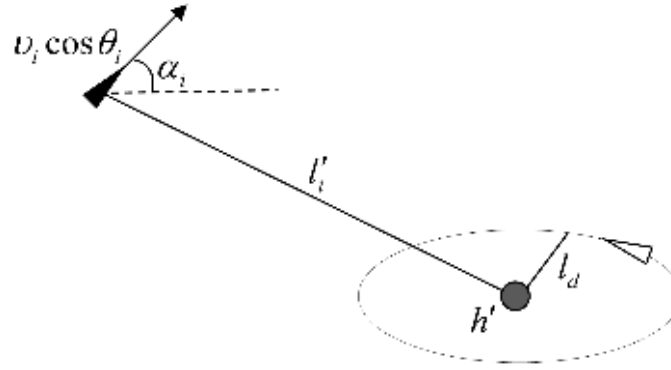
In the three-dimensional space, UAV  $i$  moves with speed  $v_i$ , and a space coordinate system is established with the UAV  $i$  as the origin, as shown in Figures 1 and 2. The target is denoted by  $h$ , and its projection point on the  $i$ -plane is denoted by  $h'$ . The distance between  $i$  and  $h'$  is denoted by  $l'_i$ , and  $l_d$  is the desired radius, that is, the desired distance between the UAV  $i$  and the target projection point  $h'$ .

**Definition 1.**  $\delta_i$  is the angle between  $ih$  and  $ih'$ .  $\gamma_i$  is the angle formed by  $ih'$  and the positive direction of the  $X$ -axis.  $\theta_i$  is the pitch angle of  $i$ , and  $\alpha_i$  is the heading angle of  $i$ . Next, make the following assumption:

**Assumption 1.** Assume that every UAV can know the angles in Definition 1, and the communication between UAVs is realized under a cyclic directed graph, i.e., UAV  $i + 1$  can get the information of UAV  $i$ ; at the same time, there is no interference in the communication between UAVs.



**Figure 1.** Schematic diagram of three-dimensional space circumnavigation.



**Figure 2.** Top view of circumnavigation.

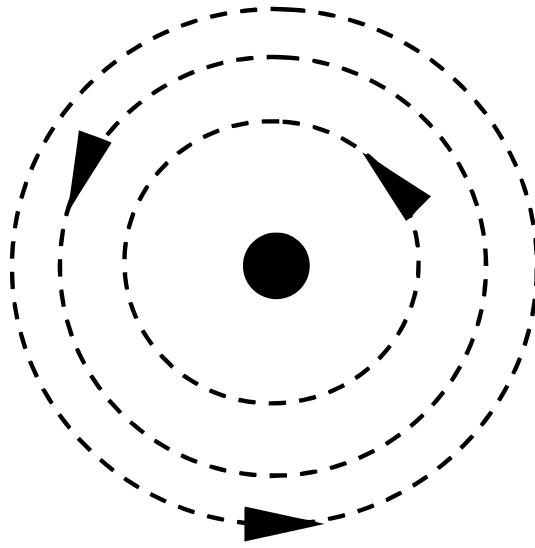
**Definition 2.** Define a set of orthogonal vectors  $[\kappa_i, \bar{\kappa}_i, \bar{\bar{\kappa}}_i]$  as shown in Figure 1, where  $\kappa_i$  represents the unit vector pointed by the  $i$  to  $h'$ ;  $\bar{\kappa}_i$  represents the unit vector in the XOY plane, perpendicular to  $\kappa_i$ ; and  $\bar{\bar{\kappa}}_i$  represents the vector perpendicular to the XOY plane.

$$\begin{cases} \kappa_i = [\cos \gamma_i & \sin \gamma_i & 0]^T, \\ \bar{\kappa}_i = [\sin \gamma_i & -\cos \gamma_i & 0]^T, \\ \bar{\bar{\kappa}}_i = [0 & 0 & \sin \delta_i]^T. \end{cases} \quad (1)$$

## 2.2. Multi-Orbit Circumnavigation

The multi-orbit circumnavigation is shown in Figure 3. For agent  $i$ , its required circumnavigation radius at time  $t$  should be set as  $l_i(t) = \tau_i l_d$ .  $l_i(t)$  represents the actual distance between the UAV  $i$  and the target projection point.  $\tau_i$  and  $l_d$  are positive real numbers, and by adjusting different  $\tau_i$ , different circumnavigation radii can be obtained.

Therefore, in the multi-orbit circumnavigation, different orbits are realized according to the given basic circumnavigation radius  $l_d$  and multiplied by the corresponding  $\tau_i$ . The relationship between different orbits is based on the multiple relationship of  $l_d$ .



**Figure 3.** Multi-orbit circumnavigation.

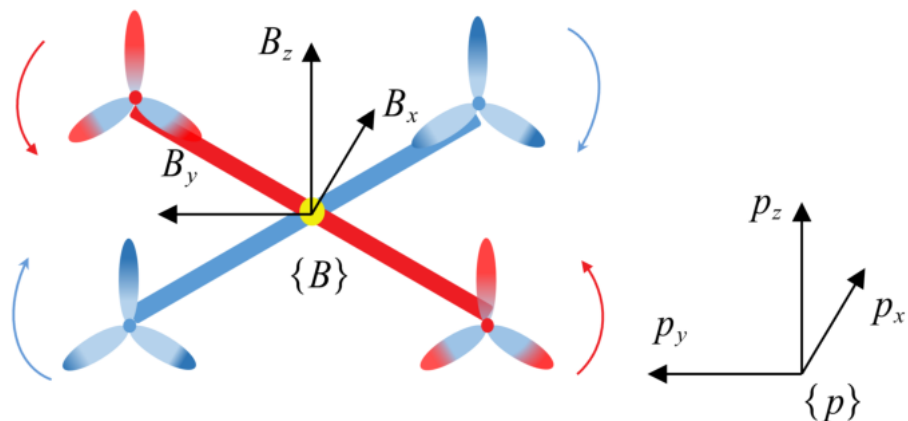
### 2.3. Dynamic Model of UAVs

First, we make the following assumption:

**Assumption 2.** In this paper, the quadrotor UAV is taken as the research object. Next, the quadrotor UAV dynamic model in the inertial coordinate system can be described as [31]

$$\begin{cases} \ddot{x}_i = \frac{F_i}{m_i} (\cos \psi_i \cos \phi_i \sin \theta_i + \sin \psi_i \sin \phi_i) + d_{xi}, \\ \ddot{y}_i = \frac{F_i}{m_i} (\sin \psi_i \cos \phi_i \sin \theta_i + \cos \psi_i \sin \phi_i) + d_{yi}, \\ \ddot{z}_i = \frac{F_i}{m_i} (\cos \phi_i \cos \theta_i) - g + d_{zi}, \end{cases} \quad (2)$$

where  $m_i$  represents the mass of the UAV. Defining  $\mathbf{p}_i = [x_i, y_i, z_i]^T$  indicates the position of the  $i$ -th UAV in the inertial coordinate system, and  $\mathbf{B}_i = [\psi_i, \theta_i, \phi_i]^T$  indicates the roll, pitch and yaw of the UAV in the body coordinate system. The relationship between the inertial coordinate system and the body coordinate system is given in Figure 4.  $F_i$  is the total thrust generated by the motors of the UAV, and  $g$  is the acceleration of gravity.  $\mathbf{d}_i = [d_{xi}, d_{yi}, d_{zi}]^T$  indicates system external interference.



**Figure 4.** Quadcopter UAV coordinate system.

In order to convert the coordinates of the UAV from the body coordinate system to the inertial coordinate system, we need the following rotation matrix  $\boldsymbol{\vartheta}_i$

$$\boldsymbol{\vartheta}_i = \begin{bmatrix} \cos \theta_i \cos \psi_i & \sin \theta_i \cos \psi_i \sin \phi_i - \sin \psi_i \cos \phi_i & \sin \theta_i \cos \psi_i \cos \phi_i + \sin \psi_i \sin \phi_i \\ \cos \theta_i \sin \psi_i & \sin \theta_i \sin \psi_i \sin \phi_i + \cos \psi_i \cos \phi_i & \sin \theta_i \sin \psi_i \cos \phi_i - \cos \psi_i \sin \phi_i \\ -\sin \theta_i & \cos \theta_i \sin \phi_i & \cos \theta_i \cos \phi_i \end{bmatrix}. \quad (3)$$

After some transformations, the dynamics model can be written as in [32]

$$m_i \begin{bmatrix} \ddot{p}_{xi} \\ \ddot{p}_{yi} \\ \ddot{p}_{zi} \end{bmatrix} = \boldsymbol{\vartheta}_i F_i + \begin{bmatrix} 0 \\ 0 \\ -m_i g \end{bmatrix} + m_i \begin{bmatrix} d_{xi} \\ d_{yi} \\ d_{zi} \end{bmatrix}, \quad (4)$$

where  $T_i$  is a continuous variable that represents the external input of UAV  $i$ , and can be written as

$$T_i = m_i \boldsymbol{\vartheta}_i^T [u_{xi}, u_{yi}, u_{zi} + g]^T, \quad (5)$$

where  $\mathbf{u}_i = [u_{xi}, u_{yi}, u_{zi}]^T$  represents the components of the control force of the UAV  $i$  on the three axes in the inertial coordinate system.

Next, combining (2), (4) and (5), a new expression can be obtained

$$\begin{aligned} F_i = & m_i u_{xi} (\cos \psi_i \cos \phi_i \sin \theta_i + \sin \psi_i \sin \phi_i)^{-1} \\ & + m_i u_{yi} (\sin \psi_i \cos \phi_i \sin \theta_i + \cos \psi_i \sin \phi_i)^{-1} \\ & + m_i u_{zi} (\cos \phi_i \cos \theta_i)^{-1}. \end{aligned} \quad (6)$$

From Equation (6), it can be seen that by designing the control  $\mathbf{u}_i = [u_{xi}, u_{yi}, u_{zi}]^T$ , the UAV can be controlled by applying the external input  $T_i$  and the attitude angle  $\mathbf{B}_i = [\psi_i, \theta_i, \phi_i]^T$ , so as to realize the circumnavigation control.

#### 2.4. Control Objective

Given a group of  $n$  UAVs and a moving target  $h$ , the purpose of this paper is to design a control law to make the UAV group around the given target in elliptical orbits. Ultimately, the UAV group needs to meet the following conditions:

- (1) The UAV group should circumnavigate the target on multiple elliptical orbits, so the circumnavigation radius of each UAV is different.
- (2) During the circumnavigation process, the circumnavigation angular velocity of the UAV group should be consistent.
- (3) The angular spacing between adjacent UAVs should remain unchanged.

The above objective can be written in the following form:

$$\begin{cases} \lim_{t \rightarrow +\infty} l_i(t) = \tau_i l_d, \\ \lim_{t \rightarrow +\infty} \dot{\alpha}_i = \omega_d, \\ \lim_{t \rightarrow +\infty} \alpha_i - \alpha_j = \beta_d, \end{cases} \quad \forall i, j \in N, \quad (7)$$

where  $N = \{1, \dots, n\}$ ,  $\omega_d$  represents the desired circumnavigation angular velocity;  $\beta_d$  is the desired angular spacing between two UAVs, with a value of  $2\pi/n$ ; and  $n$  is the number of UAVs. This means that at time  $t \rightarrow +\infty$ , the angular spacing between adjacent UAVs will be a fixed value.

### 3. Circumnavigation Control

Due to the target coordinate position, an estimator is needed to estimate the position of the target. We use the following estimator

$$\begin{aligned}\dot{\hat{\mathbf{p}}}_{ti} &= k(\mathbf{I} - \boldsymbol{\varepsilon}(t)\boldsymbol{\varepsilon}(t)^T)(\mathbf{p}_i - \hat{\mathbf{p}}_{ti}), \\ \boldsymbol{\varepsilon}(t) &= \frac{\mathbf{p}_t - \mathbf{p}_i}{\|\mathbf{p}_t - \mathbf{p}_i\|} = \begin{bmatrix} \cos \delta_i \cos \gamma_i \\ \cos \delta_i \sin \gamma_i \\ \sin \delta_i \end{bmatrix},\end{aligned}\quad (8)$$

where  $k$  is a constant gain,  $\mathbf{p}_t = [x_t, y_t, z_t]^T$  is the coordinates of the target, and  $\hat{\mathbf{p}}_{ti} = [\hat{x}_{ti}, \hat{y}_{ti}, \hat{z}_{ti}]$  is the coordinate estimation of the target by the UAV  $i$ .  $\mathbf{I}$  is the identity matrix. Through this three-dimensional position estimator, the position of the target can be estimated with only the angle information known. The desired elliptical circumnavigation radius is set as follows

$$l_d = \frac{ab}{\sqrt{a^2 \sin^2 \gamma_i + b^2 \cos^2 \gamma_i}}, \quad (9)$$

where  $a$  and  $b$  are the major and minor axes of the ellipse, respectively. By setting different values for  $a$  and  $b$ , the desired elliptical orbit  $l_d$  can be obtained. Using  $l_d$  multiplied by constant  $\tau_i$ , we can obtain different elliptical orbits; then, we can cause each UAV circle to be in its own orbit. It is worth noting that when  $a = b$ , it is a circular orbit.

Next, we write the dynamic Equation in (4) in the following form:

$$\dot{\mathbf{p}}_i = \mathbf{s}_i, \dot{\mathbf{s}}_i = \mathbf{u}_i + \mathbf{d}_i. \quad (10)$$

At the same time, the angle in Definition 1 can be represented by known information and estimated values

$$\begin{cases} \delta_i = \arctan \frac{\hat{z}_{ti} - z_i}{\sqrt{(\hat{x}_{ti} - x_i)^2 + (\hat{y}_{ti} - y_i)^2}}, \\ \gamma_i = (-1)^\zeta \arccos \frac{\hat{x}_{ti} - x_i}{\sqrt{(\hat{x}_{ti} - x_i)^2 + (\hat{y}_{ti} - y_i)^2}} + 2\pi\zeta, \end{cases} \quad (11)$$

where

$$\zeta = \begin{cases} 0, & \hat{y}_{ti} < y_i, \\ 1, & \hat{y}_{ti} \geq y_i. \end{cases} \quad (12)$$

It can be seen that  $\gamma_i$  is discontinuous, but the derivative of  $\gamma_i$  with respect to time can be expressed in the following form

$$\dot{\gamma}_i = \frac{v_i \cos \theta_i \sin(\gamma_i - \alpha_i)}{l'_i(t)} - \omega_i, \quad (13)$$

where  $l'_i(t)$  is the distance between the UAV and the target projection point.

**Proof.** Suppose the motion of UAV  $i$  at time  $t$  is as shown in Figure 5; at this time, the relationship between the angles  $\gamma_i$  and  $\alpha_i$  is also marked in Figure 5. The velocity of  $i$  on the horizontal plane can be decomposed orthogonally into  $v_{ia}$  and  $v_{ib}$ ; then,  $v_{ia}$  can be written as:

$$v_{ia} = v_i \cos \theta_i \sin(\gamma_i - \alpha_i) \quad (14)$$

The derivative of  $\gamma_i + (2\pi - \alpha_i)$  can be written as:

$$\dot{\gamma}_i - \dot{\alpha}_i = \frac{v_{ia}}{l'_i(t)}. \quad (15)$$

Then, (14) and (15) can be combined to obtain the result in (13). The proof is finished.  $\square$

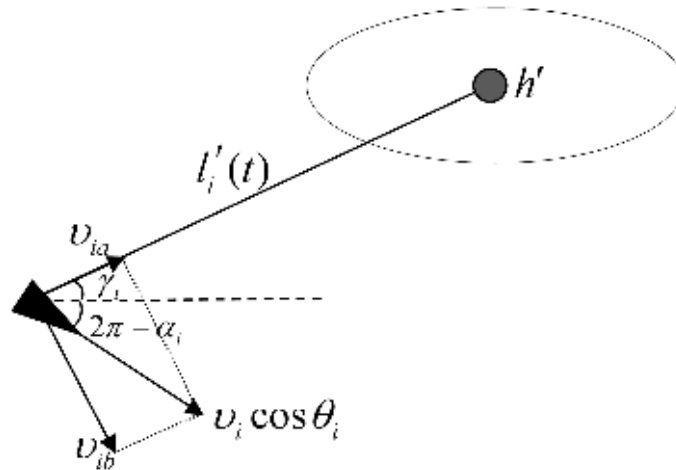


Figure 5. Top view of the UAV  $i$  at time  $t$ .

Next, the ideal relative velocity is constructed from the orthogonal vector in (1)

$$v_{id} = c_1(l'_i(t) - \tau_i l_d) \kappa_i + \tau_i l_d \omega_d (1 - \frac{2}{\pi} \arctan \beta_i) \bar{\kappa}_i + c_2 \bar{\kappa}_i, \quad (16)$$

where  $c_1 > 0, c_2 > 0, \beta_i = \alpha_i - \alpha_j - (i - j) \frac{2\pi}{n}$ .

$\sigma_{1i}$  is defined as the difference between the actual relative position and the ideal relative position of the UAV  $i$ , and  $\sigma_{2i}$  as the difference between the actual relative speed and the ideal relative speed of the UAV  $i$ ; thus, we can obtain

$$\begin{cases} \sigma_{1i} = p_i - p_t - \int v_{id} dt, \\ \dot{\sigma}_{1i} = \sigma_{2i} = \dot{p}_i - \dot{p}_t - v_{id}, \\ \dot{\sigma}_{2i} = u_i + d_i - \ddot{p}_t - \dot{v}_{id}. \end{cases} \quad (17)$$

In order to improve the robustness of the system, sliding mode control is adopted. The sliding surface is designed as

$$S = \chi \sigma_{1i} + \sigma_{2i}, \quad (18)$$

where  $\chi > 0$ , using the exponential reaching law as follows

$$\dot{S} = -c_3 S - c_4 \text{sgn}(S), \quad (19)$$

where  $c_3 > 0, c_4 > 0$ . Then, we obtain the following control law

$$u_i = -c_3 S - c_4 \text{sgn}(S) - \chi \sigma_{2i} + \ddot{p}_t + \dot{v}_{id} - d_i. \quad (20)$$

**Theorem 1.** Considering a group of UAVs as a nonlinear system (4), if the three-dimensional position estimators are set as (8), the ellipse surround radius is set as (9), and the parameters  $c_1, c_2, c_3, c_4$ , and  $\chi$  are selected to be greater than 0, then the controller (20) can make the group of UAVs circle the target on multiple elliptical orbits.

**Proof.** Considering the Lyapunov candidate function  $V = \frac{1}{2} S^2$ , we take its derivative with respect to time as

$$\begin{aligned} \dot{V} &= S \dot{S} \\ &= S(\chi \sigma_{2i} + \dot{\sigma}_{2i}) \\ &= S(\chi \sigma_{2i} + (u_i + d_i - \ddot{p}_t - \dot{v}_{id})). \end{aligned} \quad (21)$$

Next, we substitute (20) into (21)

$$\begin{aligned}\dot{V} &= S(\chi\sigma_{2i} - c_3S - c_4\text{sgn}(S) - \chi\sigma_{2i} + \ddot{p}_t + \dot{v}_{id} - \dot{d}_i - \ddot{p}_t - \dot{v}_{id} + \dot{d}_i) \\ &= S(-c_3S - c_4\text{sgn}(S)) \\ &= -c_3\|S\|^2 - c_4\|S\|,\end{aligned}\quad (22)$$

where  $c_3 > 0$ , and  $c_4 > 0$ ; therefore,  $\dot{V} \leq 0$  satisfies the Lyapunov stability criterion.  $\square$

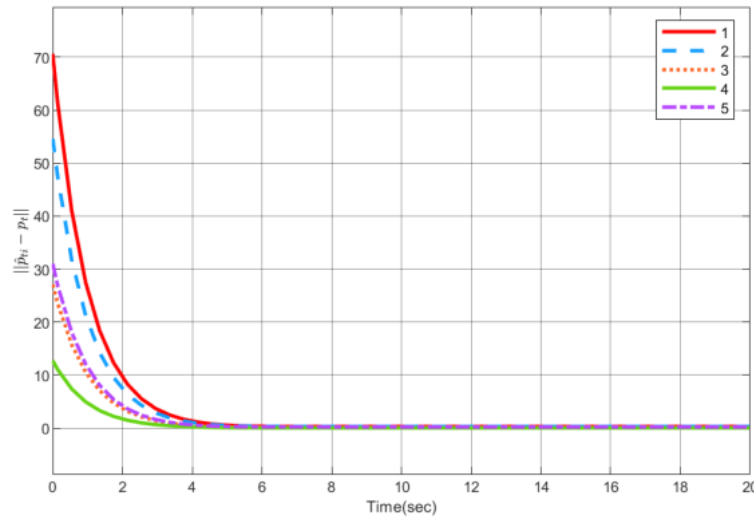
#### 4. Simulation Results

In this section, the performance of the estimator (8) and control law (20) is verified by considering a group of five UAVs circling a moving target on multiple trajectories, where  $\beta_d$  is  $2\pi/5$ . We set the mass of each UAV to  $m_i = 1$  kg, and the acceleration of gravity to  $g = 9.8$  m/s<sup>2</sup>. For the ideal relative velocity in (16),  $c_1 = 1$  and  $c_2 = 5$  were selected. Parameters in the controller (20) were selected as  $c_3 = 5$ ,  $c_4 = 10$ ,  $\chi = 1$ .

##### 4.1. Case 1: Target Moves in a Straight Line

The speed of the moving target in three-dimensional space is selected as  $\dot{p}_t = [2 \ 2 \ 4]^T$ .

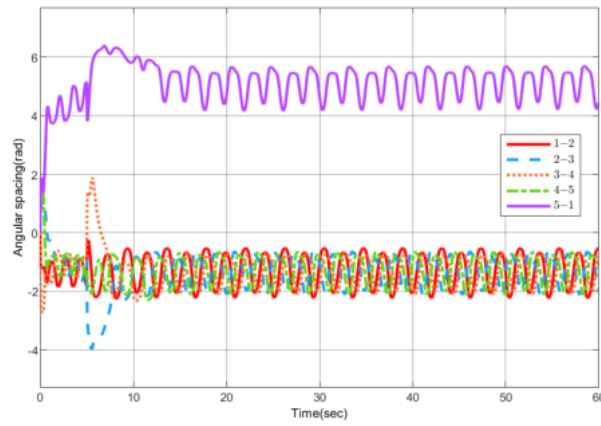
First, the parameters of the estimator are selected,  $k = 1$ , and the initial positions of the five UAVs are randomly generated. Figure 6 presents the simulation results of the convergence effect of the position estimator.



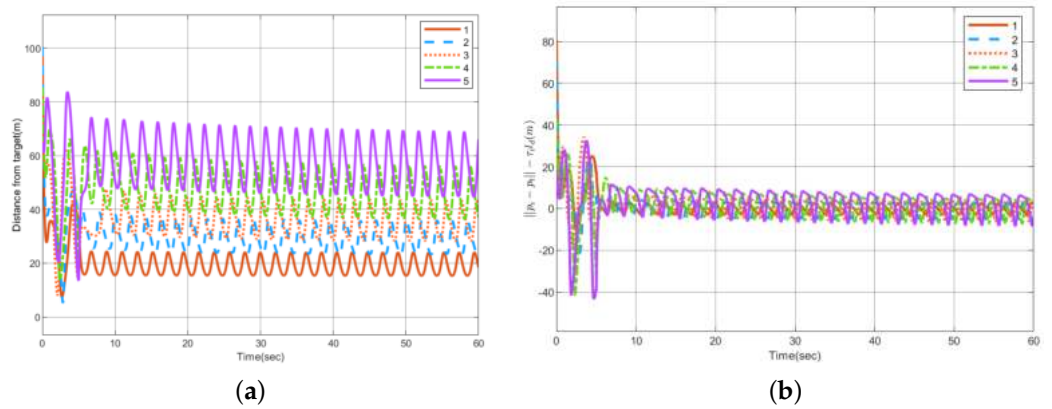
**Figure 6.** Position estimator error convergence  $\|\hat{p}_{ti} - p_t\|$ .

Next, when external interference is not considered, that is  $\dot{d}_i = [0, 0, 0]^T$ , the parameters of the elliptical orbit are taken as  $a = 25$  and  $b = 15$ , with  $\tau_i$  selected as  $\tau_i = [1, 1.5, 1.8, 2.3, 2.8]$ .

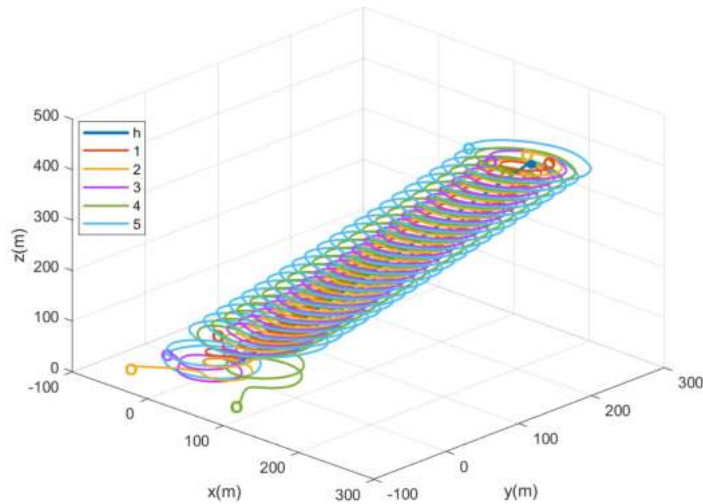
Figure 7 shows the angular spacing between two adjacent UAVs, Figure 8 gives the circumnavigation control error, and Figure 8a shows distance between each UAV and the target, that is  $\|p_i - p_t\|$ . Since the circumnavigation track is elliptical, the distance should change all the time. Figure 8b is  $\|p_i - p_t\| - \tau_i l_d$ , that is, the error between the actual distance and estimated distance from the position of UAV to target. The three-dimensional simulation diagram of the UAVs circumnavigation is given in Figure 9.



**Figure 7.** Angular spacing between adjacent UAVs.

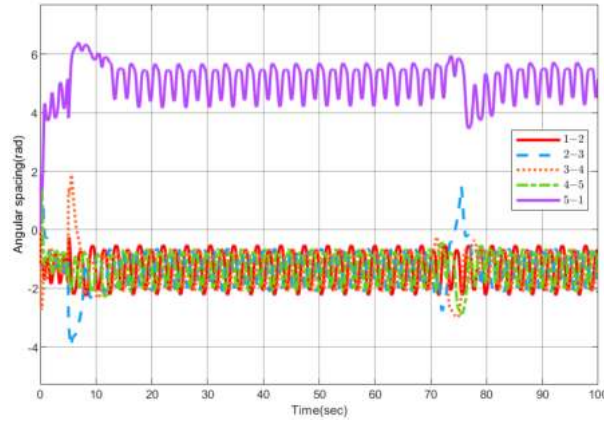


**Figure 8.** Circumnavigation control error. (a)  $\|p_i - p_t\|$ . (b)  $\|p_i - p_t\| - \tau_i l_d$ .

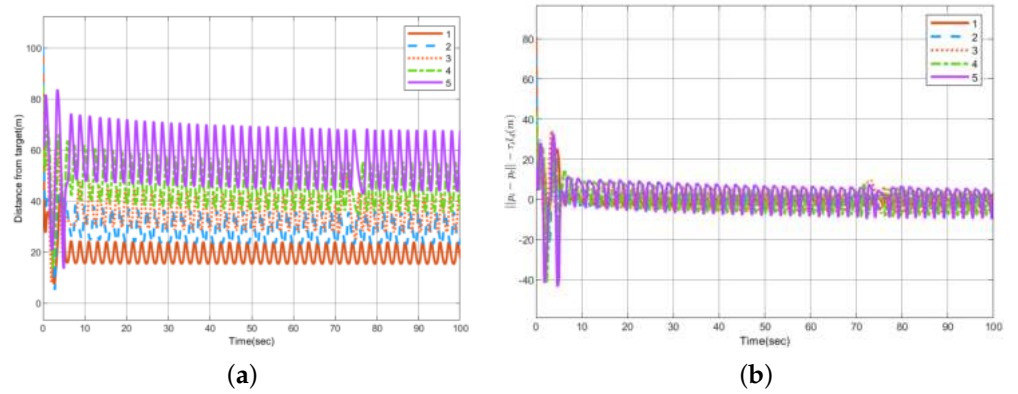


**Figure 9.** Three-dimensional diagram of circumnavigation control.

Since the UAV is easily interfered with by external factors during the operation, the robustness of the control system is tested. When  $t = 70$  s, the external interference of the third UAV is set to  $d_3 = [0, 30, 70]^T$ . Figure 10 shows the change in the angular spacing between each UAV, and Figure 11 shows the circumnavigation control error, where Figure 11a is  $\|p_i - p_t\|$ , and Figure 11b is  $\|p_i - p_t\| - \tau_i l_d$ . It can be seen from the above pictures that when there is interference, the sliding mode control can quickly eliminate the influence of the interference and maintain the formation of the UAV group.



**Figure 10.** Variation in angular separation between adjacent UAVs when interference occurs.

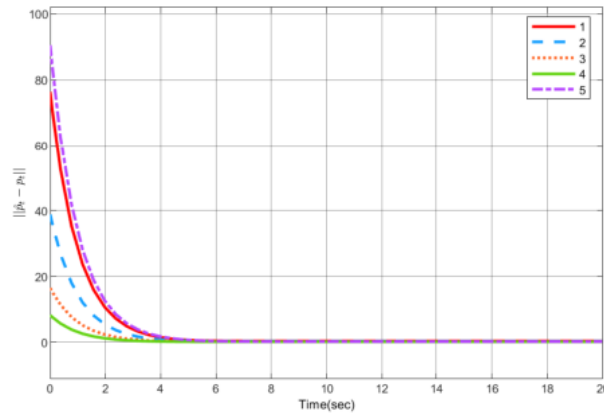


**Figure 11.** The circumnavigation control error when disturbance occurs. (a)  $\|p_i - p_t\|$ . (b)  $\|p_i - p_t\| - \tau_i l_d$ .

#### 4.2. Case 2: Target Moves in a Curve

The speed of the moving target in three-dimensional space is selected as  $\dot{p}_t = [3 \cos(0.5t) \ 3 \sin(0.5t) \ t]^T$ .

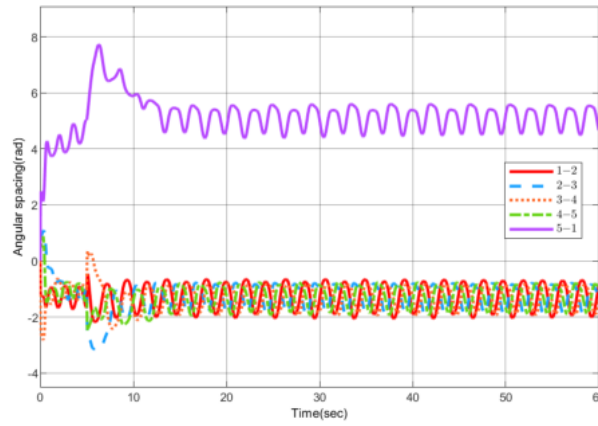
The parameters of the estimator are selected,  $k = 1$ , and the initial positions of the five UAVs are randomly generated. Figure 12 presents the simulation results of the convergence effect of the position estimator.



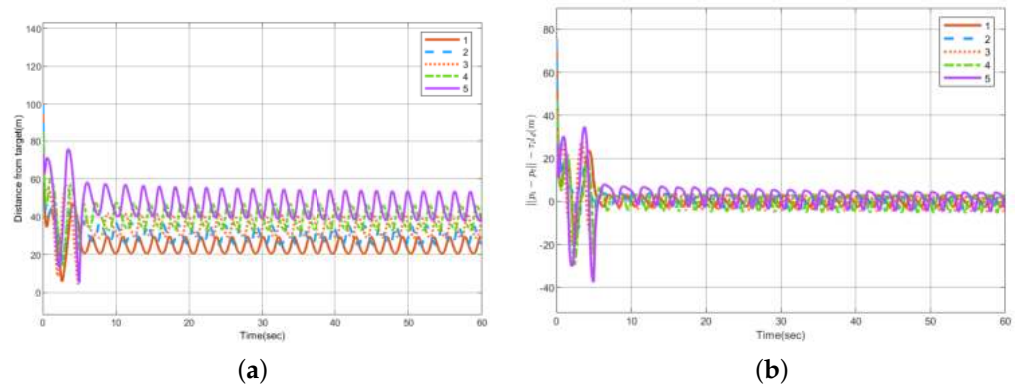
**Figure 12.** Position estimator error convergence  $\|\hat{p}_{ti} - p_t\|$ .

When  $d_i = [0, 0, 0]^T$ , we take the parameters of the elliptical orbit as  $a = 30$  and  $b = 20$ , with  $\tau_i$  selected as  $\tau_i = [1, 1.2, 1.4, 1.6, 1.8]$ .

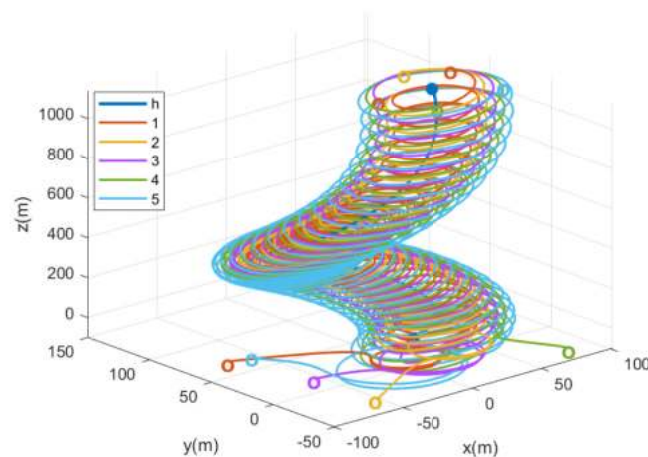
Figure 13 shows the angular spacing between two adjacent UAVs, Figure 14 gives the circumnavigation control error, and Figure 14a shows the distance between each UAV and the target, that is  $\|p_i - p_t\|$ . Figure 14b is  $\|p_i - p_t\| - \tau_i l_d$ . The three-dimensional simulation diagram of the UAVs circumnavigation is given in Figure 15.



**Figure 13.** Angular spacing between adjacent UAVs.

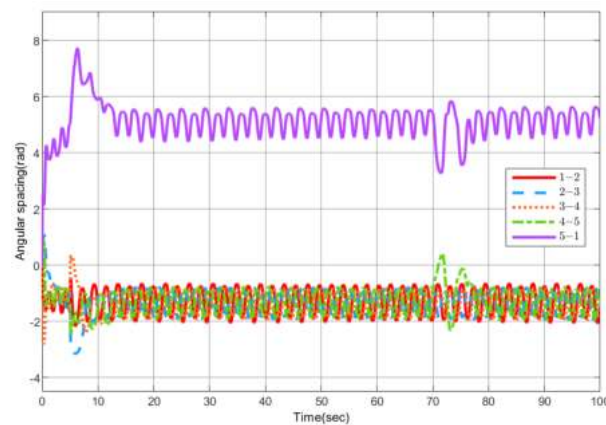


**Figure 14.** Circumnavigation control error. (a)  $\|p_i - p_t\|$ . (b)  $\|p_i - p_t\| - \tau_i l_d$ .

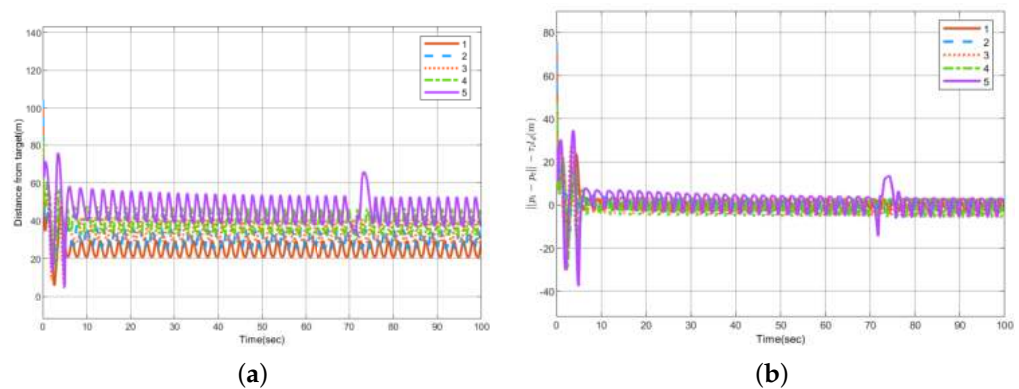


**Figure 15.** Three-dimensional diagram of circumnavigation control.

The robustness of the system is verified again. When  $t = 70$  s, the external interference of the fifth UAV is set to  $d_3 = [30, 10, 50]^T$ . Figure 16 shows the change in the angular spacing between each UAV, and Figure 17 shows the circumnavigation control error, where Figure 17a is  $\|p_i - p_t\|$ , and Figure 17b is  $\|p_i - p_t\| - \tau_i l_d$ . The above experiments once again prove that the sliding mode control adopted in this paper has strong robustness.



**Figure 16.** Variation in angular separation between adjacent UAVs when interference occurs.



**Figure 17.** The circumnavigation control error when disturbance occurs. (a)  $\|p_i - p_t\|$ . (b)  $\|p_i - p_t - \tau_i l_d\|$ .

## 5. Conclusions

In this paper, the problem of circumnavigation control in three-dimensional space is studied, the group of UAVs is made to circle the target in elliptical formation and with multiple orbits. The target is made to move with curvilinear variable speed, and UAVs estimate the position of the target only from the angle information. The error dynamic equation is constructed using the ideal relative velocity and the actual relative velocity, and the circumnavigation control is transformed into a velocity tracking problem. In order to improve the robustness of the system, sliding mode control is used to design the control law. Finally, the effectiveness of the proposed control law is proved by simulation, and disturbance is added to the simulation process to verify the robustness of sliding mode control.

**Author Contributions:** Conceptualization, Y.L. and Z.W.; methodology, Z.W.; software, Z.W.; validation, Y.L. and Z.W.; formal analysis, Z.W.; investigation, Z.W.; resources, Y.L.; data curation, Z.W.; writing—original draft preparation, Z.W.; writing—review and editing, Y.L.; visualization, Z.W.; supervision, Y.L.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (62173082), Shenyang Science & Technology Innovation Program for Young and Middle-aged Talents (RC210503), and the Liaoning Joint Fund of the National Natural Science Foundation of China (U1908217).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.




## References

1. Luo, Y.; Yu, X.; Yang, D.; Zhou, B. A survey of intelligent transmission line inspection based on unmanned aerial vehicle. *Artif. Intell. Rev.* **2022**, 1–29. [CrossRef]
2. Israr, A.; Ali, Z.A.; Alkhamash, E.H.; Jussila, J.J. Optimization Methods Applied to Motion Planning of Unmanned Aerial Vehicles: A Review. *Drones* **2022**, 6, 126. [CrossRef]
3. Ming, Z.; Huang, H. A 3d vision cone based method for collision free navigation of a quadcopter UAV among moving obstacles. *Drones* **2021**, 5, 134. [CrossRef]
4. Lan, Y.; Lin, Z.; Cao, M.; Yan, G. A distributed reconfigurable control law for escorting and patrolling missions using teams of unicycles. In Proceedings of the 49th IEEE Conference on Decision and Control, Atlanta, GA, USA, 15–17 December 2010; pp. 5456–5461.
5. Zhang, Y.; Wen, Y.; Li, F.; Chen, Y. Distributed observer-based formation tracking control of multi-agent systems with multiple targets of unknown periodic inputs. *Unmanned Syst.* **2019**, 7, 15–23. [CrossRef]
6. Zhang, M.; Jia, J.; Mei, J. A composite system theory-based guidance law for cooperative target circumnavigation of UAVs. *Aerosp. Sci. Technol.* **2021**, 118, 107034. [CrossRef]
7. Le, W.; Xue, Z.; Chen, J.; Zhang, Z. Coverage Path Planning Based on the Optimization Strategy of Multiple Solar Powered Unmanned Aerial Vehicles. *Drones* **2022**, 6, 203. [CrossRef]
8. Yan, J.; Yu, Y.; Wang, X. Distance-Based Formation Control for Fixed-Wing UAVs with Input Constraints: A Low Gain Method. *Drones* **2022**, 6, 159. [CrossRef]
9. Leonard, N.E.; Paley, D.A.; Lekien, F.; Sepulchre, R.; Fratantoni, D.M.; Davis, R.E. Collective motion, sensor networks, and ocean sampling. *Proc. IEEE* **2007**, 95, 48–74. [CrossRef]
10. Kothari, M.; Sharma, R.; Postlethwaite, I.; Beard, R.W.; Pack, D. Cooperative target-capturing with incomplete target information. *J. Intell. Robot. Syst.* **2013**, 72, 373–384. [CrossRef]
11. Sepulchre, R.; Paley, D.A.; Leonard, N.E. Stabilization of planar collective motion: All-to-all communication. *IEEE Trans. Autom. Control* **2007**, 52, 811–824. [CrossRef]
12. Sepulchre, R.; Paley, D.A.; Leonard, N.E. Stabilization of planar collective motion with limited communication. *IEEE Trans. Autom. Control* **2008**, 53, 706–719. [CrossRef]
13. Marshall, J.A.; Broucke, M.E.; Francis, B.A. Formations of vehicles in cyclic pursuit. *IEEE Trans. Autom. Control* **2004**, 49, 1963–1974. [CrossRef]
14. Deghat, M.; Shames, I.; Anderson, B.D.; Yu, C. Localization and circumnavigation of a slowly moving target using bearing measurements. *IEEE Trans. Autom. Control* **2014**, 59, 2182–2188. [CrossRef]
15. Wang, J.; Ma, B.; Yan, K. Mobile Robot Circumnavigating an Unknown Target Using Only Range Rate Measurement. *IEEE Trans. Circ. Syst. I Express Briefs* **2021**, 69, 2. [CrossRef]
16. Greiff, M.; Deghat, M.; Sun, Z.; Robertsson, A. Target Localization and Circumnavigation with Integral Action in  $\mathbb{R}^2$ . *IEEE Control Syst. Lett.* **2021**, 6, 1250–1255. [CrossRef]
17. Summers, T.H.; Akella, M.R.; Mears, M.J. Coordinated standoff tracking of moving targets: Control laws and information architectures. *J. Guid. Control Dyn.* **2009**, 32, 56–69. [CrossRef]
18. Huo, M.; Duan, H.; Fan, Y. Pigeon-inspired circular formation control for multi-UAV system with limited target information. *Guid. Navig. Control* **2021**, 1, 2150004. [CrossRef]
19. Seyboth, G.S.; Wu, J.; Qin, J.; Yu, C.; Allgöwer, F. Collective circular motion of unicycle type vehicles with nonidentical constant velocities. *IEEE Trans. Control Netw. Syst.* **2014**, 1, 167–176. [CrossRef]
20. Hernandez, S.; Paley, D.A. Three-dimensional motion coordination in a spatiotemporal flowfield. *IEEE Trans. Autom. Control* **2010**, 55, 2805–2810. [CrossRef]
21. Kim, T.H.; Hara, S.; Hori, Y. Cooperative control of multi-agent dynamical systems in target-enclosing operations using cyclic pursuit strategy. *Int. J. Control* **2010**, 83, 2040–2052. [CrossRef]
22. Dong, F.; You, K.; Song, S. Target encirclement with any smooth pattern using range-based measurements. *Automatica* **2020**, 116, 108932. [CrossRef]
23. Cao, Y. UAV circumnavigating an unknown target under a GPS-denied environment with range-only measurements. *Automatica* **2015**, 55, 150–158. [CrossRef]
24. Shi, L.; Zheng, R.; Liu, M.; Zhang, S. Distributed circumnavigation control of autonomous underwater vehicles based on local information. *Syst. Control Lett.* **2021**, 148, 104873. [CrossRef]
25. Yu, X.; Liu, L. Cooperative control for moving-target circular formation of nonholonomic vehicles. *IEEE Trans. Autom. Control* **2016**, 62, 3448–3454. [CrossRef]
26. Sen, A.; Sahoo, S.R.; Kothari, M. Circumnavigation on multiple circles around a nonstationary target with desired angular spacing. *IEEE Trans. Cybern.* **2019**, 51, 222–232. [CrossRef]
27. Mehrabian, A.R.; Tafazoli, S.; Khorasani, K. *Coordinated Attitude Control of Spacecraft Formation without Angular Velocity Feedback: A Decentralized Approach*; Aerospace Research Central: Lawrence, KS, USA, 2009; p. 6289.

28. Chen, Y.; Liang, J.; Miao, Z.; Wang, Y. Distributed Formation Control of Quadrotor UAVs Based on Rotation Matrices without Linear Velocity Feedback. *Int. J. Control Autom. Syst.* **2021**, *19*, 3464–3474. [CrossRef]
29. Speck, C.; Bucci, D.J. Distributed uav swarm formation control via object-focused, multi-objective sarsa. In Proceedings of the 2018 Annual American Control Conference, Milwaukee, WI, USA, 27–29 June 2018; pp. 6596–6601.
30. Qiao, L.; Zhang, W. Adaptive second-order fast nonsingular terminal sliding mode tracking control for fully actuated autonomous underwater vehicles. *IEEE J. Ocean. Eng.* **2018**, *44*, 363–385. [CrossRef]
31. Bai, A.; Luo, Y.; Zhang, H.; Li, Z. L2-gain robust trajectory tracking control for quadrotor UAV with unknown disturbance. *Asian J. Control* **2021**, 1–13. [CrossRef]
32. Li, D.; Cao, K.; Kong, L.; Yu, H. Fully Distributed Cooperative Circumnavigation of Networked Unmanned Aerial Vehicles. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 709–718. [CrossRef]

## Review

# Comprehensive Review of UAV Detection, Security, and Communication Advancements to Prevent Threats

Ghulam E. Mustafa Abro <sup>1,2,3,4,\*</sup> , Saiful Azrin B. M. Zulkifli <sup>1,3</sup> , Rana Javed Masood <sup>5</sup> ,  
Vijanth Sagayan Asirvadam <sup>3</sup> and Anis Laouti <sup>2</sup>

- <sup>1</sup> Center for Automotive Research and Electric Mobility (CAREM), Universiti Teknologi PETRONAS, Seri Iskandar 32610, Perak, Malaysia  
<sup>2</sup> Samovar, Telecom SudParis, CNRS, Institut Polytechnique de Paris, 9 Rue Charles Fourier, 91011 Paris, France  
<sup>3</sup> Electrical and Electronic Engineering Department, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Perak, Malaysia  
<sup>4</sup> Condition Monitoring Systems (CMS) Lab, NCRA, Mehran University of Engineering and Technology (MUET), Jamshoro 67480, Sindh, Pakistan  
<sup>5</sup> Electronic Engineering Department, Usman Institute of Technology (U.I.T.), Karachi 75300, Sindh, Pakistan  
\* Correspondence: ghulam\_20000150@utp.edu.my or mustafa.abro@ieee.org

**Abstract:** It has been observed that unmanned aerial vehicles (UAVs), also known as drones, have been used in a very different way over time. The advancements in key UAV areas include detection (including radio frequency and radar), classification (including micro, mini, close range, short range, medium range, medium-range endurance, low-altitude deep penetration, low-altitude long endurance, and medium-altitude long endurance), tracking (including lateral tracking, vertical tracking, moving aerial pan with moving target, and moving aerial tilt with moving target), and so forth. Even with all of these improvements and advantages, security and privacy can still be ensured by researching a number of key aspects of an unmanned aerial vehicle, such as through the jamming of the control signals of a UAV and redirecting them for any high-assault activity. This review article will examine the privacy issues related to drone standards and regulations. The manuscript will also provide a comprehensive answer to these limitations. In addition to updated information on current legislation and the many classes that can be used to establish communication between a ground control room and an unmanned aerial vehicle, this article provides a basic overview of unmanned aerial vehicles. After reading this review, readers will understand the shortcomings, the most recent advancements, and the strategies for addressing security issues, assaults, and limitations. The open research areas described in this manuscript can be utilized to create novel methods for strengthening the security and privacy of an unmanned aerial vehicle.

**Keywords:** unmanned aerial vehicle; advancement; classification; tracking and communication threats

**Citation:** Abro, G.E.M.; Zulkifli, S.A.B.M.; Masood, R.J.; Asirvadam, V.S.; Laouti, A. Comprehensive Review of UAV Detection, Security, and Communication Advancements to Prevent Threats. *Drones* **2022**, *6*, 284. <https://doi.org/10.3390/drones6100284>

Academic Editor: Diego González-Aguilera

Received: 22 August 2022

Accepted: 11 September 2022

Published: 1 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

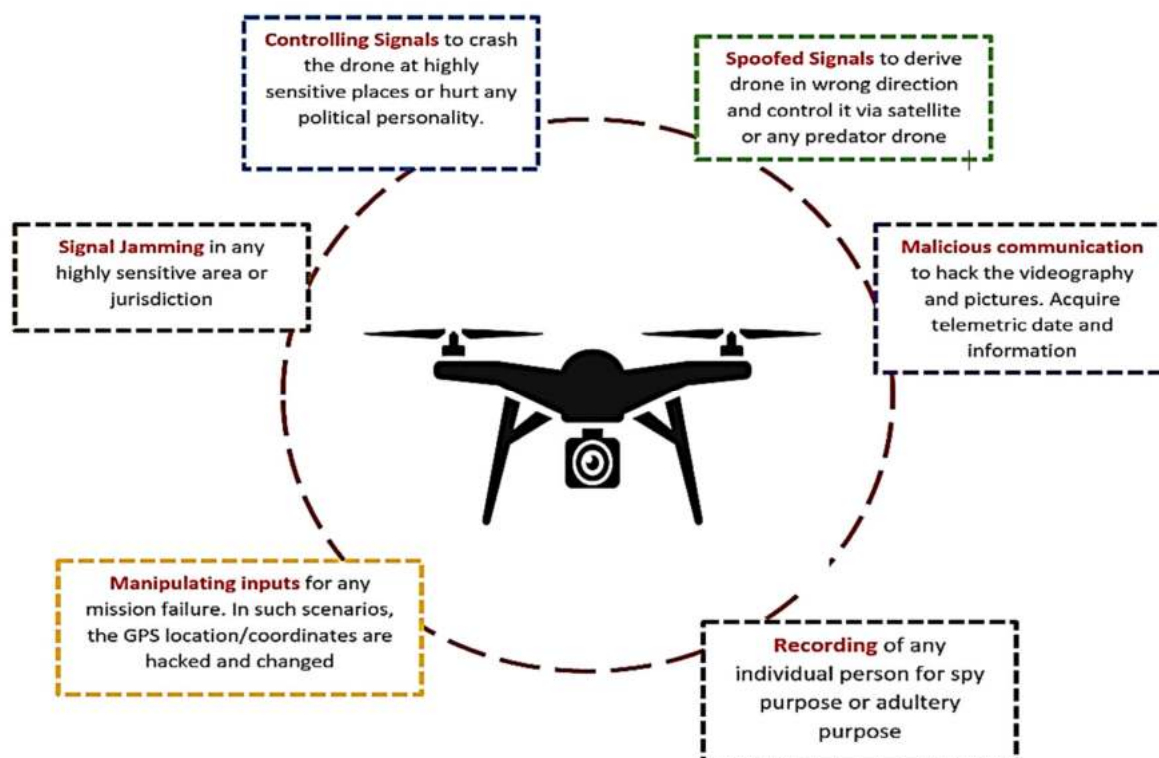


**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Technology has advanced, and as a result, the world of today has seen a number of ground-breaking developments. These outcomes have been demonstrated to be more trustworthy, approachable, and economical in our everyday lives. In addition, people now engage with one another in novel ways in their social circles. Additionally, unmanned aerial vehicles (UAVs) are employed for both commercial and private purposes in addition to being heavily utilized in military contexts. The market potential for medium-sized drones has been estimated by the China Unmanned Aerial Vehicle Industry (CUAVI) to reach CNY 80 billion by 2025 [1], whereas the Federal Aviation Administration (FAA) concluded that there are currently 3 million drones flying in the US sky and that number will increase by four times by the end of 2022 [2]. Drone use is increasing because of its value in a variety of jobs, including through the live broadcasting of events, aerial video shoots, the mobility to move packages from one location to another, and simple navigation as shown

in Figure 1. These drones are commonly employed for transportation purposes due to their cheap maintenance requirements, ability to take-off and land vertically, ability to hover, and high degree of mobility. These drones are frequently outfitted with computer vision and internet of things (IoT)-like features, particularly for the swarming of drones [3,4], and they have proven to be an effective choice for surveillance and rescue-related missions [5]. There are, however, a few important elements that are connected to UAV security worries. This collection contains the story of the Iranian military jamming an American drone's control signals [6]; nonetheless, it is still difficult to create a security control module for UAVs that is completely foolproof. Additionally, the first drones were unmanned balloons loaded with explosives that were used to assault Venice in Italy [7]. Later, in 1915, the British troops employed these unmanned balloons for photographic-based surveillance during the renowned Battle of Neuve Chapelle [8]. During this time, cameras were not as advanced; hence, this strategy was suggested to improve visibility [8]. In order to find various terrorists, several of them were also used during the Afghan War [9,10]. Prior to today, these drones were usually utilized for military operations, but they are now also chosen for the majority of domestic applications, to the point that Amazon began using drones to carry packages in 2014 [11]. They have also been utilized in fields including agriculture [1,2], for checking building sites [3], and to greatly assist law enforcement organizations with emergency rescue operations. The United States of America started making pilotless aircrafts that could maneuver for roughly a kilometer in the early 1910s. During World War II, the US started developing advanced UAV programs, such as the N2C-2 drone and the OQ-2 communications plane [9], but these endeavors were both expensive and unreliable. In the late 1980s, the US started developing sophisticated drones, and they already have some top-notch micro unmanned aerial vehicles. Drones are also being used in the media business for aerial photography and filmmaking. Drone use is expanding quickly, and at the same time, security and privacy issues have grown more complicated and serious.



**Figure 1.** Security and privacy threats of UAVs.

The major goal of this review article is to give readers a comprehensive understanding of the new advancements that have led to the issues surrounding unmanned aerial vehicles (UAVs), including security threats, privacy concerns, and other limits that are important and cannot be disregarded. To give readers a thorough understanding of the subject, the manuscript has been organized into 10 sections. The major goal is to identify these issues and give all scholars access to a single resource that will allow them to fully understand the most recent trends and work to advance their field.

Section 2 contains the regulatory standards, whereas Section 3 describes the classification of various unmanned aerial vehicles (UAVs). Section 4 of the document discusses the structures and techniques of communication. In Section 5, it is specifically mentioned how and why drones are used. Section 6 covers the key security challenges and weaknesses, whereas Section 7 covers the present constraints. The most recent methods to address these restrictions are also discussed in Section 8, along with open research areas and recommendations in Section 9. The thorough conclusion to this work can be found last, but by no means least, in Section 10.

## 2. Study Related to Regulations

Many countries have been following the standard regulations to ensure the security and privacy implications of drones. Many of them have started to propose several step-by-step procedures to license their UAVs [12,13]. If these regulations are not followed, then unlicensed drones are taken under custody and proper legal action is taken against the pilot [12]. As per the media news broadcasted by the British Broadcasting Corporation (BBC), the CAA and FAA have declared some standard operating procedures (SOPs) to maneuver the UAVs at a low altitude [14], which are mentioned below:

- The users or operators of a registered UAV must carry the proof of license while operating the UAV.
- The maximum height at which the UAV can maneuver is 400 feet only.
- UAVs must be kept away from the airfields and, in case of necessity, one may acquire the written permission from relevant boards or authorities.
  - In the case of a UAV crash, legal action can be taken against any harmful actions or the damage that occurred from UAV failure.
  - UAVs with computer vision or camera surveillance are not allowed to maneuver within 50 m of people or any crowd.
  - UAVs will be summoned if they are not flown within the operator's line of sight.
  - UAVs will be summoned if they are flown at night without proper lighting.

The above standard rules and SOPs ensure the secure operation of drones.

It is also noted that with a dramatic rise in the drone industry, various countries have inducted their own rules as well [15]. Mainly, to operate a UAV, there are three fundamental components: the first is the ground control room (GCR); the second one is the communication method, for example, satellite, radio frequency, etc., as illustrated in Figure 2; and last, but certainly not the least, is the UAV itself. There are three different methods to communicate with a drone, i.e., satellite, radio signal, and internet, as shown in Figure 2 [15]. The essential license-exempted radio equipment, along with the frequencies, is mentioned in Table 1 [16].

The standard bandwidth through which a communication is established between a UAV and the ground control room (GCR) is mentioned in Table 1, whereas the other standards are still in progress for the safe operations of a UAV in any vicinity [17].



**Figure 2.** Communication channels mostly used to control UAVs.

**Table 1.** Frequencies through which GCR communicates with UAVs.

Sr. No.	Bandwidth	Description
1	2.4 KHz to 2483.500 MHz	The appropriate standard is EN 300 328, which is digital wideband data transmission equipment, and sometimes, the standard used is EN 300 440, which is general short-range devices. <b>Purpose:</b> Mostly used for short-range surveillance or short-range maneuvering missions.
2	5.47 KHz to 57250 MHz	Operational power is less or equal to 1 watt, whereas the power spectral density is less the 50mW/1 MHz frequency. The standard is EN 201 893, which is known as RLAN equipment. <b>Purpose:</b> Long stay in sky operations, used mainly for aerial photography.
3	5.725 KHz to 5875 MHz	Its operational power rating is less than 25 mW and standard is EN 300 440, which is general short-range devices. <b>Purpose:</b> Used for short-range surveillance with fast maneuvering and manipulating tasks.
4	5030 to 5091 MHz	This is the frequency used only for the International Telecommunication Union (ITU) and, therefore, cannot be used for communication with drones. <b>Purpose:</b> Used in such operations where data sharing is important with ground control room (GCR).

### 3. Classification of UAVs

One must understand the real sense of calling any drone a UAV. Not all drones can be classified as UAVs. A UAV can be controlled autonomously without a pilot and can be controlled remotely [17].

#### 3.1. Classification of Drones

Drone is a very generic term and can refer to intelligent or autonomous vehicles such that there are unmanned aerial vehicles of different types. This can be hexarotors, quadrotors, multirotors or wing-based air vehicles. Mainly discussing the flying drones, they can be classified into three main categories as follows:

- Rotary-wing drones;
- Fixed-wing drones;
- Hybrid-wing drones;
- Flapping-wing drones.

The drones with a vertical take-off and landing (VTOL) feature and that can hover at a high rate are known as rotary-wing drones. The most common example is a quadrotor unmanned aerial vehicle that has four brushless DC motors. Drones with the capability to fly aggressively and glide even with heavy payloads are known as fixed-wing drones. They perform a horizontal take-off and landing (HTOL). Lastly, the drones that have both fixed and rotary wings are known as hybrid-wing drones. They are designed to have both features of rotary- and fixed-wing drones so that they can perform HTOL and VTOL along with high-rate hovering. Some of the robots are designed to exhibit the flying motion of a fly [18], and here they proposed a 5% more power-efficient wing by changing the wing stiffer pattern parametrically. An experimental aerodynamic analysis found that this could relate to increased wing stiffness, as well as indications of vortex generation during the flap cycle. The experiments reported an improved generated lift, allowing the DelFly to be outfitted with a yaw-rate gyro, pressure sensor, and microprocessor. These flapping-wings were later scaled to the micro level and are known as flapping-wing micro air vehicles (FWMAVs), due to inspiration taken from microscopic insects. These FWMAVs have the ability to perform activities in urban and interior environments. However, there are many hurdles for the successful flight of these vehicles that are replicating insect flight, including their design, manufacture, control, and propulsion [19,20].

### 3.2. Classification of UAVs Based on Ground Command and Control

It is already shown in Figure 2 that any UAV can be controlled remotely using a ground command and control mechanism, either by mobile phone, radio channel frequency, or the internet of things [21]. Therefore, these UAVs are classified based on their ability to fly over long distances without any intervention. These types are mentioned below as:

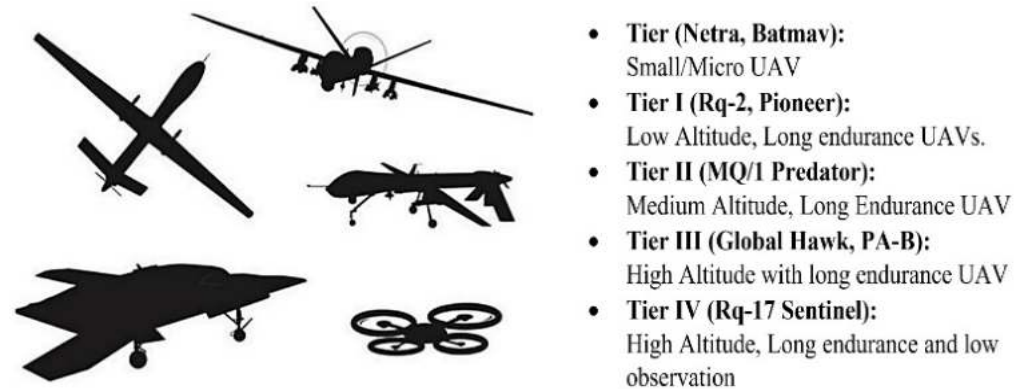
- Fully autonomous controlled UAVs: These are the UAVs that can perform different tasks without any intervention from human beings and are fully automated.
- Remotely operated UAVs: These UAVs are designed to execute the task as directed by a human being. Thus, they have a human as their main operator.
- Remotely pilot-controlled UAVs: Drones where all tasks and maneuvers are performed by the human-based remote control from the GCR.

The above-referred classification is summarized in Table 2 [22] along with the pros and cons of the UAVs.

**Table 2.** Classification of UAVs based on wing type and altitude.

Factors	Based on Wing Type			Based on Altitude	
	Fixed-Wing Type	Rotary-Wing Type	Hybrid-Wing Type	Low Altitude Below 400 ft	High Altitude Above 400 ft
Hovering	No	Yes	Yes	Yes	Yes
Small Size	No	Yes	Yes	Yes	No
Transport goods	Yes	Low weight	Yes	Yes	No
Battery time (in hour)	>1 h	1 h	>1 h	>1 h	>1 h
Maneuver speed	High Speed	Low speed	High speed	High Speed	High Speed
Flexible deployment of communication	No	No	No	Yes	No
Cost effective	Expensive	Cheap	Expensive	Cheap	Cheap
Endurance	High	Low	Medium	Low	High

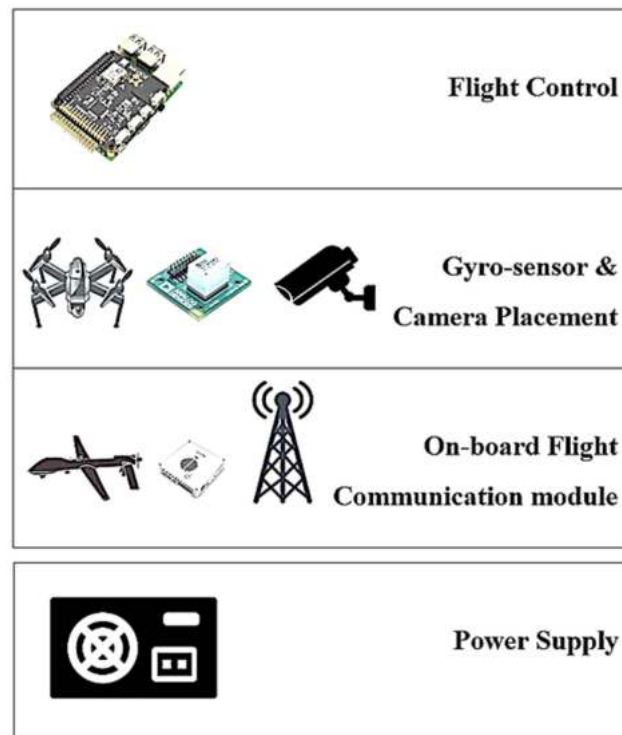
In the above, the reader might be confused with the term of flexible deployment of communication. This states the proper coverage at which the drone can be controlled and stabilized for any sort of task. There are some research contributions that have classified the drones based on their altitude, as demonstrated in Figure 3 along with their examples [23,24].



**Figure 3.** Classification of UAVs in terms of altitude.

#### 4. Communication Methods and Architecture

Today, one may see the variety of several drones being opted for commercial and domestic use. This is because they are cost effective and are controlled remotely from anywhere. In military operations, mostly the micro or miniature-type UAVs are used, but there are some major limitations in terms of size and weight. In Figure 3, tier II and III UAVs have several requirements such as being able to deploy sensors, and having a global positioning system (GPS), communication module, and efficient batteries. This is illustrated further in Figure 4. Although there are huge advancements being noticed in the field of drones or unmanned aerial vehicles (UAVs), at the same time, there are some limitations associated with the software and hardware support.



**Figure 4.** Components of an unmanned aerial vehicle.

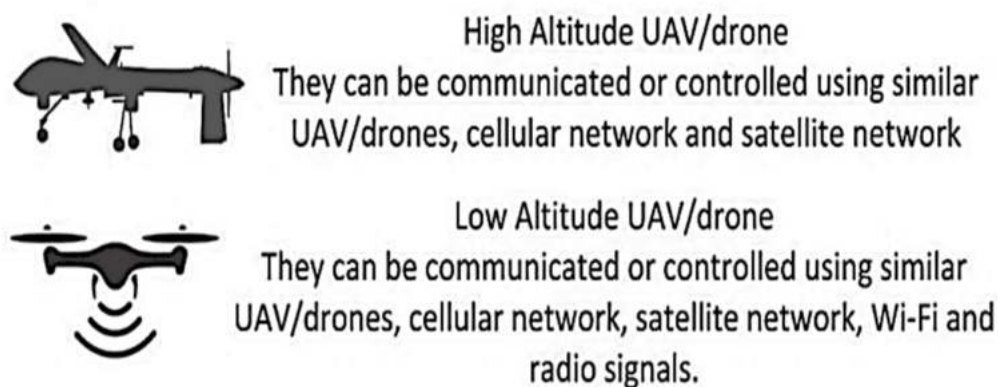
The UAVs which are proposed for military operations have some advanced sensors which are not accessible to ordinary people. These sensors enable drones to carry additional payloads. After studying different UAVs, one may subdivide UAVs into their five major components as follows:

- Drone airframe;
- Onboard controller;
- Payload capability;
- Communication system;
- Efficient batteries.

Discussing the airframe of a UAV, one must consider aspects such as aerodynamics, a lightweight structure, and stability. These can be some of the constraints to designing the UAV airframe. Moreover, the onboard controller is the main thing that maneuvers the drone. Therefore, it must be equipped with all essential sensors such as the accelerometer, gyro sensor, pressure sensor, GPS, and camera. While designing the drone, one should consider the factor of payload variation. In this way, the drone may carry some nominal amount of weight from one place to another [25]. Another important component is the communication system where the drone requires some communication equipment such as a sitcom, modem, or radio channel-based equipment. This will ensure the communication and control between a UAV and the ground control room (GCR). Lastly, the UAV must have a reliable power source that can help it to fly for a specific time to fulfil the task. Mostly, lithium batteries are considered as the main power source for these UAVs [26].

#### *Communication Methods*

When one discusses the communication aspect of UAVs, one is directed toward several integral subcomponents such as the communication protocols, the network type, and the UAV model itself. This means that with the change of communication method, one may induct the number of components and this will change the architecture of the system [27]. Many researchers have suggested several topologies and designs. This has been illustrated in Figure 4 along with the altitude range. One may opt for a different communication module and protocol as per the mission and the nature of task type [28]. Moreover, with the advent of 5G networks, several constraints such as data rate, latency, and coverage have been resolved. This advancement in communication technology not only improved these areas, but also helps to improvise the positioning and control of drones in several critical rescue and surveillance missions; for that purpose, people have used advanced flight controls and multiple sensors as well, along with camera placement, controlled and communicated in different ways as per their altitude as shown in Figure 5. These technologies are summarized later in Table 3 [29].



**Figure 5.** Communication methods for high- and low-altitude levels.

**Table 3.** Classification of UAVs based on communication channels.

Technique	Channel Width	Band	Bit Rate	Range	Latency	Mobility Support
Wi-Fi	20 MHz	2.4 GHz to 5.2 GHz	6–54 Mbps	100 m	10 ms	Low
GPS	2 MHz	1176 to 1576 MHz	50 bps	-	10 ms	Higher
UMTS	5 MHz	700 to 2600 MHz	2 Mbps	10 Km	20–70 ms	High
5G	2.16 GHz	57 to 64 GHz	Up to 4 Gbps	50 m	-	Ultra-High
LTE	20 MHz	700 to 2690 MHz	Up to 300 Mbps	30 Km	10 ms	Very High
LTE-A	Up to 100 MHz	450 MHz to 4.99 GHz	Up to 1 Gbps		-	Very High

Due to high security concerns, the modern UAVs are controlled using the line-of-sight method at a low altitude, whereas for high altitudes, researchers have given preference to GPS and the beyond-line-of-sight (BLoS) technique. Table 4 describes these techniques in brief [30].

**Table 4.** Communication based on satellite type.

Type of Communication	Elevation in Km	Number of Satellites	Satellite Life	Handoff Frequency	Doppler	Gateway Cost	Propagation Path Loss
Geostationary Earth orbit (GEO)	Up to 36,000	3, no polar coverage	15+	NA	Low	Very expensive	Highest
Medium Earth orbit (MEO)	5000–15,000	8–20 global	10–15	Low	Medium	Expensive	High
Low Earth orbit (LEO)	500–1500	40–800 global	3–7	High	High	Cheap	Least

Tables 3 and 4 are very important for the readers to understand the significance of the several channels based on different bandwidths and satellites, respectively. Discussing Table 3, it communicates the different wireless communication methods, but at the same time, it shares that the communication will have a latency rate as well. The table also shares the channel width, band interval, and most importantly, the mobility support for the readers to design their drones accordingly.

Discussing Table 4, it communicates the type of communication based on satellite type. This will help the reader to see the elevation in kilometers, number of satellites, satellite life, handoff frequency doppler gateway cost, and most importantly, propagation path loss of each satellite communication [31].

## 5. Utilization of UAVs in Different Domains

The potential of UAVs and drones has been proved already, and this domain covers every type of utilization from personal usage to military purpose, as illustrated in Figure 6.

These UAVs can be more efficient in performing several missions if they are equipped with a camera, smart sensors, and processors. With these essential components, one may see 100 plus applications of drones mentioned by several researchers, such as in [32].

Please note that Figure 6 shows the areas where UAVs are utilized mostly in general, whereas Figure 7 shows the benevolent usage where UAVs are commonly used. The term malevolent usage shows the areas and specific domains where people have witnessed an incremental increase in utilizing the drones over the last few decades. Factors such as diligence, cost, mobility in the areas where humans are unable to reach, payload options, and risk compel everyone to use drones/UAVs. Now, depending on the type of drone, they may be used in a better way. Commonly, it is seen that the design of drones is dependent on the type of mission they perform in the field [33]. Thus, categorizing them all with respect to their domains may lead to understanding their architecture in a better way. This has been illustrated in Figure 7, where every domain has its own privacy and security needs [33,34].



Figure 6. Utilization of drones in several domains.

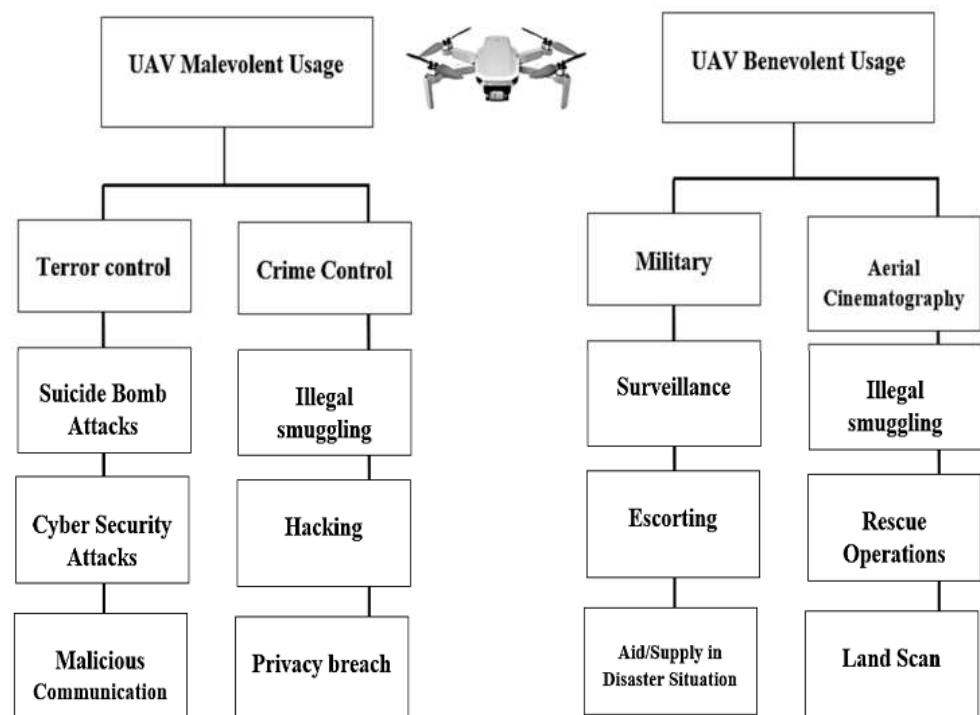


Figure 7. Malevolent and benevolent usages of a drone.

## 6. Security Threats Related to UAVs

UAVs offer several perks as the technology advances, but still, there are some constraints associated with privacy, security, and safety concerns [35,36]. Regularization and some important measures to license the utilization of drones is a very significant aspect. This limits unnecessary aerial photography. Most of the authorities in the world ensure this aspect and provide strict policies over uninformed aerial photography. If one discusses the network security point and the risk analysis, it is an admitted fact that the coverage is quite different as compared to any sort of wireless sensor network (WSN) or any mobile ad hoc networks (MANETs) [37]. The reason is because of the resource constraints, as UAV-related coverage is broader and wider than WSNs or MANETs.

The framework that sets the rules to operate drones in any vicinity is known as authentication, authorization, and accounting (AAA), which states several privileges to the controller of a drone to operate as per the mentioned administrative rights, whereas it also shares some of the rigid authentication procedures for drones to protect the control of a drone so that it may not be diverted to any other unknown entity. Moreover, in case of any uncertainty or illegal activity by drone, one may easily track down the operator. This is done to limit illegal surveillance, cyberattacks, and privacy threats. Thus, several mechatronic engineering solutions have been presented to overcome these malicious activities [38].

These drones are low cost and easily available in markets nowadays, and therefore, they are easy to use for any sort of criminal activity. Their ability to carry a wide range of external payloads make them more dangerous as it could lead to drones carrying any harmful chemical or explosive thing. Moreover, their ability to reach places where normal human beings cannot makes them more lethal because they can deliver anything without coming under anyone's notice [39]. It should be noted that security is not the only concern, but one may also see a safety concern if drones are flying over any populous place and, due to any number of faults, may crash, which can lead to several types of tragedies [40]. These sorts of incidents have been reported often. One of the examples is when a UAV faced a collision with British Airways BA727, which was a passenger aircraft in April 2016. After looking over these incidents and issues, one may ensure below the mentioned public safety measures:

- It is a high probability that a drone can be hacked or may deviate from its path due to heavy wind disturbance. Thus, there should be a reset option available which may turn the drone to a hovering condition only and help to gain the control back.
- There are certain areas where drones may face signal jammers and, later, can be controlled for a cyberattack. Thus, drones must have some sort of filter that may detect if there is any signal jammer nearby.

The third safety measure is related to its design, as most drones have open propellers as shown in Figure 8. In case of uncertainty, these propellers may go off and may harm anyone nearby; thus, the safety design as shown in Figure 9 is necessary to avoid any harm during a crash.



**Figure 8.** UAVs with open propellers.



**Figure 9.** UAVs with closed and safety propellers.

Lastly, there are some serious privacy concerns as well. Since UAVs on the market can easily be procured with high-definition cameras, this may lead to the recording of any

private property without permission. Due to this reason, Canadian Public Safety (CPS) stated that these UAVs are prohibited from flying over any property without mutually agreed permission [41].

## 7. Current Vulnerability Issues of UAVs

For these UAVs, unfortunately, there is neither a standardization of policies nor the availability of wireless security [42–44]. This leads to several threats, as highlighted in Table 5. There are researchers who have addressed different types of cyber-attacks associated with the several types of UAVs in a pre-controlled environment [45–50]. Such practical validations include the crashing of drones with many parallel requests and modifying the request packet known as the buffer-overflow attack, whereas some researchers went for the cache-poisoning approach that leads to the shutdown of communication between the drone and GCR. In all conditions, most attacks occur to target the operating system or, in other words, the microcontroller of the drone [51]. Since there are huge advancements in the technology, UAVs have a high probability of experiencing such attacks, as shown in Figure 9 [52–58]. From these attacks, the most common attack is GPS spoofing, such as signal jamming, de-authentication. and zero-day attacks.

**Table 5.** Summary of all current vulnerability issues in UAVs.

Vulnerability Type	Description
Malware issue	In various cases, it has been observed that these UAVs are generally connected and controlled via cell phone or any sort of remote control. These techniques are, thus far, not safe [43] and, therefore, the UAVs are easy to be hacked using a reverse-shell TCP payload that can be injected into UAV memory. Furthermore, this leads to installation of malware over UAVs automatically.
Spoofing	These are the issues related to the communication method, usually with serial port connections that are not encrypted properly [44]. Due to this spoofing issue, the information associated with GPS can be taken and altered.
Manipulation and other common concerns	The flying paths which UAVs must track are pre-programmed before; therefore, these paths can be altered [45], whereas the common issues are related to wind, overheating, or any predator bird harming the lightweight drone easily [46].
Physical design and control system constraints	There are various challenges with unmanned aerial vehicle control system design, such as the sluggish convergence rate, which prevents the drone from performing fast or aggressive maneuvers, and one may notice faults in the flight or divergence from the target trajectory [47,59,60]. This slow convergence rate and glitches are caused by the physical architecture of drones or the planned control system, which is primarily intended to stabilize the drone in uncertain conditions.
Sensorization issue	Since these UAVs depend on sensors, thus, it is also proved that the ultrasonic waves may attack the MEMS gyro sensors [47].
Wi-Fi constraints	Operating drones using a Wi-Fi facility may be risky. This is proved in [48] where the connection was disrupted with the help of software and changing the control of the UAV.
GPS issue	Automatic Dependent Surveillance–Broadcast depends on the GPS module, which is not encrypted sometimes and may lead to spoofing [49].
Firmware issue	The bugs available in the first prototype and first algorithm which come to the front after usage [50].
Sky Jack-based attacks	Sky Jack is one software used to conduct the attacks related to de-authentication of targets during control [51].
Controller issues	These issues are related to the operation control unit and may puzzle the controller by changing the live feed to some other video [52].

## 8. Current State-of-the-Art Solutions

The very first and significant thing to resolve the threats is to identify them first. Thus, Table 5 and Figure 10 classify these attacks. Moreover, there are several contributions which address these sorts of attacks along with the suitable measures [58]. In recent times, researchers have utilized machine learning approaches to demonstrate the intrusion detection system (IDS). Thus, machine learning (ML)-based IDS is one of the areas where researchers are still working to improve the results [59,60]. Blockchain is also among the most effective approaches for UAV/drone security and privacy [12]. This ML-based IDS technique is from the robust technique, and it is categorized into three kinds as mentioned below:

- Rule-based IDS;
- Signature-based IDS;
- Anomaly-based IDS.

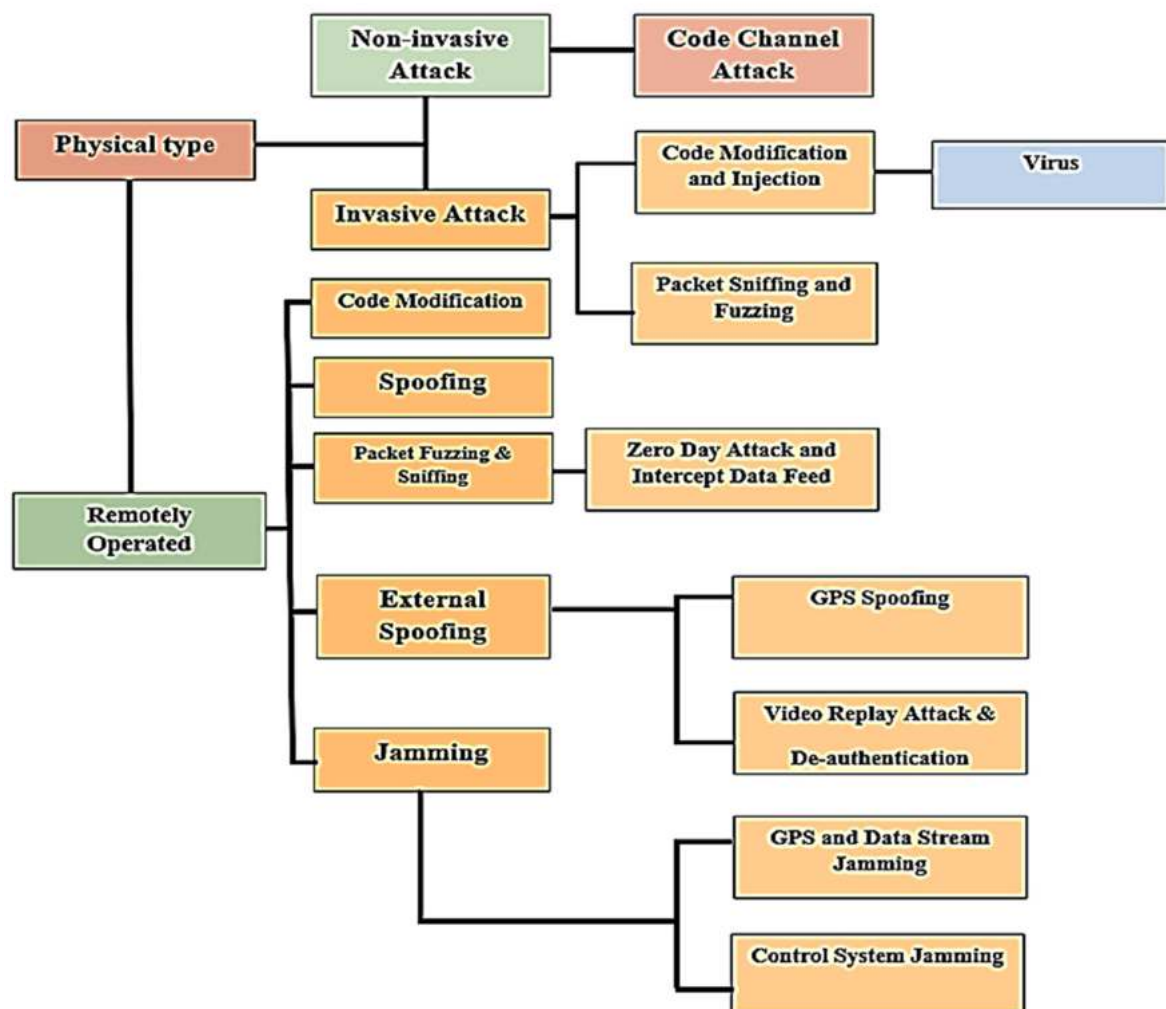


Figure 10. UAV attack vector with reported incidents.

Above are the major approaches for detecting threats that intrusion detection systems utilize to inform the operator in the ground control room (GCR). Rule-based threat detection is a new approach enabled by artificial intelligence (AI) [61]. In comparison to others, it is more reliant on technology and less on manual interaction. Signature-based detection works well for recognizing known threats. It uses a pre-programmed list of known threats and their indicators of compromise to operate (IOCs). An indicator of compromise (IOC) could be a distinctive behavior that typically precedes a malicious network attack, such

as file hashes, malicious sites, known byte sequences, or even the content of email subject headings. A signature-based IDS examines network packets and compares them to a database of known IOCs or attack signatures to detect any suspicious behavior. Anomaly-based intrusion detection systems, on the other hand, can alert you to unusual behavior. An anomaly-based detection system uses machine learning to educate the detection system to recognize a normalized baseline rather than searching for known threats. All network activity is compared to the baseline, which represents how the system ordinarily performs. Rather than looking for known IOCs, anomaly-based IDS simply detects any unusual behavior to generate alarms.

To identify the false data injection attacks, one may use the rule-based approach. This is used to target the signal strength in between the UAV and ground control room (GCR) and can be useful for any sort of known attack, pattern, or technique only. Some research papers have proposed a signature-based IDS over drones [62] where they addressed bio-inspired cyberattacks associated with air-born networks. Last, but certainly not least, is the anomaly-based IDS scheme which is used against jamming attacks [62]. The only limitation of anomaly IDS is the huge resource requirement.

Similarly, there are some researchers who have suggested some algorithm schemes with forensic methods to address the advanced and complex attacks. They are complex and difficult to identify [63–69]. With the help of forensics, both perpetrator and method of attack can be identified. With the identification of the attack type, appropriate countermeasures can be implemented to avoid any future incident. As per the survey of [70], between 2014 and 2017 incidents among airplanes and drones amplified from 6 to 93, which makes it very important for the authorities to address security and privacy issues for UAVs. Due to an increase in cyberattacks on drones/UAVs, the government needs to introduce strict policies and standards to minimize these concerns. With the popularity of UAVs among the civilian population, attacks and the illegal use of UAVs will likely proliferate. Civilian or domestic UAV countermeasures are divided into physical and local countermeasures, which are already proposed but still can be improved.

There are several survey papers that address the latest integration of UAVs into cellular networks and discuss the inference issues [71], as well as those, like this paper, that address the significant concerns related to the standardization and regulation of drones and their privacy. In addition to this, the manuscript focuses on the issues related to addressing these limitations while communicating from the drone to the ground control room (GCR). Some of the researchers proposed survey papers also on the quantity and quality of service requirements and discussed network-relevant mission parameters [72], which is unlike this paper that discusses the safety, privacy, and adaptability features of drones.

## 9. Open Research Areas and Recommendations

After studying the previous sections, it is noted that there is still a need to improve some of the areas associated with unmanned aerial vehicles (UAVs). These areas are very significant and one may address these concerns to enhance the utilization of drones [73,74]. One of the important areas is path loss, where one needs to propose the channel model to hold on to the communication at higher carrier frequencies, and even in the presence of tall concrete buildings. This area is in regard to the latency in the communication, which should be less than 1 millisecond and remains as an area of concern [75,76].

In addition to these areas, one may work over the reliability aspect, where one may improve the drones with ultra-reliable communication so that even with the increase in UAVs in the sky, the communication can be performed easily. It is noted that these drones have not been operated in the sky for a long time, which is because of the battery life. Hence, battery optimization for drones is also one of the areas where researchers may engage themselves to increase the flight time.

Last, but certainly not least, is the amalgamation of artificial intelligence and computer vision algorithms in a drone to improve the mobility of the drone without any collision. This will protect the drone in terms of data logging and security [77–84]. This manuscript

also suggests some of the recommendations that may improve the privacy and security aspects, such as the registration of drone licenses. This will ensure the authorities identify the specific drone that has created an inconvenience in the jurisdiction [85–91]. Moreover, there must be flying permits allotted after necessary training to limit the illegal utilization of drones in any activity.

Another recommendation is to educate the public about the legal and illegal usage of such autonomous unmanned aerial vehicles and the laws related to it so that if they witness anything around, they may easily report it. In any vicinity, there are some restricted zones; thus, the market drones must be operated based on a built-in map [92–95] as per the local regions. In this way, when any drone is forced to enter into any barred jurisdiction, it will automatically revert to the ground using the vertical take-off and landing (VTOL) mode. In terms of security tools, this paper proposes the machine learning-based IDS system [96–98] to improve the security infrastructure of UAVs, and lastly, there should be rigid multi-factor authentication methods that tackle the security threats easily.

There are several future recommendations to increase the standards for the security and privacy aspects of UAVs. These aspects are improved by proposing an approach which is based on a pairing certificate so that other strange entities may not easily connect or communicate with our UAV. Some of them are based on identification/authentication protocols [99–103]. To secure the drone more and make it less vulnerable, researchers have also used a symmetric searchable encryption method (SSE) [104] as well. Some researchers have proposed an internet of things (IoT) feature also for the same purpose [105–108]. In terms of identifying an unknown input observance, one may see an intelligent control algorithm that stabilizes the UAV in the presence of an unknown input [108–110] and devise it in trajectory and altitude levels to identify unknown system dynamics online by utilizing filtering manipulations that possess a concise structure, low calculation consumption, and asymptotic error convergence. In addition to this, the manuscript highlights the major domains and compares them with some of the latest review papers on UAVs for contrast. This is the significance of this article, which is seen in Table 6.

**Table 6.** Summary of all major domains along with reference list.

Area/Domain	[82]	[75,76]	[57]	[47]	[20,32]	[4]
Regulations and classification					•	
Communication methods				•	•	•
Applications	•	•		•		•
Security issues and solutions	•	•			•	
Physical and logical attacks					•	
Open research area				•	•	•
Recommendations			•	•	•	

One can observe from the above chart that the first column lists the issues that have already been covered in the paper, whereas the first row lists the number of review manuscripts that state or debate the same theme. After reading through Table 6, one can find this article to be more thorough in determining the future answer quickly and effectively. A black dot in the table above indicates the articles in which the topics were directly mentioned.

A statement in support of integrating UAVs with contemporary trends of communication and a control system is developed by evaluating various research contributions linked to UAVs and communication aspects to identify the limits [104,111]. To get over the limitations, more research is still needed to examine the subtopics below:

- There is a need to address the area of high-speed mobility, as there are huge chances to hack the communication links through the ground control room or with neighboring UAVs.

- In some of the integrated solutions, i.e., the space-air-ground network, one may see a frequent issue of synchronization, and thus, it is desirable to re-design some cooperation incentives for using cross-layered protocols with linked reliability. In this way, there will be less chances of any security attacks.
- One more aspect is to recommend a lightweight mechanism for UAVs to prevent attacks, such as eavesdropping, a man-in-the-middle attack [112,113], and so on. There are a number of artificial intelligence solutions which are recommended in [28] for addressing the security in cellular network-based controlled UAVs for delivering packages.
- Integrating UAVs with the IoT can result in endurance and reliability, but at the same time, it consumes the maximum battery capacity which is generally small; thus, this may lead UAVs toward possible collisions and can be a high-risk threat.
- Lastly, proposing a big data deep reinforcement learning approach to enable the dynamic arrangement of networking, caching, and computing resources for improving the performance of UAVs with secure operations in smart cities.

Thus, with all the above recommendations and in contrast to these topics, our manuscript provides a detailed direction for future work.

## 10. Conclusions

The use of UAVs has increased dramatically, ushering in an era of autonomous systems and vehicles. These drones are quite important because they have many benefits for both civil and military concerns. However, with this rise in usage, severe privacy and security concerns are also evident. The most frequent reason why these drones are chosen in any sneaky assault is because they are readily available and inexpensive to obtain.

There have been numerous scientific contributions that have already addressed the countermeasures to these worries; however, there are still some areas that have not been addressed and can, therefore, still be exploited for any negative purposes, such as privacy and security issues. In this current, technological age, these two challenges cannot be disregarded. As a result, this review paper offers a thorough examination of these two pressing issues by providing a quick summary of the causes of each worry, as well as potential solutions. The existing solutions and a number of recommendations are presented in this study, which claims that the UAV drones can be enhanced if proper data integration, authentication, and accessibility factors are treated seriously.

**Author Contributions:** Conceptualization, G.E.M.A., A.L., R.J.M., V.S.A. and S.A.B.M.Z.; methodology, A.L.; software, G.E.M.A. and A.L.; validation, V.S.A., R.J.M. and S.A.B.M.Z.; formal analysis, A.L.; investigation, V.S.A. and A.L.; resources, S.A.B.M.Z. and G.E.M.A.; data curation, S.A.B.M.Z., V.S.A. and A.L.; writing—original draft preparation, G.E.M.A.; writing—review and editing, G.E.M.A. and S.A.B.M.Z.; visualization, V.S.A. and S.A.B.M.Z.; supervision, A.L., S.A.B.M.Z., R.J.M. and V.S.A.; project administration, G.E.M.A.; funding acquisition, S.A.B.M.Z. and G.E.M.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** The article processing charges were financed by the Research Management Center, Universiti Teknologi PETRONAS, Malaysia, under the Research Fund, which was supported by Yayasan Universiti Teknologi Petronas (YUTP), award number 015LC0-316.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to express their gratitude to the Centre of Graduate Studies (CGS), Universiti Teknologi, PETRONAS, Malaysia, and the Erasmus+ Program for giving them the opportunity to study abroad as exchange students at Telecom SudParis, France and conduct this research using the cutting-edge resources of that country's university. Finally, I would like to express my gratitude to the Department of Electrical and Electronic Engineering for offering me a position to pursue PhD studies through YUTP Funding, grant number 015CL0-316.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bombe, M.K. Unmanned Aerial Vehicle (UAV) Market Worth \$21.8 billion by 2027- Pre and Post COVID-19 Market Analysis Report by Meticulous Research. 11 June 2020. Available online: [https://www.meticulousresearch.com/download-sample-report/cp\\_id=5086](https://www.meticulousresearch.com/download-sample-report/cp_id=5086) (accessed on 18 August 2022).
2. Kumar, R.; Kumar, P.; Tripathi, R.; Gupta, G.P.; Gadekallu, T.R.; Srivastava, G. SP2F: A secured privacy-preserving framework for smart agricultural Unmanned Aerial Vehicles. *Comput. Networks* **2021**, *187*, 107819. [CrossRef]
3. Israr, A.; Abro, G.E.M.; Sadiq Ali Khan, M.; Farhan, M.; Zulkifli, B.M.; ul Azrin, S. Internet of things (IoT)-Enabled unmanned aerial vehicles for the inspection of construction sites: A vision and future directions. *Math. Problems Eng.* **2021**. [CrossRef]
4. Chen, R.; Yang, B.; Zhang, W. Distributed and Collaborative Localization for Swarming UAVs. *IEEE Internet Things J.* **2020**, *8*, 5062–5074. [CrossRef]
5. Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint. *IEEE Commun. Surv. Tutorials* **2016**, *18*, 2624–2661. [CrossRef]
6. Chan, K.W.; Nirmal, U.; Cheaw, W.G. Progress on drone technology and their applications: A comprehensive review. *AIP Conf. Proc.* **2018**, *2030*, 020308. [CrossRef]
7. Hartmann, K.S.C. The vulnerability of UAVs to cyber, in Cyber Conflict (CyCon). In Proceedings of the 2013 5th International Conference, Tallinn, Estonia, 4–7 June 2013.
8. Bowden, M. How the Predator Drone Changed the Character of War, Smithsonian Magazine. November 2013. Available online: <https://www.smithsonianmag.com/history/how-the-predatordrone-changed-the-character-of-war-3794671/> (accessed on 18 August 2022).
9. Ekramul, D. First Successful Air-Raid in History. 22 August 2019. Available online: <https://www.daily-bangladesh.com/english/First-successful-Air-raid-in-history/27424> (accessed on 18 August 2022).
10. O'Donnell, S. Consortiq. Available online: <https://consortiq.com/short-history-unmanned-aerialvehicles-uavs/> (accessed on 18 August 2022).
11. Berg, T.R. Air Space Mag. 10 January 2020. Available online: <https://www.airspacemag.com/daily-planet/first-map-compiled-aerial-photographs-180973929/> (accessed on 18 August 2022).
12. Abdullah, Q.A. Introduction to the Unmanned Aircraft Systems. Available online: <https://www.eeducation.psu.edu/geog892/node/643> (accessed on 18 August 2022).
13. Miah, A. *Drones: The Brilliant, the Bad and the Beautiful*; Emerald Group Publishing: Bently, UK, 2020. [CrossRef]
14. Ali, B.S.; Saji, S.; Su, M.T. An assessment of frameworks for heterogeneous aircraft operations in low-altitude airspace. *Int. J. Crit. Infrastruct. Prot.* **2022**, *37*, 100528. [CrossRef]
15. Wright, S. Ethical and safety implications of the growing use of civilian drone. UK Parliam. *Website Sci. Technol. Commit.* **2019**.
16. Coach, U. Master List of Drone Laws (Organized by State & Country). Available online: <https://uavcoach.com/drone-laws/> (accessed on 18 August 2022).
17. Aljehani, M.; Inoue, M.; Watanbe, A.; Yokemura, T.; Ogyu, F.; Iida, H. UAV communication system integrated into network traversal with mobility. *SN Appl. Sci.* **2020**, *2*, 2749. [CrossRef]
18. de Croon, G.C.H.E.; Groen, M.A.; De Wagter, C.; Remes, B.; Ruijsink, R.; van Oudheusden, B.W. Design, aerodynamics and autonomy of the DelFly. *Bioinspir. Biomim.* **2012**, *7*, 025003. [CrossRef]
19. Cheaw, B.H.; Ho, H.W.; Abu Bakar, E. Wing Design, Fabrication, and Analysis for an X-Wing Flapping-Wing Micro Air Vehicle. *Drones* **2019**, *3*, 65. [CrossRef]
20. Teoh, Z.E.; Fuller, S.B.; Chirarattananon, P.; Prez-Arancibia, N.O.; Greenberg, J.D.; Wood, R.J. A hovering flapping-wing microrobot with altitude control and passive upright stability. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 3209–3216. [CrossRef]
21. Professionals, Drones and Remotely Piloted Aircraft (UAS/RPAS)-Frequencies and Radio Licenses, Traficom. 17 July 2021. Available online: <https://www.traficom.fi/en/transport/aviation/drones-and-remotely-piloted-aircraft-uasrpasfrequencies-and-radio-licences> (accessed on 18 August 2022).
22. Carnahan, ISO/TC 20/SC 16 Unmanned Aircraft Systems. 2014. Available online: <https://www.iso.org/committee/5336224.html> (accessed on 18 August 2022).
23. Irizarry, M.J.; Gheisari, B. Walker, Usability Assessment of Drone Technology as Safety Inspection Tools. *Electron. J. Inf. Technol. Constr.* **2012**, *17*, 194–212.
24. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.-H.; Debbah, M. A tutorial on UAVs for wireless networks: Applications, challenges, and open problems. *IEEE Commun. Surv. Tut.* **2019**, *21*, 2334–2360. [CrossRef]
25. Lowbridge, C. Are Drones Dangerous or Harmless Fun? *BBC News*. 5 October 2015. Available online: <https://www.bbc.com/news/uk-england-34269585> (accessed on 18 August 2022).
26. Federal Aviation Authorities, Recreational Flyers & Modeler Community-Based Organizations. 18 February 2020. Available online: [https://www.faa.gov/uas/recreational\\_fliers/](https://www.faa.gov/uas/recreational_fliers/) (accessed on 18 August 2022).
27. Pilot, What's the Difference Between Drones, UAV, and UAS? Definitions and Terms. Pilot Institute. 22 March 2020. Available online: <https://pilotinstitute.com/drones-vs-uav-vs-uas/> (accessed on 18 August 2022).

28. Fotouhi, A.; Qiang, H.; Ding, M.; Hassan, M.; Giordano, L.G.; Garcia-Rodriguez, A.; Yuan, J. Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3417–3442. [CrossRef]
29. Nagpal, K. Unmanned Aerial Vehicles (UAV) Market, Q Tech Synergy. 24 December 2016. Available online: <https://defproac.com/?p=2041> (accessed on 18 August 2022).
30. Pastor, E.; Lopez, J.; Royo, P. A Hardware/Software Architecture for UAV Payload and Mission Control. In Proceedings of the 2006 IEEE/AIAA 25TH Digital Avionics Systems Conference, Portland, OR, USA, 15–18 October 2006; pp. 1–8. [CrossRef]
31. VanZwol, J. Design Essentials: For UAVs and Drones, Batteries are Included, Machine Design. 4 April 2017. Available online: <https://www.machinedesign.com/mechanical-motionsystems/article/21835356/design-essentials-for-uavs-and-drones-batteries-are-included> (accessed on 18 August 2022).
32. Sharma, A.; Vanjani, P.; Paliwal, N.; Basnayaka, C.M.; Jayakody, D.N.K.; Wang, H.-C.; Muthuchidambaramanathan, P. Communication and networking technologies for UAVs: A survey. *J. Netw. Comput. Appl.* **2020**, *168*, 102739. [CrossRef]
33. Ullah, H.; Nair, N.G.; Moore, A.; Nugent, C.; Muschamp, P.; Cuevas, M. 5G Communication: An Overview of Vehicle-to-Everything, Drones, and Healthcare Use-Cases. *IEEE Access* **2019**, *7*, 37251–37268. [CrossRef]
34. Luo, C.; Miao, W.; Ullah, H.; McClean, S.; Parr, G.; Min, G. Unmanned aerial vehicles for disaster management. In *Geological Disaster Monitoring Based on Sensor Networks*; Springer: Singapore, 2019; pp. 83–107.
35. Hosseini, N.; Jamal, H.; Haque, J.; Magesacher, T.; Matolak, D.W. UAV Command and Control, Navigation and Surveillance: A Review of Potential 5G and Satellite Systems. In Proceedings of the 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019; pp. 1–10. [CrossRef]
36. IvyPanda, Unmanned Aerial Vehicles Essay. 13 January 2020. Available online: <https://ivypanda.com/essays/unmanned-aerial-vehicles-essay/> (accessed on 20 December 2020).
37. Valavanis, K.P.; Vachtsevanos, G.J. UAV Applications: Introduction. In *Handbook of Unmanned Aerial Vehicles*; Springer: Dordrecht, The Netherlands, 2015; pp. 2639–2641. [CrossRef]
38. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]
39. Cook, K.L.B. The Silent Force Multiplier: The History and Role of UAVs in Warfare. In Proceedings of the 2007 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2007; pp. 1–7. [CrossRef]
40. Siddiqi, M.A.; Khoso, A.M. Aziz, Analysis on Security Methods of Wireless Sensor Network (WSN). In Proceedings of the SJCMS 2018, Sukkur, Pakistan, 10 December 2018.
41. Cavoukian, A. *Privacy and Drones: Unmanned Aerial Vehicle*; Information and Privacy Commissioner: Toronto, ON, Canada, 2012.
42. Kafi, M.A.; Challal, Y.; Djenouri, D.; Doudou, M.; Bouabdallah, A.; Badache, N. A Study of Wireless Sensor Networks for Urban Traffic Monitoring: Applications and Architectures. *Procedia Comput. Sci.* **2013**, *19*, 617–626. [CrossRef]
43. Mansfield, K.; Eveleigh, T.; Holzer, T.H.; Sarkani, S. Unmanned aerial vehicle smart device ground control station cyber security threat model. In Proceedings of the 2013 IEEE International Conference Technology Homeland Security (HST), Waltham, MA, USA, 12–14 November 2013; pp. 722–728. [CrossRef]
44. Smith, K.W. Drone Technology: Benefits, Risks, and Legal Considerations. *Seattle J. Environ. Law (SJEL)* **2015**, *5*, 291–302.
45. Eyerman, J.; Hinkle, K.; Letterman, C.; Schanzer, D.; Pitts, W.; Ladd, K. *Unmanned Aircraft and the Human Element: Public Perceptions and First Responder Concerns*; Institute of Homeland Security and Solutions: Washington, DC, USA, 2013.
46. Syed, N.; Berry, M. *Journo-Drones: A Flight over the Legal Landscape*; American Bar Association: Chicago, IL, USA, 2014.
47. Rahman, M.F.B.A. *Smart CCTVS for Secure Cities: Potentials and Challenges*; Rajaratnam School of International Studies (RSIS): Singapore, 2017.
48. Kim, A.; Wampler, B.; Goppert, J.; Hwang, I.; Aldridge, H. Cyber Attack Vulnerabilities Analysis for Unmanned Aerial Vehicles. *Aerospace Res. Cent.* **2012**, 2438. [CrossRef]
49. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [CrossRef]
50. Soria, P.R.; Bevec, R.; Arrue, B.C.; Ude, A.; Ollero, A. Extracting Objects for Aerial Manipulation on UAVs Using Low Cost Stereo Sensors. *Sensors* **2016**, *16*, 700. [CrossRef]
51. Erdelj, M.; Natalizio, E. Drones, Smartphones and Sensors to Face Natural Disasters. In Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, Paris, France, 10–15 June 2018; pp. 75–86. [CrossRef]
52. Son, Y.; Shin, H.; Kim, D.; Park, Y.; Noh, J.; Choi, K. Rocking Drones with Intentional Sound Noise on Gyroscopic Sensors. In Proceedings of the 24th USENIX Security Symposium, Washington, DC, USA, 12–14 August 2015.
53. Zhi, Y.; Fu, Z.; Sun, X.; Yu, J. Security and Privacy Issues of UAV: A Survey. *Mob. Netw. Appl.* **2019**, *25*, 95–101. [CrossRef]
54. Strohmeier, M.; Schafer, M.; Lenders, V.; Martinovic, I. Realities and challenges of nextgen air traffic management: The case of ADS-B. *IEEE Commun. Mag.* **2014**, *52*, 111–118. [CrossRef]
55. Hooper, M.; Tian, Y.; Zhou, R.; Cao, B.; Lauf, A.P.; Watkins, L.; Robinson, W.H.; Alexis, W. Securing commercial WiFi-based UAVs from common security attacks. In Proceedings of the MILCOM 2016-2016 IEEE Military Communications Conference, Baltimore, MD, USA, 1–3 November 2016; pp. 1213–1218. [CrossRef]

56. Hartmann, K.; Giles, K. UAV exploitation: A new domain for cyber power. In Proceedings of the 2016 8th International Conference Cyber Conflict, Tallinn, Estonia, 31 May–3 June 2016; pp. 205–221. [CrossRef]
57. Rivera, E.; Baykov, R.; Gu, G. A Study on Unmanned Vehicles and Cyber Security. In Proceedings of the Rivera 2014 ASO, Austin, TX, USA, 2014.
58. Junejo, I.N.; Foroosh, H. GPS coordinates estimation and camera calibration from solar shadows. *Comput. Vis. Image Underst.* **2010**, *114*, 991–1003. [CrossRef]
59. Wang, W.; Huang, H.; Zhang, L.; Su, C. Secure and efficient mutual authentication protocol for smart grid under blockchain. *Peer-to-Peer Netw. Appl.* **2020**, *14*, 2681–2693. [CrossRef]
60. Zhang, L.; Zhang, Z.; Wang, W.; Jin, Z.; Su, Y.; Chen, H. Research on a Covert Communication Model Realized by Using Smart Contracts in Blockchain Environment. *IEEE Syst. J.* **2021**, *16*, 2822–2833. [CrossRef]
61. Currier, C.; Moltke, H. *Spies in the Sky*; The Intercept: New York, NY, USA, 2016.
62. Yağdereli, E.; Gemci, C.; Aktaş, A.Z. A study on cyber-security of autonomous and unmanned vehicles. *J. Déf. Model. Simulation: Appl. Methodol. Technol.* **2015**, *12*, 369–381. [CrossRef]
63. Lee, Y.S.; Dongseo University; Kang, Y.-J.; Lee, S.-G.; Lee, H.; Ryu, Y. An Overview of Unmanned Aerial Vehicle: Cyber Security Perspective. *IT Converg. Technol.* **2016**, *4*, 30. [CrossRef]
64. Wu, L.; Cao, X.; Foroosh, H. Camera calibration and geo-location estimation from two shadow trajectories. *Comput. Vis. Image Underst.* **2010**, *114*, 915–927. [CrossRef]
65. Krishna, C.G.L.; Murphy, R.R. A review on cybersecurity vulnerabilities for unmanned aerial vehicles. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 194–199. [CrossRef]
66. Siddiqi, M.A.; Pak, W. Optimizing Filter-Based Feature Selection Method Flow for Intrusion Detection System. *Electronics* **2020**, *9*, 2114. [CrossRef]
67. Strohmeier, M.; Lenders, V.; Martinovic, I. Intrusion Detection for Airborne Communication Using PHY-Layer Information. In *Proceedings of the International Conference Detection of Intrusions and Malware, and Vulnerability Assessment*; Springer: Cham, Switzerland, 2015; pp. 67–77. [CrossRef]
68. Gil Casals, S.; Owezarski, P.; Descargues, G. Generic and autonomous system for airborne networks cyber-threat detection. In Proceedings of the 2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC), IEEE, New York, NY, USA, 5–10 October 2013; pp. 4A4-1–4A4-14. [CrossRef]
69. Rani, C.; Modares, H.; Sriram, R.; Mikulski, D.; Lewis, F.L. Security of unmanned aerial vehicle systems against cyber-physical attacks. *J. Déf. Model. Simul. Appl. Methodol. Technol.* **2015**, *13*, 331–342. [CrossRef]
70. Zhang, G.; Wu, Q.; Cui, M.; Zhang, R. Securing UAV Communications via Joint Trajectory and Power Control. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 1376–1389. [CrossRef]
71. Shao, X.; Wang, L.; Li, J.; Liu, J. High-order ESO based output feedback dynamic surface control for quadrotors under position constraints and uncertainties. *Aerosp. Sci. Technol.* **2019**, *89*, 288–298. [CrossRef]
72. Li, B.; Fei, Z.; Zhang, Y. UAV Communications for 5G and Beyond: Recent Advances and Future Trends. *IEEE Internet Things J.* **2018**, *6*, 2241–2263. [CrossRef]
73. Lee, Y.-S.; Kim, E.; Kim, Y.-S.; Seol, D.-C. Effective Message Authentication Method for Performing a Swarm Flight of Drones. *Emergency* **2015**, *3*, 95–97. [CrossRef]
74. Pilli, E.S.; Joshi, R.; Niyogi, R. A Generic Framework for Network Forensics. *Int. J. Comput. Appl.* **2010**, *1*, 251–408. [CrossRef]
75. Beebe, N.L.; Clark, J.G. A hierarchical, objectives-based framework for the digital investigations process. *Digit. Investig.* **2005**, *2*, 147–167. [CrossRef]
76. Jain, U.; Rogers, M.; Matson, E.T. Drone forensic framework: Sensor and data identification and verification. In Proceedings of the 2017 IEEE Sensors Application Symposium (SAS), Glasgow, UK, 31 October 2017; pp. 1–6. [CrossRef]
77. Roder, K.; Choo, N.K.R.A. Le-Khac, Unmanned Aerial Vehicle Forensic Investigation Process: Dji Phantom 3 Drone As A Case Study. *arXiv Prepr.* **2018**, arXiv:1804.08649.
78. Siddiqi, M.A.; Ghani, N. Critical Analysis on Advanced Persistent Threats. *Int. J. Comput. Appl.* **2016**, *141*, 46–50. [CrossRef]
79. Siddiqi, M.A.; Yu, H.; Joung, J. 5G Ultra-Reliable Low-Latency Communication Implementation Challenges and Operational Issues with IoT Devices. *Electronics* **2019**, *8*, 981. [CrossRef]
80. Khan, N.A.; Jhanjhi, N.Z.; Brohi, S.N.; Almazroi, A.A.; Almazroi, A.A. A secure communication protocol for unmanned aerial vehicles. *CMC-COMPUTERS MATERIALS CONTINUA* **2022**, *70*, 601–618. [CrossRef]
81. Wild, G.; Murray, J.; Baxter, G. Exploring Civil Drone Accidents and Incidents to Help Prevent Potential Air Disasters. *Aerospace* **2016**, *3*, 22. [CrossRef]
82. Goodrich, M. Drone Catcher: “Robotic Falcon” can Capture, Retrieve Renegade Drones, Michigan Tech. 7 January 2016. Available online: <https://www.mtu.edu/news/stories/2016/january/drone-catcher-robotic-falcon-can-capture-retrieve-renegade-drones.html> (accessed on 18 August 2022).
83. McNabb, M. DEDRONE Acquires the Anti Drone Shoulder Rifle, Batelle’s Drone Defender, Drone Life. 9 October 2019. Available online: <https://dronelife.com/2019/10/09/dedrone-acquires-theanti-drone-shoulder-rifle-batelles-drone-defender/> (accessed on 18 August 2022).

84. Capello, E.; Dentis, M.; Mascarello, L.N.; Primates, S. Regulation analysis and new concept for a cloud-based UAV supervision system in urban environment. In Proceedings of the Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), IEEE, (2017), Carnfield, UK, 25–27 November 2017; pp. 90–95. [CrossRef]
85. Joglekar, R. 4 Strategies for Stopping ‘Rogue’ Drones from Flying in Illegal Airspace. *ABC News*. 23 December 2018. Available online: <https://abcnews.go.com/Technology/strategies-stoppingrogue-drones-flying-illegal-airspace/story?id=59973853> (accessed on 18 August 2022).
86. Friedberg, S. A Primer on Jamming, Spoofing, and Electronic Interruption of a Drone, Dedrone. 19 April 2018. Available online: <https://blog.dedrone.com/en/primer-jamming-spoofing-andelectronic-interruption-of-a-drone> (accessed on 18 August 2022).
87. Cyber, T.E.O. How To Crack WPA/WPA2 Wi-Fi Passwords Using Aircrack-ng, Medium. 5 November 2019. Available online: <https://medium.com/@TheEyeOfCyberBuckeyeSecurity/howto-crack-wpa-wpa2-wi-fi-passwords-using-aircrack-ng-8cb7161abcf9> (accessed on 18 August 2022).
88. Yaacoub, J.-P.; Noura, H.; Salman, O.; Chehab, A. Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet Things* **2020**, *11*, 100218. [CrossRef]
89. Bonilla, C.A.T.; Parra, O.J.S.; Forero, J.H.D. Common Security Attacks on Drones. *Int. J. Appl. Eng. Res.* **2018**, *13*, 4982–4988.
90. Gaspar, J.; Ferreira, R.; Sebastião, P.; Souto, N. Capture of UAVs Through GPS Spoofing Using Low-Cost SDR Platforms. *Wirel. Pers. Commun.* **2020**, *115*, 2729–2754. [CrossRef]
91. Ezuma, M.; Erden, F.; Anjinappa, C.K.; Ozdemir, O.; Guvenc, I. Micro-UAV Detection and Classification from RF Fingerprints Using Machine Learning Techniques. In Proceedings of the 2019 IEEE Aerospace Conference; IEEE: Piscataway, NJ, USA, 2019; pp. 1–13. [CrossRef]
92. Digulescu, A.; Despina-Stoian, C.; Stănescu, D.; Popescu, F.; Enache, F.; Ioana, C.; Rădoi, E.; Rîncu, I.; Șerbănescu, A. New Approach of UAV Movement Detection and Characterization Using Advanced Signal Processing Methods Based on UWB Sensing. *Sensors* **2020**, *20*, 5904. [CrossRef]
93. Bisio, I.; Garibotto, C.; Lavagetto, F.; Sciarrone, A.; Zappatore, S. Unauthorized Amateur UAV Detection Based on WiFi Statistical Fingerprint Analysis. *IEEE Commun. Mag.* **2018**, *56*, 106–111. [CrossRef]
94. Choudhary, G.; Sharma, V.; Gupta, T.; Kim, J.; You, U. Internet of Drones (IoD): Threats, Vulnerability, and Security Perspectives. In Proceedings of the MobiSec 2018: The 3rd International Symposium on Mobile Internet Security, Cebu, Philippines, 29 August–1 September 2018.
95. Noura, H.N.; Salman, O.; Chehab, A.; Couturier, R. DistLog: A distributed logging scheme for IoT forensics. *Ad Hoc Networks* **2019**, *98*, 102061. [CrossRef]
96. Cohen, R.S. The Drone Zappers, Air Force Magazine. 22 March 2019. Available online: <https://www.airforcemag.com/article/the-drone-zappers/> (accessed on 18 August 2022).
97. Mizokami. Air Force Downs Several Drones with New ATHENA Laser Weapon System, Popular Mechanics. 8 November 2019. Available online: <https://www.popularmechanics.com/military/research/a29727696/athena-laser-weapon/> (accessed on 18 August 2022).
98. Federal Aviation Administration, Become a Drone Pilot, Become a Pilot. 19 May 2021. Available online: [https://www.faa.gov/uas/commercial\\_operators/become\\_a\\_drone\\_pilot/](https://www.faa.gov/uas/commercial_operators/become_a_drone_pilot/) (accessed on 18 August 2022).
99. Transport Canada, Where to Fly your Drone, Drone Safety. 19 February 2021. Available online: <https://tc.canada.ca/en/aviation/drone-safety/where-fly-your-drone> (accessed on 18 August 2022).
100. Ch, R.; Srivastava, G.; Gadekallu, T.R.; Maddikunta, P.K.R.; Bhattacharya, S. Security and privacy of UAV data using blockchain technology. *J. Inf. Secur. Appl.* **2020**, *55*, 102670. [CrossRef]
101. Sinha Satyajit, Securing IoT with Blockchain, Counterpoint. 11 May 2018. Available online: <https://www.counterpointresearch.com/securing-iot-blockchain/> (accessed on 18 August 2022).
102. Wang, W.; Xu, H.; Alazab, M.; Gadekallu, T.R.; Han, Z.; Su, C. Blockchain-Based Reliable and Efficient Certificateless Signature for IIoT Devices. *IEEE Trans. Ind. Inform.* **2021**, *18*, 7059–7067. [CrossRef]
103. Zhang, L.; Zou, Y.; Wang, W.; Jin, Z.; Su, Y.; Chen, H. Resource allocation and trust computing for blockchain-enabled edge computing system. *Comput. Secur.* **2021**, *105*, 102249. [CrossRef]
104. Zhang, L.; Peng, M.; Wang, W.; Jin, Z.; Su, Y.; Chen, H. Secure and efficient data storage and sharing scheme for blockchain-based mobile-edge computing. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4315. [CrossRef]
105. Shao, X.; Yue, X.; Liu, J. Distributed adaptive formation control for underactuated quadrotors with guaranteed performances. *Nonlinear Dyn.* **2021**, *105*, 3167–3189. [CrossRef]
106. Li, M.; Guo, C.; Yu, H.; Yuan, Y. Event-triggered containment control of networked underactuated unmanned surface vehicles with finite-time convergence. *Ocean Eng.* **2022**, *246*, 110548. [CrossRef]
107. Rahman, Z.; Yi, X.; Khalil, I. Blockchain based AI-enabled Industry 4.0 CPS Protection against Advanced Persistent Threat. *IEEE Internet Things J.* **2022**. [CrossRef]
108. Yu, S.; Das, A.K.; Park, Y.; Lorenz, P. SLAP-IoD: Secure and Lightweight Authentication Protocol Using Physical Unclonable Functions for Internet of Drones in Smart City Environments. *IEEE Trans. Veh. Technol.* **2022**. [CrossRef]
109. Gaurav, B.; Kumar, D.; Vidyarthi, D.P. BARA: A blockchain-aided auction-based resource allocation in edge computing enabled industrial internet of things. *Future Gener. Comput. Syst.* **2022**.

110. Yuan, L.; Zhang, Y.; Wang, J.; Xiang, W.; Xiao, S.; Chang, L.; Tang, W. Performance analysis for covert communications under faster-than-Nyquist signaling. *IEEE Commun. Lett.* **2022**.
111. Zhang, W.; Shao, X.; Zhang, W.; Qi, J.; Li, H. Unknown input observer-based appointed-time funnel control for quadrotors. *Aerosp. Sci. Technol.* **2022**, *126*, 107351. [CrossRef]
112. Challita, U.; Ferdowsi, A.; Chen, M.; Saad, W. Machine Learning for Wireless Connectivity and Security of Cellular-Connected UAVs. *IEEE Wirel. Commun.* **2019**, *26*, 28–35. [CrossRef]
113. Sanjab, A.; Saad, W.; Basar, T. Prospect theory for enhanced cyber-physical security of drone delivery systems: A network interdiction game. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6. [CrossRef]

## Article

# Detection of Micro-Doppler Signals of Drones Using Radar Systems with Different Radar Dwell Times

Jiangkun Gong <sup>1</sup> , Jun Yan <sup>1</sup>, Deren Li <sup>1</sup> and Deyong Kong <sup>2,\*</sup>

<sup>1</sup> State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China

<sup>2</sup> School of Information Engineering, Hubei University of Economics, Wuhan 430205, China

\* Correspondence: kdykong@hbue.edu.cn; Tel.: +86-027-68778527

**Abstract:** Not any radar dwell time of a drone radar is suitable for detecting micro-Doppler (or jet engine modulation, JEM) produced by the rotating blades in radar signals of drones. Theoretically, any X-band drone radar system should detect micro-Doppler of blades because of the micro-Doppler effect and partial resonance effect. Yet, we analyzed radar data detected by three radar systems with different radar dwell times but similar frequency and velocity resolution, including Radar- $\alpha$ , Radar- $\beta$ , and Radar- $\gamma$  with radar dwell times of 2.7 ms, 20 ms, and 89 ms, respectively. The results indicate that Radar- $\beta$  is the best radar for detecting micro-Doppler (i.e., JEM signals) produced by the rotating blades of a quadrotor drone, DJI Phantom 4, because the detection probability of JEM signals is almost 100%, with approximately 2 peaks, whose magnitudes are similar to that of the body Doppler. In contrast, Radar- $\alpha$  can barely detect any micro-Doppler, and Radar- $\gamma$  detects weak micro-Doppler signals, whose magnitude is only 10% of the body Doppler's. Proper radar dwell time is the key to micro-Doppler detection. This research provides an idea for designing a cognitive micro-Doppler radar by changing radar dwell time for detecting and tracking micro-Doppler signals of drones.

**Keywords:** cognitive micro-Doppler radar; drone detection; Doppler resolution; JEM signals; radar dwell time

**Citation:** Gong, J.; Yan, J.; Li, D.; Kong, D. Detection of Micro-Doppler Signals of Drones Using Radar Systems with Different Radar Dwell Times. *Drones* **2022**, *6*, 262. <https://doi.org/10.3390/drones6090262>

Academic Editors: Daobo Wang and Zain Anwar Ali

Received: 29 August 2022

Accepted: 17 September 2022

Published: 19 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recently, researching topics about using micro-Doppler to detect, classify, and track radar echoes of drones have been hot spots. The most common drones in these studies are drones with rotating blades, such as single-rotor drones, quadrotor drones, six-rotor drones, and even hybrid vertical take-off and landing (VTOL) drones. They are small in size, fly at a slow speed, and are mainly active at low-altitude airspace [1,2]. The rotating movement of rotating blades can modulate the incident radar wave and produce an additional micro-Doppler on the base of the body Doppler contributed by the flying motion of the drone body. Micro-Doppler signals are thought to be useful signatures for radar applications.

Not any one radar system is suitable for detecting and classifying micro-Doppler in radar signals of drones. Currently, both academia and industry have revealed many drone detection radar solutions. They are a wide and diverse variety of types, such as pulse-Doppler marine radar [3], FMCW (frequency-modulated continuous wave) radar [4–7], millimeter-wave radar [8], CW (continuous wave) radar [9], staring radar [10,11], airborne weather radar [12], multistatic radar [13], wide/ultrawideband radar systems [14,15], and even radar networks [16]. Radar vendors also launch commercial off-the-shelf (COTS) counter-drone radars from generation to generation; for example, there are some commercial drone detection radar systems listed in Table 1. Their radar dwell times can be estimated roughly using the rotating rate of the antennas. Generally, with a faster rotating rate comes a shorter radar dwell time. No matter what drone radar is configured, radar dwell time is

one of many factors to extracting both body Doppler signals and micro-Doppler signals of drones in background clutter.

**Table 1.** Some drone detection radars <sup>1</sup>.

Model (Vendor; Country)	Radar Band	Update Rate (Hz) <sup>2</sup>	Range (km) <sup>3</sup>	Identification Strategy <sup>4</sup>
Retinar FAR-AD (Meteksan; Turkey)	Ku	4/15	4.4	Micro-Doppler
Gamekeeper 16U (AVEILLANT; UK)	L	4	5	Micro-Doppler, tracking data.
A800(Blighter; UK)	Ku	1/4	3	Micro-Doppler
XENTA-M1 (Weibel; Danish)	X	1	10	Range-Doppler, micro-Doppler.
ReGUARD (Retia; Czech Republic)	X	1/4	6	Rada cross section (RCS)
ELM/2026BF (IAI; Israel)	X		5.2	Tracking data
Spyglass™ (Numerica; USA)	Ku			Tracking data
Gryphon R1400/R1410 (SRC; USA)	X		8.5	Tracking data
ELVIRA (Robin; Netherlands)	X	2/3	2.7	AI, micro-Doppler
Giraffe 1X (SAAB; Sweden)	X	1	13	AI, kinematic, RCS micro-Doppler, etc.
GO20 MM (Thales; France)	X	1/6	4	AI, micro-Doppler

<sup>1</sup> These data can be found on their official websites. <sup>2</sup> The update rate is the typical value. Some of them can be selectable. <sup>3</sup> The detection range is for drones with RCS of  $\sim 0.01 \text{ m}^2$ , such as DJI Phantom-4. The classification range is normally shorter than the detection range. <sup>4</sup> The specific identification signatures are not available, and those terms are reported in their official brochures.

Generally, the longer the radar dwell time means the better Doppler resolution. Yet, there is still an upper limitation of radar dwell time for a practical radar sensor. First of all, a radar needs a rotating motion with a rotating rate to achieve the  $360^\circ$  cycle-scanning ability. Then, there is a conflict in radar parameter design, in that a rapid rotating update rate and fast beam scanning will result in a short dwell time, and then poor Doppler resolution. The result of cycle-scanning is that the radar dwell time in one scanning cannot be infinitely long. Second, even if a radar can stare in some direction and obtain a long radar dwell time, the micro-Doppler could migrate between different space resolution cells during the long radar dwell time, and then there is an overlap of micro-Doppler either in the range-Doppler cell or time-Doppler cell. Moreover, Radar dwell time not only affects micro-Doppler but also body Doppler. Since micro-Doppler is the additional Doppler around the body Doppler, the ratio of micro-Doppler to body Doppler is also the factor related to the signal extraction. Although some typical high-resolution algorithms have also been investigated for improving Doppler resolution, such as compressed sensing (CS) [17], minimum variance distortionless response (MVDR) [18], multiple signal classification (MUSIC) [19], iterative adaptive algorithm (IAA) [20], and machine learning (ML) technology [21], radar dwell time is still the base factor related to the micro-Doppler. A balance of a suitable radar dwell time is required for extracting and observing micro-Doppler in radar signals of drones.

In this paper, we investigate the proper radar dwell time for detecting the micro-Doppler signals (i.e., jet engine modulation, JEM) modulated by the rotating blades of drones. In Section 2, we discuss the relationship between radar dwell time and the micro-Doppler produced by the rotating blades theoretically and then introduce the three typical radar dwell times of our three radar systems, i.e., Radar- $\alpha$ , Radar- $\beta$ , and Radar- $\gamma$ . In Section 3 and Section 4, we analyze the detection performance of micro-Doppler detected by the three radars and then propose our explanation of the detection results. Finally, we

conclude our point in Section 5. The objectives of this paper include three points. (1) We argue that not any radar dwell time is suitable for detecting micro-Doppler produced by the rotating blades in radar signals of drones, and the proper radar dwell time depends on the rotating period of the blades of drones. (2) We propose that two parameters can be used for evaluating the detection performance of micro-Doppler modulated by drones' blades, including the JEM number and the ratio of the first blade's magnitude to that of the body. (3) We suggest that a cognitive radar can be designed by adjusting the radar dwell time to detect micro-Doppler signals of drones.

## 2. Materials and Methods

### 2.1. Micro-Doppler of Rotating Blades of Drones

Micro-Doppler is the additional Doppler related to the micromotion in addition to the body Doppler. Figure 1a demonstrates the geometry of a radar and a rotating rotor blade of a quadrotor drone. Assume that the azimuth angle of  $\alpha$  and the elevation angle of  $\beta$  is zero. The shape of a blade is not, and nor can it be seen as, a thin rectangular bar with rotating movement. According to micro-Doppler theory in [22], the received signals of the  $k$ th blade and its micro-Doppler frequency are given by

$$|S_{k,mD}(t)| = \sum_{k=0}^{N-1} L \text{sinc} \left( \frac{2\pi L}{\lambda} \sin \left( \Omega t + \theta_{0,k} + \frac{k2\pi}{N} \right) \right) \quad (1)$$

$$f_{k,mD}(t) = \sum_{k=0}^{N-1} \frac{2\pi L}{\lambda} \Omega \left[ -\sin \left( \theta_{0,k} + \frac{k2\pi}{N} \right) \sin(\Omega t) + \cos \left( \theta_{0,k} + \frac{k2\pi}{N} \right) \cos(\Omega t) \right] \quad (2)$$

where  $N$  is the number of blades,  $L$  is the blade length,  $\lambda$  is the wavelength,  $\Omega$  is the rotation rate, and  $\theta_{0,k}$  is the initial rotation angle. Equations (1) and (2) indicate that such micro-Doppler signals are modulated by the rotation rate  $\Omega$  through two sinusoidal functions. The maximum values of both  $|S_{k,mD}(t)|$  and  $f_{k,mD}(t)$  appear when the direction of the incident wave is perpendicular to the long face of the blade, which means that the rotation angle is  $90^\circ$ , and then we can obtain the "blade flash" signals in the time series and the JEM-like peaks in the spectrum [22–25].

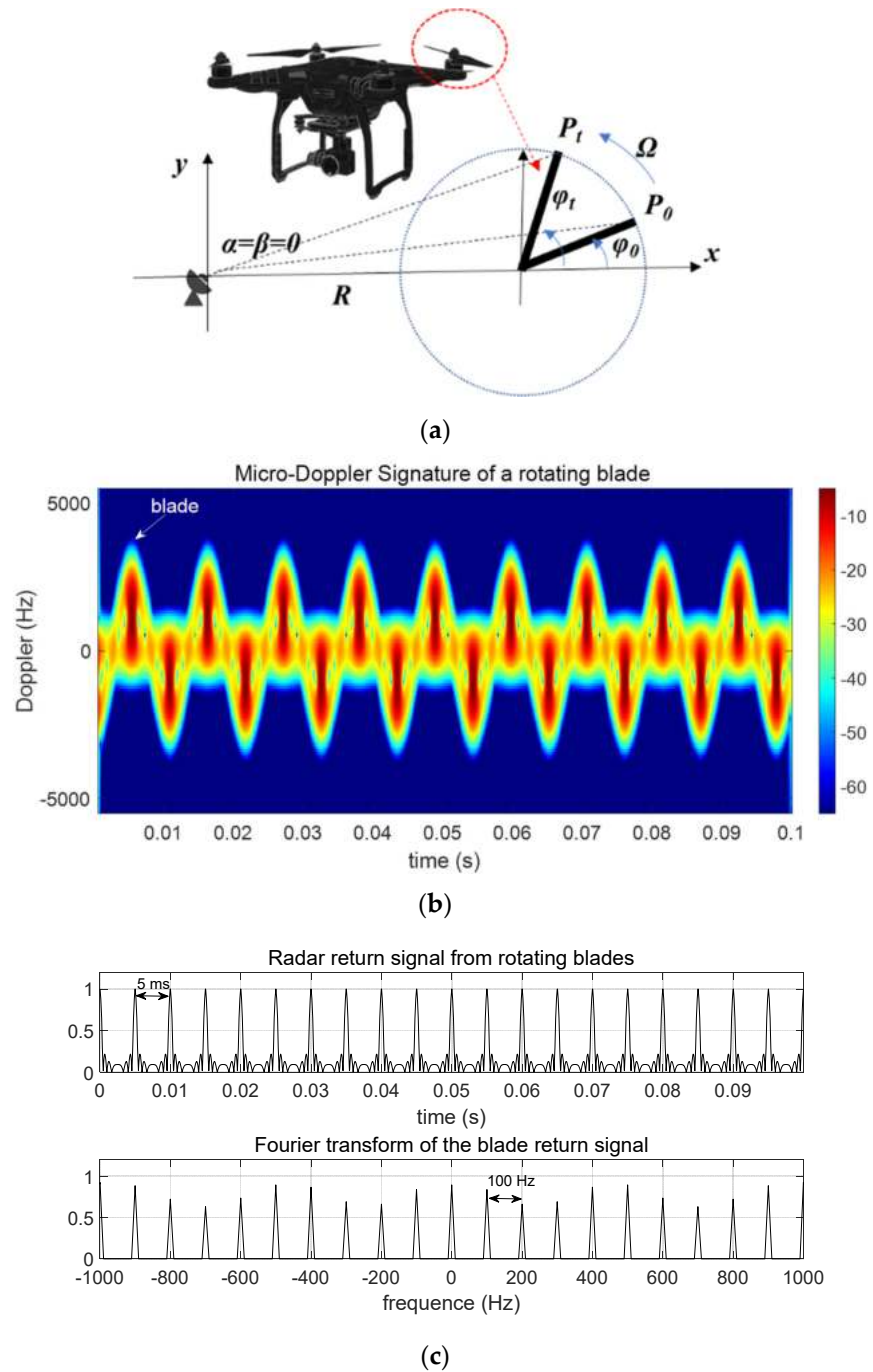
The ideal micro-Doppler or JEM could be simulated when the radar dwell time and frequency resolution are enough. Figure 1b shows the simulated micro-Doppler of rotating blades. The number of blades is 1, and the rotating rate is 100 Hz. The length of a single blade is 0.2 m. The elevation angle is  $15^\circ$ , and the detection range is 20 km. The test band is the X-band, working on 10 GHz. The simulated radar time is 100 ms. Figure 1b shows the flash signals produced by the blade, which are modulated by the rotating rate on the radar images processed by short-time Fourier transform (STFT) algorithm. Figure 1c demonstrates the "blade flash" signals in the time domain with a modulation period of 5 ms and the JEM signals with a frequency interval of about 100 Hz.

Radar dwell time is the key to observing such micro-Doppler. According to Nyquist Theorem, the radar dwell time must be at least longer than twice the rotating period of the rotating blades. Given that the rotating rate of the blades of the drone is  $\Omega$ , the minimum radar dwell time for obtaining such sufficient micro-Doppler is given by

$$T_s = \frac{2}{\Omega} \quad (3)$$

where  $T_s$  is the minimum radar dwell time. If the general rotation rate of blades of drones is 100 Hz, the minimum dwell time is approximately 20 ms. Different radar systems have different radar dwell times. If radar dwell time of radar is much shorter than the required one,  $T_s$ , then we may observe the insufficient micro-Doppler with weaker magnitude and a smaller number of JEM peaks. Furthermore, what happens when the radar dwell time of radar is much longer than the required one? Can we obtain a much clearer micro-Doppler

in the radar signals of drones? What is the best radar dwell time for observing such micro-Doppler?



**Figure 1.** Simulated micro-Doppler of rotating blades within X-band data. (a) The geometry of the radar and the rotating rotor blades, (b) micro-Doppler on the STFT image, (c) blade flash and JEM signals.

To evaluate the detection performance of micro-Doppler (i.e., JEM) of rotating blades of drones, we select some parameters about JEM signatures. They are (1) the number of JEM peaks, and (2) the ratio of the first blade's magnitude to the body's magnitude. They can be given, respectively, by

$$N_{mD} = N - 1 \quad (4)$$

where  $N_{mD}$  is the number of JEM peaks,  $N$  is the number of Doppler peaks in the spectrum, and the 1 represents the body Doppler (i.e., bulk Doppler); and

$$r_{mb} = \frac{A_{mD}}{A_{bD}} \quad (5)$$

where  $r_{mb}$  is the ratio of blade's magnitude to body's magnitude,  $A_{mD}$  is the magnitude of the first neighboring blade Doppler, and  $A_{bD}$  is the magnitude of the body Doppler. Theoretically, the better detection performance of such micro-Doppler (i.e., JEM) means a bigger  $N_{mD}$  and a higher  $r_{mb}$ .

## 2.2. Experimental Conditions

To investigate the detection performance of such micro-Doppler, a software-defined radar platform that can be changed with radar dwell times as well as other parameters should be used for collecting radar data of drones and then seeking the best radar dwell time for detecting micro-Doppler after investigating the relationship between the radar dwell time and micro-Doppler. Unfortunately, we do not have such resources to conduct this research, and we can only use some radar data of drones detected by three radar systems with typical radar dwell times to explore the first step of such a topic. Instead, we obtained some drone detection radar data detected by three radar systems with typical radar dwell times (i.e., insufficient, moderate, and sufficient). Table 2 lists some parameters of the three radar systems. They are pulse-Doppler radar systems equipped with phased-array antennas. The pulse repetition frequency (PRF) and Doppler resolutions of three radar sensors are similar, but Radar- $\alpha$  (the insufficient one) has the shortest radar dwell time in one coherent pulse interval (CPI) of 2.7 ms, Radar- $\beta$  (the moderate one) has a moderate time of 20 ms, and Radar- $\gamma$  (the sufficient one) has the longest time of 89 ms. Thereby, given the general rotating period (i.e., 10 ms) of the blade, Radar- $\alpha$  can detect insufficient micro-Doppler because the radar dwell time is only 27% of the rotating period, Radar- $\gamma$  can detect sufficient micro-Doppler because the radar dwell time is about 4 times than the rotating period, and Radar- $\beta$  may detect either sufficient or insufficient micro-Doppler based on the rotating rate of the blades in actual cases. The quadcopter drone is a DJI Phantom 4, fabricated by DJI Inc., China. It is a small drone with a flight weight of 1.38 kg. Both its body and propellers are mainly composed of plastic. Its wheelbase is approximately 0.35 m. There are four rotor blades with a length of 0.2 m. The cruise speed is approximately 15 m/s. The maximum flight time is approximately 28 min, with a maximum flight height lower than 500 m. The rotating rate of blades is from 5000 RPM to 7000 RPM (revolutions per minute).

**Table 2.** Parameters of the three drone detection radars.

Parameters	Radar- $\alpha$	Radar- $\beta$	Radar- $\gamma$
Radar band	X	X	X
CPI (ms)	2.7	20	89
PRF (kHz)	33.3	5	2.8
Sampling points after zero padding	2048	256	256
Frequency resolution (Hz)	16	19	11
Doppler resolution (m/s)	0.163	0.285	0.165
Range resolution (m)	3.75	12	10
Beamwidth	0.97°	0.72°	2°
Detection range (m)	3000	10,000	6000
Width of the wavefront (m)	50.7	125.6	209.4
Space resolution (m <sup>2</sup> )	190.1	1507.2	2094
Radar dwell time per square meter (ms/m <sup>2</sup> )	0.014	0.013	0.042

The radars collected these data at three areas. The detection background is mainly ground clutter, but the detection ranges were different. The range of the drones from

Radar- $\alpha$  was about 3 km, the one from Radar- $\beta$  was about 10 km, and that from Radar- $\gamma$  was about 6 km. The drones were flying in the radar beams, and the radars worked in a tracking mode. Thereby, we collected tracking data of drones. The drones were flying in a range widow with a size of 1 km, at an altitude below 300 m. The width of the wavefront at the radar range of a target is calculated by the beamwidth and the detection range, which is

$$W = \theta_{re} R \quad (6)$$

where  $W$  is the width of the wavefront,  $\theta_{re}$  is the beamwidth, and  $R$  is the detection range of the target. Thereby, the spatial resolution of the sector where the target is in can be given approximately by

$$Sp_r = WR_{re} \quad (7)$$

where  $Sp_r$  is the spatial resolution and  $R_{re}$  is the range resolution. Table 2 also lists the range resolutions, detection ranges, and space resolutions in the three cases. If we divide the spatial resolution by the CPI time, we can obtain the radar dwell time per square meter values of the three radars, which are 0.014 ms/m<sup>2</sup>, 0.013 ms/m<sup>2</sup>, and 0.042 ms/m<sup>2</sup>, respectively. These numbers are similar to each other. Figure 2 shows an example of detecting and tracking the DJI Phantom 4, using Radar- $\gamma$  working in a tracking mode. The blue solid line was the radar beam, and the blue dotted lines were the tracking trace of the drone when flying towards the radar location. We selected some radar data collected in this area and used them in this paper. Similarly, we also collected some data using Radar- $\alpha$  and Radar- $\beta$  in some other areas.



**Figure 2.** Example of tracking trace of the drone on the radar (i.e., Radar- $\gamma$ ) screenshot.

Although our radar data were collected in a different area, they could still be used for this research for the following reasons. First, we investigated the detection performance of micro-Doppler signals of drones over the radar signals of drones. This means that the radar data of drones were already detected and tracked in the background clutter. Second, we compared the micro-Doppler with the body Doppler; thus, it involves no background clutter. We can also use the normalized magnitudes to remove the interference from the background environment. Thereby, these radar data collected from different ranges in different areas can be used for research to investigate micro-Doppler signals of drones.

### 3. Results

Radar detection is always accompanied by interference from background clutter, and different radar systems with different dwell times can have different performances.

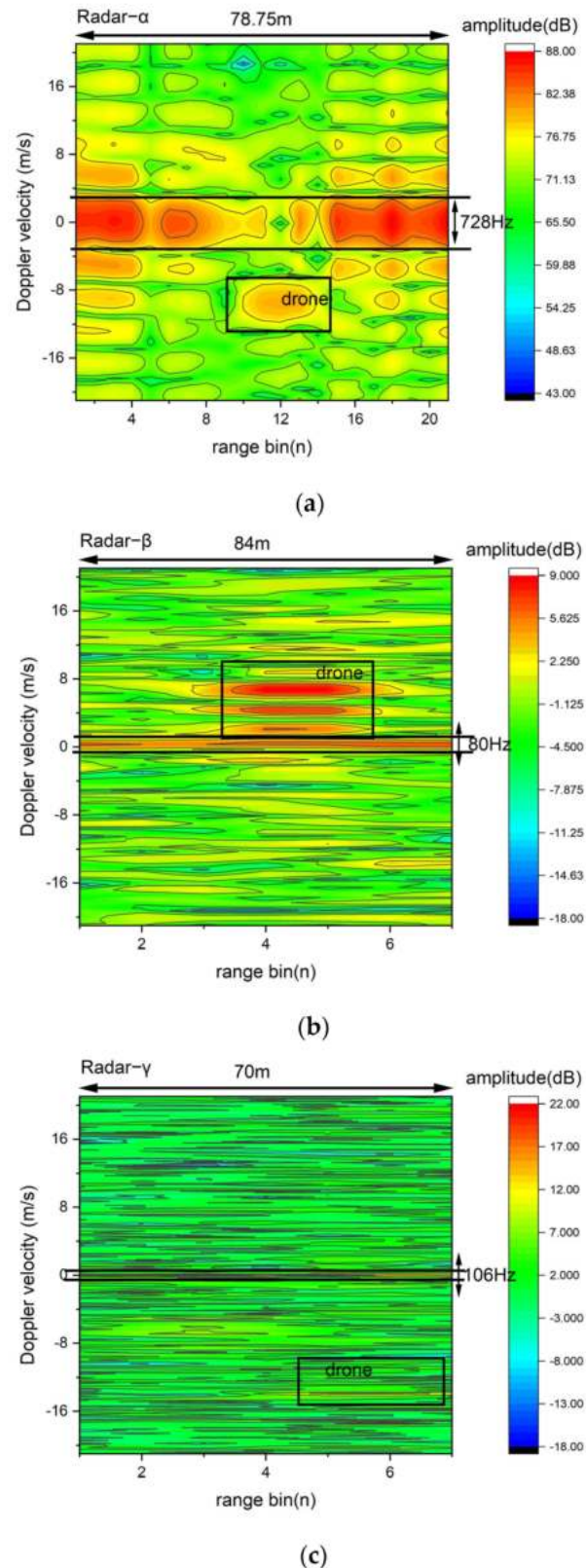
Generally, the longer the radar dwell time, the higher the SNR value of the target. Yet, the noise level of clutter also increases along with the increased radar dwell time. Figure 3 demonstrates radar signals of drones in a similar range window. The black frames in the range-Doppler images mark the range bin that the drone was in. The length values of range windows are 78.75 m, 84 m, and 70 m, respectively. Generally, the ground clutter is mainly around 0 Hz, with different spectral widths, which are 728 Hz, 80 Hz, and 106 Hz when using Radar- $\alpha$ , Radar- $\beta$ , and Radar- $\gamma$ . Although the SNR values of the drone (65.68 dB, 11.43 dB, 11.55 dB) are different in the three cases, its radar echoes can be detected in the clutter. Yet, Radar- $\beta$  can detect the most legible radar signals of the drone because the micro-Doppler signals produced by the rotating blades can also be detected and identified along with the body Doppler (three pots in the black frame in Figure 3b). Other radars seem to detect only body Dopplers of drones on their range-Doppler images (Figure 3a,c). In total, Radar- $\beta$  enjoys a better detection performance for detecting drones than Radar- $\alpha$  and Radar- $\gamma$ .

Micro-Doppler signatures of drones produced by the rotating blades are related directly to the radar dwell time. Figure 4 compares the raw radar signals and the spectrums of drones using the three radar systems. The data are extracted from the range windows in Figure 3. The blues words register the Doppler peaks corresponding to either blades or the body of drones. When the drone is flying away from the radar, its Doppler velocity is negative, and when it is flying approaching the radar, the velocity is positive. Each subfigure shows two cases in which the drone flew in a different direction, relative to the radars. According to Table 2, both the frequency resolution and the velocity resolution of the three radars are similar to each other, but the radar dwell times (i.e., 2.7 ms, 20 ms, and 89 ms) are different. The rotating rate of the blades ranges from 5000–7000 RPM (revolutions per minute). According to Formula (3), the minimum radar dwell time to obtain the sufficient micro-Doppler of the rotating blades is 17–24 ms. Therefore, Radar- $\alpha$  can only detect insufficient micro-Doppler signals of the drone, and Radar- $\beta$  observes either sufficient micro-Doppler signals or insufficient micro-Doppler signals based on the rotating rate of the blades, but Radar- $\gamma$  detected only sufficient micro-Doppler signals.

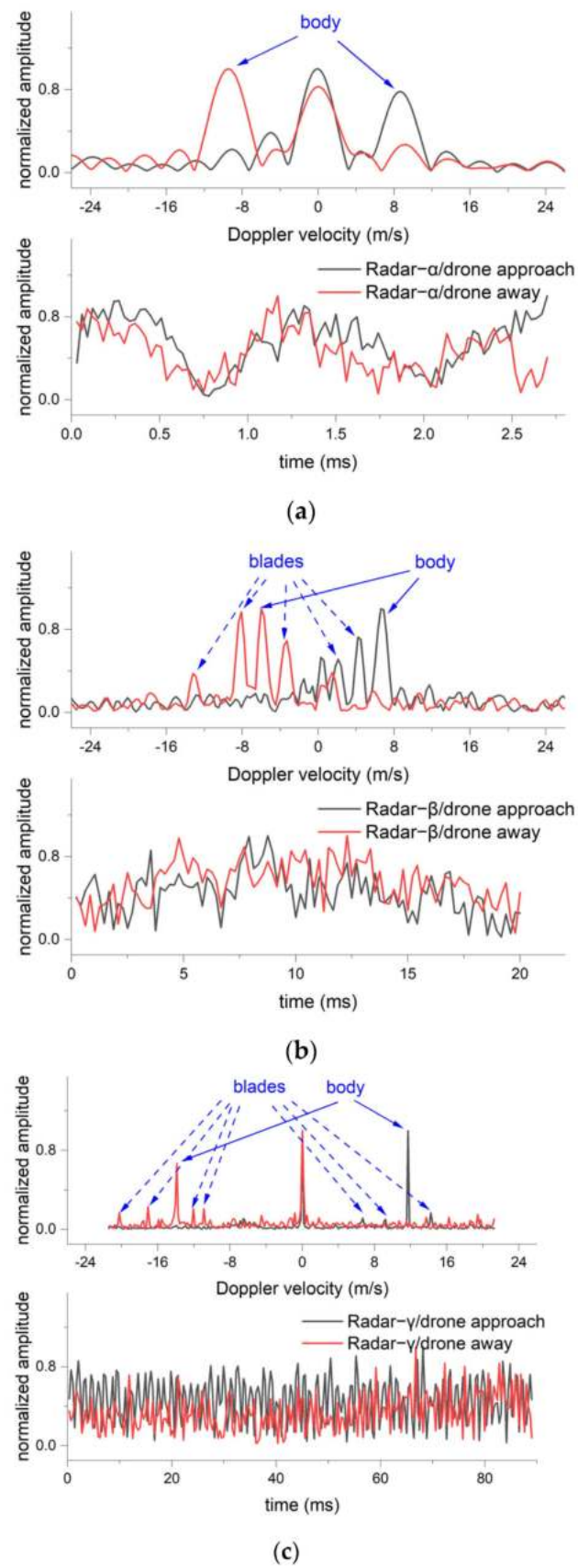
There seems to be only one bulk Doppler (i.e., body Doppler) detected by Radar- $\alpha$ , in Figure 4a, which is  $-9.4$  m/s (or  $+8.6$  m/s). In contrast, several Doppler peaks including one body Doppler and two or three micro-Dopplers appear in the spectrums detected by Radar- $\beta$  and Radar- $\gamma$ , which are  $-13.2$  m/s,  $-8.1$  m/s,  $-6$  m/s,  $-3.3$  m/s (or  $2.1$  m/s,  $4.2$  m/s,  $6.9$  m/s) in Figure 4b, and  $-20.2$  m/s,  $-17.0$  m/s,  $-13.8$  m/s,  $-12.0$  m/s,  $-10.8$  m/s (or  $6.6$  m/s,  $9.2$  m/s,  $11.7$  m/s,  $14.2$  m/s) in Figure 4c. The number of micro-Dopplers (i.e.,  $N_{mD}$ ) detected by Radar- $\gamma$  in Figure 4c is about four, and the number detected by Radar- $\beta$  in Figure 4b is about three. Moreover, the ratio of strongest micro-Doppler magnitude to that of body-Doppler (i.e.,  $r_{mb}$ ) in Figure 4b is about 1, but this number decreases below 0.2 in Figure 4c. Thereby, Radar- $\gamma$  seems to be able to detect more micro-Dopplers than Radar- $\beta$ , but Radar- $\beta$  can detect much stronger micro-Doppler than Radar- $\gamma$ . Yet, Radar- $\alpha$  detected the poorest micro-Doppler. This means that a moderate dwell time is better for the micro-Doppler of drones.

It is not proper that the longest radar dwell times come with the best detection performance of micro-Doppler signals. Figure 5 presents the tracking Doppler signals of drones using the three radar systems. The tracking intervals are different from each other. The black dotted curves in Figure 5 describe the changing body Doppler of drones in the three cases. First, similar to the range-Doppler images in Figure 3, there is always clutter when detecting drones, and the Doppler of background clutter mainly stays around 0 m/s. Doppler detection can separate the radar signals of drones from the clutter with small velocities. Second, Radar- $\beta$  can track more enriched micro-Doppler signatures than Radar- $\alpha$  and Radar- $\gamma$ . There are always attendant spots around the body Doppler in Figure 5b, which represent the distributed pattern of micro-Doppler modulated by the rotating blades of drones. Yet, these micro-Doppler spots seem to disappear on the images in Figure 5a,c. As we stated in Figure 3, Radar- $\alpha$  cannot detect insufficient micro-Doppler,

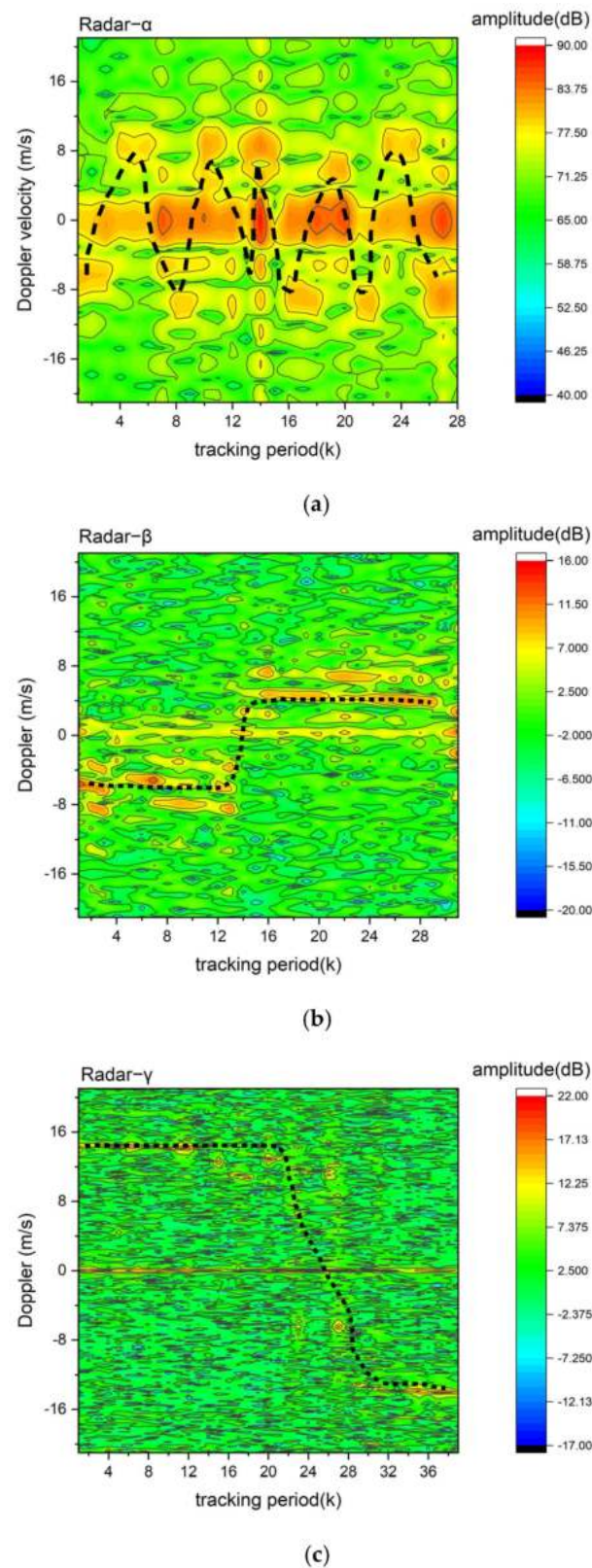
and then the micro-Doppler spots disappear in the tracking results. However, Radar- $\gamma$  can detect very strong body Doppler, which is much stronger than micro-Dopplers, and then the micro-Dopplers are suppressed by the body Dopplers and hidden in the images of Figure 5c.



**Figure 3.** Range-Doppler data of drones. (a) Radar- $\alpha$ , (b) Radar- $\beta$ , (c) Radar- $\gamma$ .



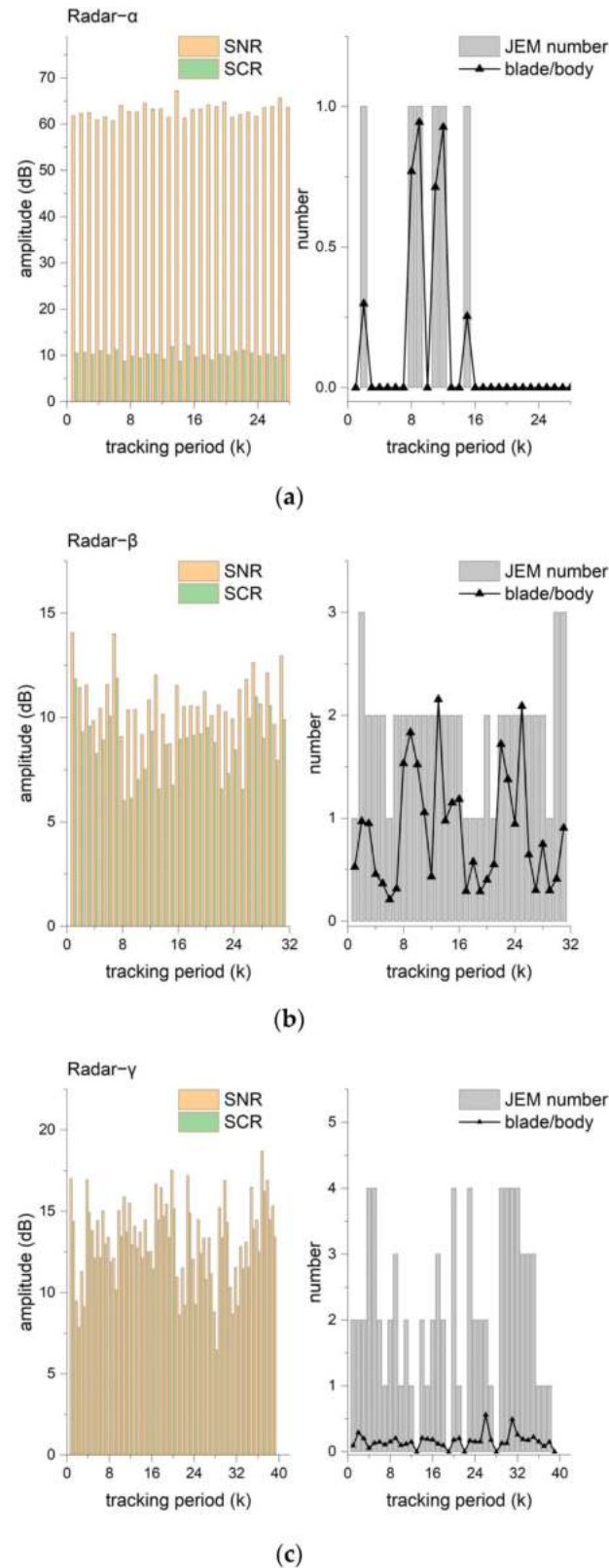
**Figure 4.** Radar signals and spectrums of drones. (a) Radar- $\alpha$ , (b) Radar- $\beta$ , (c) Radar- $\gamma$ .



**Figure 5.** Tracking Doppler data of drones. (a) Radar- $\alpha$ , (b) Radar- $\beta$ , (c) Radar- $\gamma$ .

The quantification analysis of tracking results indicates that compared to Radar- $\alpha$  and Radar- $\gamma$ , Radar- $\beta$  with moderate radar dwell time is a better solution for detecting and tracking micro-Doppler signals of drones among the three radars. Figure 6 demonstrates the tracking parameters of three cases in Figure 5. SNR means signal-to-noise ratio, which

describes the scattering power of a target, and SCR means signal-to-clutter ratio, which presents the scattering superiority of a target to the clutter. The JEM number is the number of micro-Doppler peaks in the spectrum, and blade/body means the ratio of the micro-Doppler's magnitude to the body Doppler's.



**Figure 6.** Detection results of drones. (a) Radar- $\alpha$ , (b) Radar- $\beta$ , (c) Radar- $\gamma$ .

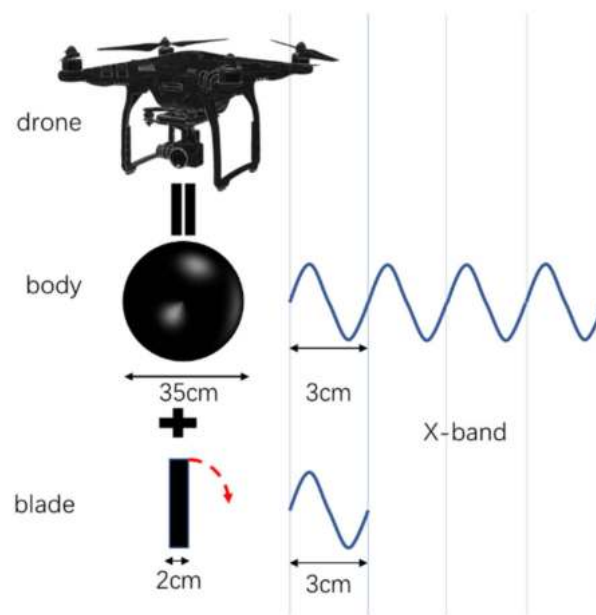
Table 3 shows the statical data of results in Figure 6. First, both SNR and SCR values are fluctuating in three cases. Since different radar systems have different transmitted power levels and noise levels, the SNR values of drones can fluctuate considerably. For example, the mean SNR of drones detected by Radar- $\alpha$  is 63.01 dB, about six times that of the 10.98 dB detected by Radar- $\beta$ . Yet, SCR seems to be much more stable, with only a range of less than 4 dB. To this degree, SCR is a better value for detecting radar signals of targets. Second, Radar- $\alpha$  can detect possible micro-Doppler in some cases (e.g., #2, #8, #9, etc.) and Radar- $\gamma$  sometimes detects no micro-Doppler (e.g., #13, #19, #2, etc.), but Radar- $\beta$  can always detect micro-Doppler in all sampling cases. Thereby, the detection probabilities of JEM detected by the three radars are 21.42%, 100%, and 12.82%, respectively. Third, the numbers of micro-Doppler detected by Radar- $\beta$  and Radar- $\gamma$  are similar at 2, much bigger than the 0.18 detected by Radar- $\alpha$ . It means that as long as the micro-Doppler is detected by the radar, there are at least two JEM peaks. Fourth, Radar- $\beta$  can detect the strongest micro-Dopplers with the ratio of blade's signal to body's signal of 0.88, but the number (i.e., 0.16) is very small in the cases detected by Radar- $\gamma$ . In some words, even if the micro-Doppler is in the radar signals, it can be neglected by an extraction algorithm. Although the number of cases containing micro-Doppler detected by Radar- $\alpha$  is only 6 among the whole 28, the ratio of blade's signal to body's signal is still about 0.65. Fifth, the frequency offsets between the body's Doppler and the first neighboring blade's Doppler are 469 Hz, 165 Hz, and 158 Hz, respectively.

**Table 3.** Comparison of radar detection of a drone1.

Contents	Radar- $\alpha$	Radar- $\beta$	Radar- $\gamma$
Detection range (km)	~3 km	~10 km	~6 km
Doppler velocity (m/s)	8.29	4.70	13.00
Mean SNR (dB)	63.01	10.98	14.22
Mean SCR (dB)	10.23	8.71	12.17
Probability of JEM signals	21.42%	100%	12.82%
Number of JEM peaks	0.18	1.87	2.05
The ratio of the blade's magnitude to that of the body	0.65	0.88	0.16
Frequency offset between blade and body (Hz)	469	165	158

#### 4. Discussion

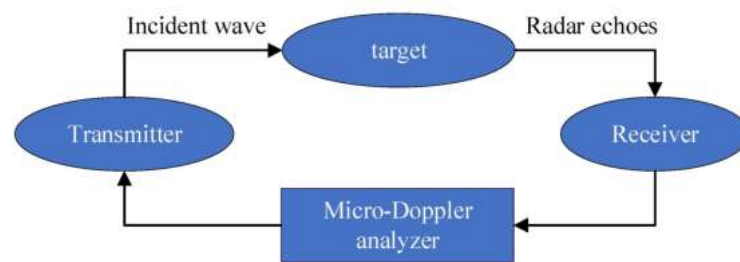
Why can Radar- $\beta$  with moderate radar dwell time detect the best micro-Doppler among the three radar systems? Radar dwell time and radar wavelength are the key factors. Micro-Doppler is the additional Doppler related to the micromotion of the microcomponent on the body of a target. For the drone, the blades are the microcomponent, and the rotating motion is the source of the additional Doppler, as shown in Figure 7. According to Equations (1) and (2), the maximum values of  $|S_{k,mD}(t)|$  and  $f_{k,mD}(t)$  appear when the direction of the incident wave is perpendicular to the long face of the blade, which means that the rotation angle is  $90^\circ$ , and then “blade flash” signals appear in the time series, and JEM-like peaks occur in the spectrum. Furthermore, the partial resonance effect will also contribute to the scattering power of blades. It is known that the resonance effect occurs when the sizes of a target (e.g., drones) are comparable with the radar wavelengths, so their scattering properties are calculated via Mie theory [26,27]. As such, the scattering power of the target is an oscillating function of the size, the materials contents, and the wavelength so that the radar reflectivity values can be amplified at another wavelength. Since the wavelength of the X-band is similar to the width of the blade of the drone, and when the transmitted wave is shot directly onto the blade in the direction of perpendicular to the transmission, the partial resonance effect will amplify only the scattering power of the blades. The micro-Doppler effect and the partial resonance effect together cause the strong JEM in the spectrum.



**Figure 7.** Diagram showing radar wave intervals with different structures of a quadrotor drone.

The different coherent integration times (i.e., radar dwell time) on the drone's body and blades also affect the scattering power from the body and the blades. It is known that if the coherent integration is performed and the magnitudes of the returns from all  $N$  pulses are added, the SNR of a target increases as  $N$ . Note that due to the rotating motion of the blades, the real radar dwell times on the blades are always shorter than the body. In our cases, the three radar systems (i.e., Radar- $\alpha$ , Radar- $\beta$ , Radar- $\gamma$ ) have radar dwell times of 2.7 ms, 20 ms, and 89 ms, respectively, and the rotating period of the blades of drones is about 10 ms. Only the radar dwell time of 20 ms of Radar- $\beta$  is similar to the two-rotation period of the blades, and then the magnitude of the blade Doppler is similar to that of the body Doppler because of the partial resonance effect. In contrast, the radar dwell time of 2.7 ms of Radar- $\alpha$  is too short, so there is only a 10% probability of "blade flash" signals (or JEM signals) contributed by the micro-Doppler effect and the partial resonance effect. Furthermore, the radar dwell time of 89 ms of Radar- $\gamma$  is about four times the required one. During this period, the blades have eight rotating periods, which means that the magnitude of the blade Doppler is smaller than 1/4 of the body Doppler's magnitude. Thereby, due to improper radar dwell time, Radar- $\alpha$  and Radar- $\gamma$  are not suitable for detecting micro-Doppler of drones.

The promising application of this finding that proper radar dwell time is the key to detecting micro-Doppler of targets is to design the cognitive radar systems detecting micro-Doppler by adjusting the radar dwell time. Cognitive radar systems use adaption between the information extracted from the sensor and the transmission of subsequent illuminating waveforms. A practicable cognitive radar is to use the micro-Doppler information and then enjoy the best detection performance. As shown in Figure 8, a cognitive micro-Doppler drone detection radar can change its radar dwell time and PRF to obtain the best performance in detecting micro-Doppler signals of drones. The radar can learn the transmitted parameters of Radar- $\beta$  and adjust the transmitted parameters by using the JEM signatures including the number of JEM Dopplers, and the ratio of the blade's signal to the body's signal. We believe that this new drone detection radar will work well, like Radar- $\beta$  in this paper. In the future, we will continue to conduct related research, design the cognitive micro-Doppler radar for detecting drones using a software-defined radar platform, and evaluate its performance.



**Figure 8.** Block diagram of a cognitive micro-Doppler radar system.

## 5. Conclusions

The radar dwell time of a drone detection radar is the key to detecting micro-Doppler in radar signals of drones. Theoretically, any X-band drone radar system should detect micro-Doppler signals because of the micro-Doppler effect and partial resonance effect. In this paper, we analyze radar data detected by three radar systems with different radar dwell times and similar frequency and velocity resolution. Radar- $\alpha$ , Radar- $\beta$ , and Radar- $\gamma$  have radar dwell times of 2.7 ms, 20 ms, and 89 ms, respectively. We use two parameters to evaluate the detection performance of micro-Doppler using the three radar systems, including the number of JEM peaks (See Equation (4)), and the ratio of the first blade's magnitude to the body's magnitude (See Equation (5)). The detection results indicate that Radar- $\beta$  is the best radar for detecting micro-Doppler (i.e., JEM signals) produced by the rotating blades of a quadrotor drone, DJI Phantom 4, because the probability of the JEM signals is almost 100%, with approximately 2 peaks, whose magnitudes are similar to that of the body Doppler. In contrast, Radar- $\alpha$  can barely detect any micro-Doppler, and Radar- $\gamma$  detects weak micro-Doppler signals, whose magnitude is only 10% of the body Doppler's. Furthermore, the best radar dwell time for detecting such micro-Doppler is similar to the two-rotation period of the blades. Our findings demonstrate that micro-Doppler signals could be used for designing a cognitive radar for detecting and tracking micro-Doppler signals of drones.

**Author Contributions:** Conceptualization, J.Y.; methodology, J.G.; software, D.K.; validation, J.Y.; formal analysis, J.G.; investigation, J.Y.; resources, D.L.; data curation, J.Y.; writing—original draft preparation, J.G.; writing—review and editing, D.K.; visualization, D.K.; supervision, J.Y.; project administration, D.L.; funding acquisition, D.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received some support from the Natural Science Foundation of Hubei Providence (General Program: 2021CFB309).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Some of the data presented in this study may be available on request from the corresponding author. The data are not publicly available due to the internal restriction of the research group.

**Acknowledgments:** We appreciate both the testers during the collection of the data, and we also want to thank the authors whose photographs are reproduced in this study. Furthermore, we would like to thank Huiping Hu for her help with processing the figures in this paper.

**Conflicts of Interest:** The authors declare that they have no conflict of interest.


## References

1. Musa, S.A.; Abdullah, R.S.A.R.; Sali, A.; Ismail, A.; Rashid, N.E.A.; Ibrahim, I.P.; Salah, A.A. A review of copter drone detection using radar systems. *Def. S&T Tech. Bull.* **2019**, *12*, 16–38.
2. Wellig, P.; Speirs, P.; Schuepbach, C.; Oechslin, R.; Renker, M.; Boeniger, U.; Pratisto, H. Radar systems and challenges for C-UAV. In Proceedings of the 2018 19th International Radar Symposium (IRS), Bonn, Germany, 20–22 June 2018; pp. 1–8.

3. Galati, G.; Pavan, G. Calibration of an X-band commercial radar and reflectivity measurements in suburban areas. *IEEE Aerosp. Electron. Syst. Mag.* **2019**, *34*, 4–11. [CrossRef]
4. Roldan, I.; del-Blanco, C.R.; Duque de Quevedo, Á.; Ibañez Urzaiz, F.; Gismero Menoyo, J.; Asensio López, A.; Berjón, D.; Jaureguizar, F.; García, N. DopplerNet: A convolutional neural network for recognising targets in real scenarios using a persistent range–Doppler radar. *IET Radar Sonar Navig.* **2020**, *14*, 593–600. [CrossRef]
5. De Wit, J.J.M.; Gusland, D.; Trommel, R.P. Radar Measurements for the Assessment of Features for Drone Characterization. In Proceedings of the 2020 17th European Radar Conference (EuRAD), Utrecht, The Netherlands, 10–15 January 2021; pp. 38–41. [CrossRef]
6. Park, J.; Jung, D.H.; Bae, K.B.; Park, S.O. Range-Doppler Map Improvement in FMCW Radar for Small Moving Drone Detection Using the Stationary Point Concentration Technique. *IEEE Trans. Microw. Theory Tech.* **2020**, *68*, 1858–1871. [CrossRef]
7. Zulkifli, S.; Balleri, A. Design and Development of K-Band FMCW Radar for Nano-Drone Detection. In Proceedings of the 2020 IEEE Radar Conference (RadarConf20), Florence, Italy, 21–25 September 2020; pp. 1–5.
8. Balal, N.; Richter, Y.; Pinhasi, Y. Identifying low-RCS targets using micro-Doppler high-resolution radar in the millimeter waves. In Proceedings of the 14th European Conference on Antennas and Propagation, EuCAP 2020, Copenhagen, Denmark, 15–20 March 2020; pp. 1–5.
9. Bjorklund, S.; Wadstromer, N. Target Detection and Classification of Small Drones by Deep Learning on Radar Micro-Doppler. In Proceedings of the 2019 International Radar Conference, RADAR 2019, Toulon, France, 23–27 September 2019; pp. 1–6.
10. Beasley, P.; Ritchie, M.; Griffiths, H.; Miceli, W.; Inggs, M.; Lewis, S.; Kahn, B. Multistatic Radar Measurements of UAVs at X-band and L-band. In Proceedings of the 2020 IEEE Radar Conference (RadarConf20), Florence, Italy, 21–25 September 2020; pp. 1–6.
11. Jahangtr, M.; Atkinson, G.M.; Antoniou, M.; Baker, C.J.; Sadler, J.P.; Reynolds, S.J. Measurements of Birds and Drones with L-Band Staring Radar. In Proceedings of the 2021 21st International Radar Symposium (IRS), Berlin, Germany, 21–22 June 2021; pp. 1–10.
12. Blake, W.; Burger, I. Small Drone Detection Using Airborne Weather Radar. In Proceedings of the 2021 IEEE Radar Conference (RadarConf21), Atlanta, GA, USA, 7–14 May 2021; pp. 1–4.
13. Palamà, R.; Fioranelli, F.; Ritchie, M.; Inggs, M.; Lewis, S.; Griffiths, H. Measurements and discrimination of drones and birds with a multi-frequency multistatic radar system. *IET Radar Sonar Navig.* **2021**, *15*, 841–852. [CrossRef]
14. Mizushima, T.; Nakamura, R.; Hadama, H. Reflection characteristics of ultra-wideband radar echoes from various drones in flight. In Proceedings of the 2020 IEEE Topical Conference on Wireless Sensors and Sensor Networks, WiSNeT 2020, Antonio, TX, USA, 26–29 January 2020; pp. 30–33.
15. Huang, A.; Sévigny, P.; Balaji, B.; Rajan, S. Fundamental Frequency Estimation of HERM Lines of Drones. In Proceedings of the 2020 IEEE International Radar Conference (RADAR), Washington, DC, USA, 28–30 April 2020; pp. 1013–1018.
16. Coluccia, A.; Parisi, G.; Fascista, A. Detection and classification of multirotor drones in radar sensor networks: A review. *Sensors* **2020**, *20*, 4172. [CrossRef] [PubMed]
17. Yang, W.Y.; Kim, H.J.; Lee, J.H.; Yang, S.J.; Myung, N.H. Automatic feature extraction from insufficient JEM signals based on compressed sensing method. In Proceedings of the 2015 Asia-Pacific Microwave Conference (APMC), Nanjing, China, 6–9 December 2016; Volume 2, pp. 1–3.
18. Wölfel, M.; McDonough, J. Minimum variance distortionless response spectral estimation. *IEEE Signal Process. Mag.* **2005**, *22*, 117–126. [CrossRef]
19. Elbir, A.M. DeepMUSIC: Multiple Signal Classification via Deep Learning. *IEEE Sens. Lett.* **2020**, *4*, 7001004. [CrossRef]
20. Sun, H.; Oh, B.S.; Guo, X.; Lin, Z. Improving the Doppler Resolution of Ground-Based Surveillance Radar for Drone Detection. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 3667–3673. [CrossRef]
21. Ritchie, M.; Capraru, R.; Fioranelli, F. Dop-net: A micro-Doppler radar data challenge. *Electron. Lett.* **2020**, *56*, 568–570. [CrossRef]
22. Chen, V.C. *The Micro-Doppler Effect in Radar*; Artech House: Norwood, MA, USA, 2011; ISBN 9781608070572/1608070573.
23. Kim, B.K.; Kang, H.S.; Park, S.O. Experimental Analysis of Small Drone Polarimetry Based on Micro-Doppler Signature. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1670–1674. [CrossRef]
24. Molchanov, P.; Harmanny, R.I.A.; De Wit, J.J.M.; Egiazarian, K.; Astola, J. Classification of small UAVs and birds by micro-Doppler signatures. *Int. J. Microw. Wirel. Technol.* **2014**, *6*, 435–444. [CrossRef]
25. Gong, J.; Yan, J.; Li, D.; Chen, R.; Tian, F.; Yan, Z. Theoretical and experimental analysis of radar micro-doppler signature modulated by rotating blades of drones. *IEEE Antennas Wirel. Propag. Lett.* **2020**, *19*, 1659–1663. [CrossRef]
26. Skolnik, M. *Radar Handbook*, 3rd ed.; McGraw-Hill Education: New York, NY, USA, 2008.
27. Tait, P. *Introduction to Radar Target Recognition*; Institution of Electrical Engineers: London, UK, 2006; ISBN 9781849190831.

Article

# ARSD: An Adaptive Region Selection Object Detection Framework for UAV Images

Yuzhuang Wan <sup>1</sup>, Yi Zhong <sup>1</sup>, Yan Huang <sup>2</sup>, Yi Han <sup>1,\*</sup> , Yongqiang Cui <sup>3</sup>, Qi Yang <sup>4</sup>, Zhuo Li <sup>5</sup>, Zhenhui Yuan <sup>6</sup> and Qing Li <sup>7</sup>

<sup>1</sup> School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China

<sup>2</sup> Zmvision Technology, Wuhan 430070, China

<sup>3</sup> College of Electronics and Information, South-Central Minzu University, Wuhan 430074, China

<sup>4</sup> School of Mathematics & Statistics, South-Central Minzu University, Wuhan 430074, China

<sup>5</sup> SAIC GM Wuling Automobile Co., Ltd., Liuzhou 545007, China

<sup>6</sup> Department of Computer and Information Science, Northumbria University, Newcastle upon Tyne NE1 8ST, UK

<sup>7</sup> Peng Cheng Laboratory, Shenzhen 518066, China

\* Correspondence: hanyi@whut.edu.cn; Tel.: +86-27-87858005

**Abstract:** Due to the rapid development of deep learning, the performance of object detection has greatly improved. However, object detection in high-resolution Unmanned Aerial Vehicles images remains a challenging problem for three main reasons: (1) the objects in aerial images have different scales and are usually small; (2) the images are high-resolution but state-of-the-art object detection networks are of a fixed size; (3) the objects are not evenly distributed in aerial images. To this end, we propose a two-stage Adaptive Region Selection Detection framework in this paper. An Overall Region Detection Network is first applied to coarsely localize the object. A fixed points density-based targets clustering algorithm and an adaptive selection algorithm are then designed to select object-dense sub-regions. The object-dense sub-regions are sent to a Key Regions Detection Network where results are fused with the results at the first stage. Extensive experiments and comprehensive evaluations on the VisDrone2021-DET benchmark datasets demonstrate the effectiveness and adaptiveness of the proposed framework. Experimental results show that the proposed framework outperforms, in terms of mean average precision (mAP), the existing baseline methods by 2.1% without additional time consumption.

**Keywords:** UAV; object detection; deep learning; adaptive cluster

**Citation:** Wan, Y.; Zhong, Y.; Huang, Y.; Han, Y.; Cui, Y.; Yang, Q.; Li, Z.; Yuan, Z.; Li, Q. ARSD: An Adaptive Region Selection Object Detection Framework for UAV Images. *Drones* **2022**, *6*, 228. <https://doi.org/10.3390/drones6090228>

Academic Editors: Daobo Wang and Zain Anwar Ali

Received: 30 July 2022

Accepted: 28 August 2022

Published: 31 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Nowadays, as a fast-growing number of Unmanned Aerial Vehicles (UAVs) start carrying high-definition cameras, object detection technology in aerial images has been widely used in various practical applications, including agricultural planting [1,2], pedestrian tracking [3], urban security [4,5], inspecting buildings [6], search and rescue [7], and rare plant monitoring [8]. These applications all require accurate object detection in visible or infrared images taken by onboard cameras. However, detecting objects in UAV images is nontrivial. Varying purposes in different applications and the limited computing power of UAVs have brought challenges to this work. To solve these problems, object detection based on a Convolutional Neural Network (CNN) is gradually applied in UAV detection tasks.

The methods of object detection commonly used today are YOLO series [9–12] and Faster-RCNN [13]. They have achieved a good performance on large-scale datasets such as MS COCO [14], ImageNet [15], and VOC2007/2012 [16]. However, compared to these datasets, UAV images have the following features:

- (1) UAV image datasets often provide higher resolution images, but the objects in these images are always in low resolution. For example, the image size in general image datasets VOC2007/2012 and MS COCO is approximately  $500 \times 400$  and  $600 \times 400$ ,

respectively. However, in the UAV image dataset VisDrone2021-DET [17], the image size is  $2000 \times 1500$  while the object size is only about  $50 \times 50$  pixels.

- (2) The size of the objects depends on the altitude at which the drone takes the image. The higher the drone is, the smaller the object is in the images [18].
- (3) The targets are not evenly distributed. Some regions in an image are plain backgrounds, while other regions are mostly occupied by objects.

To solve these issues, many researchers have attempted to change the structure of the object detection network. Extended from YOLOv5 [19], an improved network named YOLOv5-TPH [20] added a transformer model with attention mechanisms on the detection head of YOLOv5. It trained a  $1536 \times 1536$  high-resolution network on VisDrone2021-DET and achieved 35.74% mAP. However, the high-resolution network and transformer model cost huge computing resources.

Another common solution is to partition a UAV image into several uniform sub-regions and then detect each of them. However, it cannot guarantee effective improvement by directly conducting uniform cropping or random cropping as this cannot locate key sub-regions. Based on the VisDrone2021-DET dataset, the system can achieve 43.5% mAP50 and 30.3% mAP when the sub-regions are obtained via sliding window search [21] and CNN-based clustered sub-network [22], respectively. However, they are either time consuming or training based. Although these detectors could achieve a better performance, they are inefficient at performing detection in every region. Because some regions only have large-scale objects, detecting these regions does not improve the overall accuracy and efficiency. The object-dense sub-regions need to be located first.

This paper proposes the Adaptive Region Selection Detection framework (ARSD), a novel two-stage detection model that combines the Region Detection Network and the Self-adaptive Intensive Region Selecting Algorithm. ARSD aims to significantly reduce computation resource consumption while maintaining high object detection accuracy. The first stage of ARSD uses an Overall Region Detection Network, which can coarsely locate where the target is. The model then applies a Self-adaptive Intensive Region Selecting Algorithm to generate object-dense sub-regions by cluster objects detected in the first stage and sends them to the next stage for further exquisite detection. The last stage is the Key Region Detection Network, which detects the object-dense sub-regions. Based on the first stage, this stage is extended with an additional small detection head based on the original detection heads.

To sum up, the novelty of this paper is as follows:

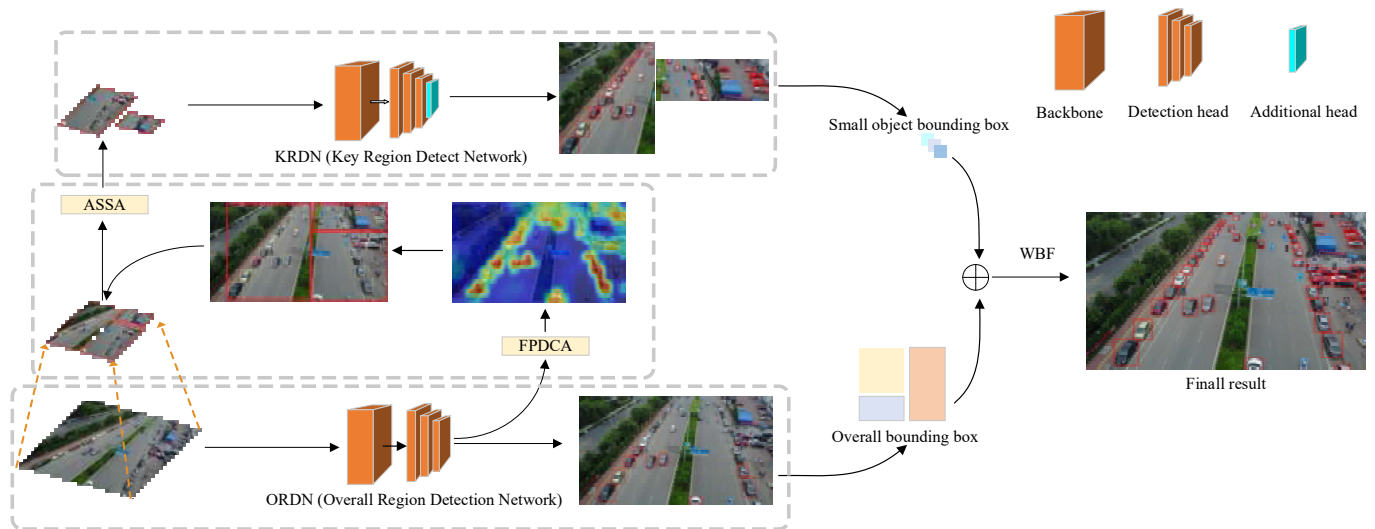
- (1) An effective and efficient object detection framework is proposed to adaptively crop high-resolution UAV images according to object density based on clustering algorithms. This can significantly reduce the training and processing time of the UAV images.
- (2) This paper proposes the Self-adaptive Intensive Region Selecting Algorithm to select the object-dense region in UAV images. It reduces the number of sub-regions for further object detection. This enables the framework to be more suitable for the limited UAV hardware computing power.
- (3) This paper also proposes that an additional detection head is added to deal with the varying object sizes in UAV images. This helps the framework detect small objects more easily and increases detection accuracy.

In this way, the proposed framework can gradually reduce the computational complexity while maintaining high object detection accuracy.

The rest of the paper is organized as follows: Section 2 provides a comprehensive overview of the components of ARSD. The specific experimental details are given in Section 3, which demonstrates the performance of the proposed ARSD in various aspects, as well as comparisons with other works. Section 4 summarizes the experimental results. Finally, Section 5 concludes the paper and lists a collection of ongoing research and future work directions.

## 2. Materials and Methods

This section first introduces the overall framework of the proposed ARSD framework and then describes each module in detail. As shown in Figure 1, the ARSD framework consists of three parts. The first part is the Overall Region Detection Network (ORDN), which is used to roughly locate the objects. The Self-adaptive Intensive Region Selecting Algorithm (SIRSA), which consists of the Fixed Points Density-based Clustering Algorithm (FPDCA) and Adaptive Sub-regions Selection Algorithm (ASSA), is then adopted to properly select the object-dense sub-regions. Finally, the Key Region Detection Network (KRDN) is responsible for detecting the objects in sub-regions selected by SIRSA. The detected objects are combined with the results of ORDN. This framework has better detection accuracy for small targets in UAV images and reduces computing resources due to the filter of the sub-regions.



**Figure 1.** The pipeline of ARSD. The ORDN predicts overall bounding boxes from the original image, which is used for the subsequent region selection. FPDCA can cluster the center point of the overall bounding boxes to obtain candidate sub-regions. ASSA then filters sub-regions that needed to be detected in KRDN. The light blue box represents the additional head of KRDN. Finally, the small object bounding boxes obtained in KRDN are merged with the overall bounding boxes by Weighted Boxes Fusion (WBF) [23] and generate the final result.

### 2.1. Overall Region Detection Network (ORDN) and Key Region Detection Network (KRDN)

The ORDN module predicts object bounding boxes on the whole images, while KRDN predicts the object based on cropped object-dense sub-regions. After cropping sub-regions with SIRSA, KRDN can predict more accurate results in a high-resolution region. The results of ORDN and KRDN are merged by WBF and produce the final results.

It should be mentioned that although ORDN and KRDN are based on the same backbone structure, they differ in width and depth. Therefore, we can easily and inexpensively implement two detection networks with different precision and different time consumption. ORDN only needs to have an accurate recall rate, so it is designed to be more lightweight to save computing power. On the contrary, KRDN is wider and deeper than ORDN. This enables KRDN to have greater accuracy than ORDN without changing the network structure.

For an anchor-based object detection network, anchor size is an important factor affecting the accuracy. The targets in UAV images are on different scales and most are small-scale targets. An additional detection head is added to KRDN based on the original detection head in ORDN. Combined with the original detection head, KRDN can easily locate small objects and reduce the adverse influence caused by different object scales. The added detection head is generated from the low-level feature map of the backbone and the high-resolution feature map obtained by up-sampling in FPN [24]. Generally, feature information will be reduced as the size of the feature map decreases during the processing

of CNN in the backbone. Since the image feature information does not decline gradually in the large-size feature map and more features of small targets are retained, this detection head can help to effectively detect small targets.

The object detection is performed by KRDN after object-dense sub-regions are clustered and selected by SIRSA. KRDN detects the object-dense region and obtains more details about the objects. This paper combines the results from two stages using WBF [23].  $B_o$  and  $B_k$  represent the coordinates of the bounding boxes obtained by ORDN and KRDN, respectively. The final result of the bounding box  $B$  is calculated by (1). The final confidence  $C$  is obtained by (2), where  $C_o$  and  $C_k$  represent the confidence values of the bounding boxes obtained by ORDN and KRDN, respectively.

$$B_{(x,y)} = \frac{B_{o(x,y)} * C_o + B_{k(x,y)} * C_k}{2} \quad (1)$$

$$C = \frac{C_o + C_k}{2} \quad (2)$$

## 2.2. Self-Adaptive Intensive Region Selecting Algorithm (SIRSA)

SIRSA is composed of two parts: (1) the Fixed Points Density-based Clustering Algorithm (FPDCA) is used to cluster the center point of the object predicted in ORDN to obtain candidate sub-regions; (2) the adaptive region selection algorithm selects object-dense sub-regions.

### 2.2.1. Fixed Points Density-Based Clustering Algorithm (FPDCA)

This paper proposes a new clustering method named FPDCA, combining the advantages of two clustering algorithms: K-means [25] and Mean-Shift [26]. K-means is competent at generating fixed clustering centers, but it cannot use density information. Mean-Shift cannot confirm the number of clustering regions, which tends to generate a large number of small-size sub-regions or only one large sub-region. It consumes intensive computation power or cannot be applied to higher resolution images. However, Mean-Shift can perform region clustering based on object-dense information, which is neglected by K-means. Combining K-means and Mean-Shift can take both the density information and generating the fixed number of sub-regions into account.

As shown in Algorithm 1, the set of centers of the bounding box is denoted as  $Q$ , with a point  $q$  in  $Q$  randomly chosen as the initial cluster center. Point  $q$  is updated by calculating the motion vector based on the points in the surrounding circle of radius  $r$ . The algorithm moves to the next Point  $q$  and repeats the above step until all points in  $Q$  have been processed. If the number of cluster centers is greater than the required number of sub-regions  $N$ , K-Means is used to regroup these clusters into  $N$  clusters and then output the  $N$  final candidate sub-regions. If the number of clusters does not exceed  $N$ , these clusters are discarded and K-Means are used directly based on the original  $Q$  points to regroup to  $N$  clusters and the corresponding  $N$  final sub-regions.

### 2.2.2. Adaptive Sub-Regions Selection Algorithm (ASSA)

A large number of candidate sub-regions are produced by FPDCA. To save computing resources on hardware platforms, this paper proposes the Adaptive Sub-regions Selection Algorithm (ASSA) to filter sub-regions that needed to be detected in KRDN. Sub-regions with higher ASSA scores will be selected for further detection by KRDN, while the other sub-regions will be discarded.

Four criteria are defined in ASSA to evaluate whether a candidate sub-region  $p$  requires further detection by KRDN, including regional target density, average confidence, the ratio of bounding boxes' total areas to the sub-region area, and average area of all bounding boxes in the sub-region. Regional target density is considered in [27], as defined in (3), where  $L$  denotes the number of the predicted boxes in  $p$  and  $A$  is the area of  $p$ . However, this definition may not necessarily be accurate in that the size of the original image should also be considered. As shown in Figure 2, when reducing the image size from  $200 \times 200$

(left image) to  $100 \times 100$  (right image), the regional target density should remain the same. However, the result does not conform to this using (3). This paper extends (3) by considering the size of the original image  $S$  in (4).

$$D = \frac{L^2}{A} \quad (3)$$

$$D = \frac{L^2 * S}{A} \quad (4)$$

---

**Algorithm 1:** Fixed Points Density-based Clustering Algorithm
 

---

**Input:**  $N$ : number of sub-regions,  
 $Q$ : the set of bounding box centers,  
 $r$ : distance of algorithm,  
 $\xi$ : threshold of the distance of vector.

**Output:** the set of sub-regions  $S$

```

1: for  $q(x, y)$  in  $Q$  but not in  $M$ 
2:    $M = M \cup q(x, y)$ 
3:   for  $q(x_i, y_i)$  in  $Q$  but not in  $M$ 
4:      $S_k(q) = \{y : (x - x_i)^2 + (y - y_i)^2 < r^2\}$ 
5:      $M = M \cup S_k(q)$ 
6:      $C_i = C_i \cup S_k(q)$ 
7:      $V_{shift} = \frac{1}{k} \sum_{q \in S_k} (q_i - q)$ 
8:      $q = q + V_{shift}$ 
9:     if  $|V_{shift_{new}} - V_{shift_{old}}| < \xi$ 
10:      break from line 3;
11:     end if
12:   end for
13: end for
14: if  $length(C) > N$ 
15:    $S = Kmeans(N, C)$ 
16: else
17:    $S = Kmeans(N, Q)$  as random center
18: end if

```

---



**Figure 2.** The left and right images should have the same target density.

The second criterion is the average confidence  $M$  as defined in (5), where  $score_i$  represents the confidence of bounding box  $i$ . It is considered that the sub-region with low average confidence is due to the classification inaccuracy caused by the lightweight network of ORDNet. It is believed that the smaller the value of  $M$ , the greater the accuracy gain.

$$M = \frac{\sum_{i=1}^L score_i}{L} \quad (5)$$

ASSA then considers the ratio of the sum area of all bounding boxes to the sub-region area  $R$  as defined in (6). The larger the value of  $R$ , the less background in this sub-region and the larger the area that contains objects. In addition, it can also reflect the overlapping degree of the objects to a certain extent, e.g.,  $R > 1$  means that there are overlaps of the object's bounding boxes in this sub-region. Larger  $R$  values indicate that the detected object is big or that many objects are detected within the sub-region.

$$R = \frac{\sum_{i=1}^L area_i}{S} \quad (6)$$

Finally, the ratio of the sum area of all bounding boxes to the number of bounding boxes is defined in (7). It reflects the average size of the objects in the sub-region.

$$E = \frac{\sum_{i=1}^L area_i}{L} \quad (7)$$

As defined in (8), a final score  $s_i$  is computed for each sub-region by adding the above four indicators, each with a corresponding weight calculated using information entropy;  $w_j$  is the weight of each indicator and  $p_{ij}$  is the indicator  $j$  of sub-region  $i$ . The sub-regions with high scores are identified as object-dense sub-regions and sent to the KRDN stage.

$$s_i = \sum_{j=1}^m w_j \cdot p_{ij} \quad (8)$$

The weight of each indicator  $w_j$  is calculated by information entropy  $e_j$  as defined in (9);  $w_j$  is then computed by (10) and (11).

$$e_j = -k \sum_{i=1}^n p_{ij} \ln(p_{ij}), \quad k = 1 / \ln(n) > 0, \quad e_j \geq 0 \quad (9)$$

$$d_j = 1 - e_j \quad (10)$$

$$w_j = \frac{d_j}{\sum_{j=1}^m d_j} \quad (11)$$

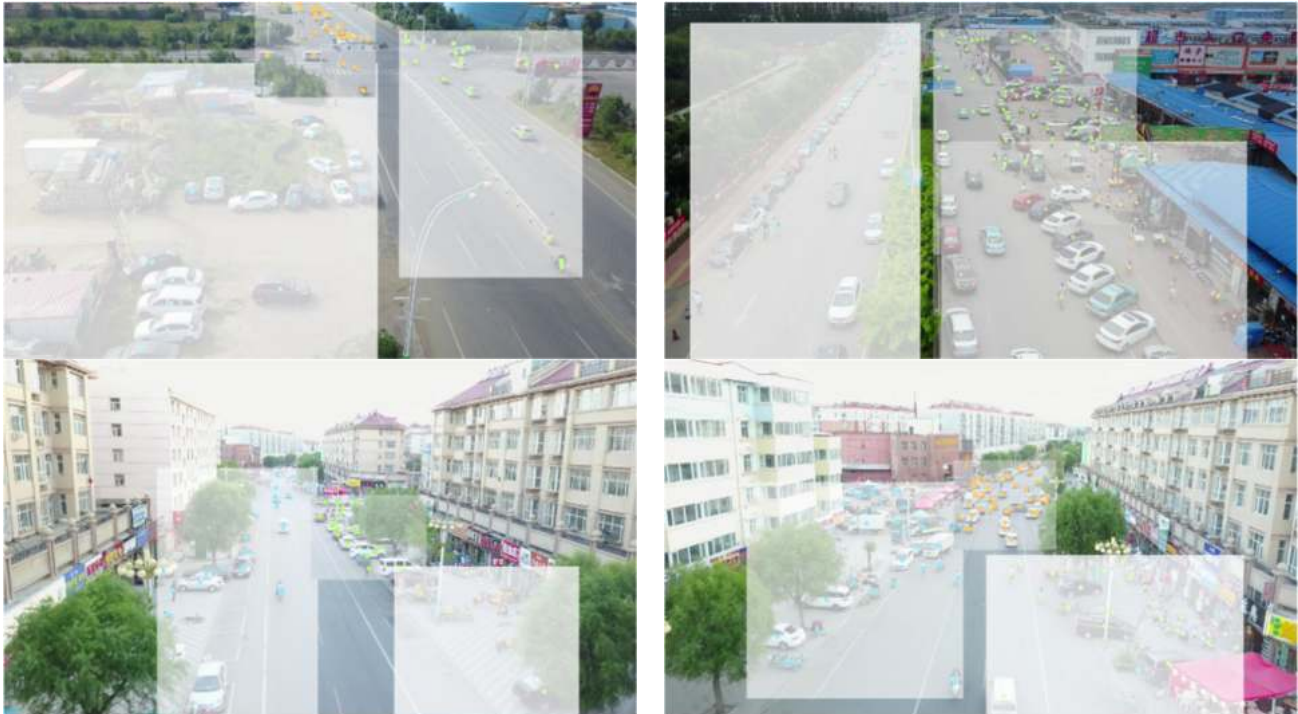
Each final indicator  $j$  of sub-region  $i$ , denoted as  $p_{ij}$ , is calculated by (14) after normalizing the origin indicators using (12) and (13). The higher the regional target density, the more likely that it should be sent to KRDN and the higher the final score. These indicators are defined as extremely large indicators. On the contrary, the smaller the average confidence, the more likely it should be sent to KRDN. These indicators are defined as extremely small indicators. To standardize and normalize these indicators, extremely small indicators are converted by (12) to extremely large indicators and extremely large indicators are normalized by (13).

$$x'_{ij} = \frac{x_{ij} - \min\{x_{ij}, \dots, x_{nj}\}}{\max\{x_{ij}, \dots, x_{nj}\} - \min\{x_{ij}, \dots, x_{nj}\}} \quad (12)$$

$$x'_{ij} = \frac{\max\{x_{ij}, \dots, x_{nj}\} - x_{ij}}{\max\{x_{ij}, \dots, x_{nj}\} - \min\{x_{ij}, \dots, x_{nj}\}} \quad (13)$$

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^n x_{ij}}, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (14)$$

The results of SIRSA are shown in Figure 3. A white rectangle indicates candidate sub-regions, while the transparency of sub-regions indicates their ASSA score. The clearer the sub-region, the higher its final ASSA score and the more it should be sent to KRDN.



**Figure 3.** Points of the same color indicate the center of the bounding box which belongs to the same sub-region. The obscurer the sub-regions, the lower the ASSA score.

### 3. Results

#### 3.1. Datasets and Evaluation Metrics

**Datasets.** The proposed approach is evaluated on the VisDrone2021-DET dataset. The VisDrone dataset is collected by drones in 14 different cities of China, at different heights and in different weather/light conditions. It contains a total of 10,209 images, which consists of 6471 training images, 548 validation images, and 3190 testing images. The objects in this dataset are mostly small and often clustered together. However, the dataset provides a high image resolution of approximately  $2000 \times 1500$  pixels. Images are annotated with bounding boxes, including 10 predefined categories (pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle).

**Evaluation Metric.** The proposed method is evaluated using the same evaluation protocol as described in MS-COCO, which has 12 evaluation indicators for object detection. AP, AP50, and AP75 are selected as the criteria for state-of-the-art comparison. AP is used to calculate the average index of all categories, and it generally defaulted to mAP. AP50 and AP75 represent indicators with threshold IOUs of 0.5 and 0.75, respectively. This paper also adopted the widely used precision–recall (PR) curves as our evaluation metric. The method of [28] is used to calculate the criteria proposed above.

**Implementation Details.** The proposed methods are implemented using PyTorch. The backbone is pretrained on MS COCO. The following training of the model is completed on a server with one NVIDIA GeForce GTX 3080Ti GPU. The experiment uses Adam [29] to train ORD and KRD for 150 epochs, with the learning rate starting at 0.001.

#### 3.2. Model Scaling Scheme for Two-Stage Framework

YOLOv5 has five networks of different sizes, namely YOLOv5x, YOLOv5l, YOLOv5m, YOLOv5s, and up-to-date YOLOv5n. Since it is the most notable and convenient one-stage

detector with a flexible structure, the experiment in this paper chooses YOLOv5 as the basic model for ORDN and KRDN.

We test YOLOv5n and YOLOv5m in different sizes in Section 3.5. The experiment demonstrates that the large-size YOLOv5n is more accurate than the small-size YOLOv5m within the same execution time. Because ORDN is required to locate the bounding box in time, we balance time consumption and accuracy, properly setting YOLOv5n as the basic model of ORDN. Considering the limited performance of the UAV hardware platform, YOLOv5X is unsuitable at this stage due to its high time consumption, despite having the highest performance. YOLOv5m is balanced in its performance and computational requirement, and is therefore adopted as the KRDN base model. Finally, the input size of ORDN is set to  $736 \times 736$  and the sub-regions are resized to  $384 \times 384$  before being fed into KRDN.

### 3.3. Qualitative Result

To demonstrate the effectiveness of the proposed model ARSD, extensive experiments have been conducted to compare its results with the baseline method (YOLOv5) on the VisDrone2021 dataset, as shown in Figure 4. It can be noted that ARSD performs better in small-scale object detection, especially in object-dense sub-regions. In contrast, YOLOv5 misses many small-scale objects in the object-dense region. According to the partially enlarged image, there is almost no missing small-scale objects such as pedestrians. This is mainly because the object-dense sub-regions are on a small scale and do not need to resize dramatically before being sent to KRDN; thus, more features remain for more accurate detection.

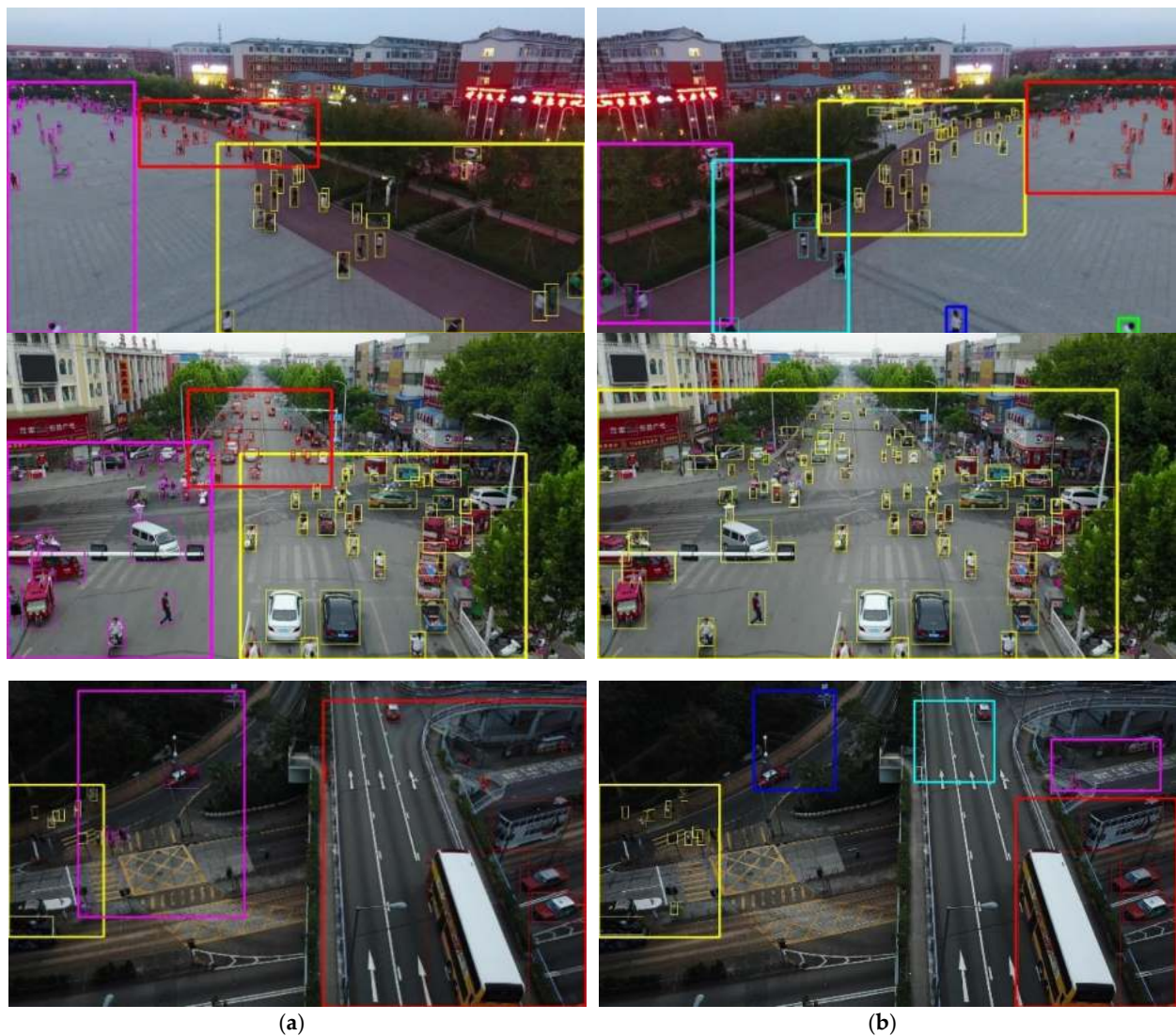


Figure 4. Cont.



**Figure 4.** The qualitative comparisons among the detection results from baseline (b,d) and our method (a,c) on the validation set of the VisDrone2021-DET dataset.

Figure 5 shows the cluster results of the proposed FPDCA and Mean-Shift. The clustering number of Mean-Shift is dynamic; sometimes it has only one cluster. If this one sub-region is sent to KRDN, it cannot achieve any improvement in accuracy. In addition, it sometimes has too many clusters which will result in cost-intensive computing resources.



**Figure 5.** The qualitative comparisons among the clusters result from different cluster methods on the test set of the VisDrone2021-DET dataset. For each testing image, we show the cluster results of the proposed method (a) and Mean-Shift (b).

### 3.4. Quantitative Evaluation

Since the traditional one-stage methods are easier to reproduce, we compare our method with them in each category; however, the comparison with state-of-the-art methods comes from their literature. Quantitative comparisons are conducted in two aspects using the VisDrone2021-DET dataset: (1) the detection accuracy of each category comparisons against different one-stage methods; (2) the detection accuracy and accuracy improvement comparisons against state-of-the-art two-stage framework methods.

Table 1 lists AP50 values for ARSD and other one-stage methods in the detection of 10 different categories of objects. The results of the first five methods are derived from [30], with the remaining results obtained from our conducted experiments. The proposed method, ARSD, consists of YOLOv5n + YOLOv5mAH and performs much better than the other methods in each category, especially in small-scale categories such as Pedestrian and People.

**Table 1.** AP50 comparison of each class among ARSD and other one-stage methods.

Method	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning-Tricycle	Bus	Motor	Average AP50
YOLOV3	18.1	9.9	2	56.6	17.5	17.6	6.7	2.9	32.4	17	17.1
SlimYOLOv3 [31]	17.4	9.3	2.4	55.7	18.3	16.9	9.1	3	26.9	17	17.6
Faster-RCNN	21.7	12.7	11.5	63.2	37.8	29.9	22.5	12.3	50.6	28.4	29.1
FPN	33	25.8	13.9	69.4	40	34.3	27.4	13.4	49.1	37.6	35.6
YOLOv5m	45.2	35.7	13.7	77.8	41	37.9	20.7	10.8	50.9	30.1	36.8
YOLOv5l-TPH	53.5	29.7	25.9	87	55.3	61.5	34.9	31.2	73.5	50.6	50.3
<b>ARSD</b>	<b>68.8</b>	<b>56.8</b>	<b>40.68</b>	<b>88.17</b>	<b>61.53</b>	<b>53.74</b>	<b>49</b>	<b>26.19</b>	<b>72.78</b>	<b>61.3</b>	<b>57.9</b>

As shown in Table 2, AP $\nearrow$ , AP50 $\nearrow$ , and AP75 $\nearrow$  highlight the outperformance of ARSD over other previous state-of-the-art detectors. Compared with GLASN and UCGNet, ARSD improves by 2.1% and 4.8% (AP50).

**Table 2.** Comparison with state-of-the-art object detection methods in the VisDrone2021-DET validation set.

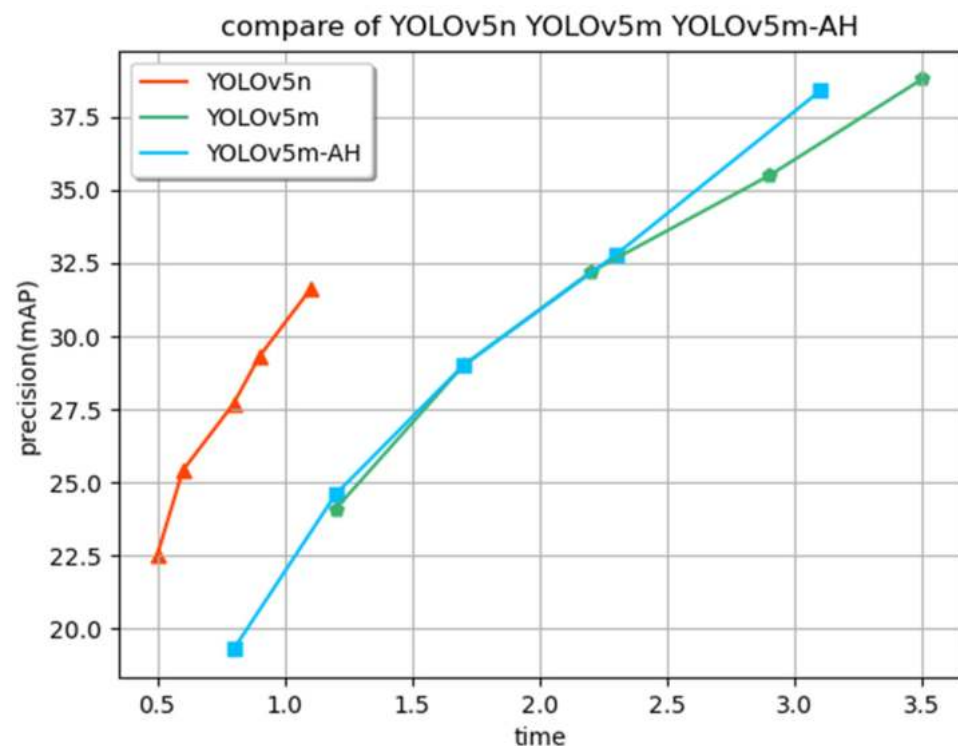
Method	AP	AP50	AP75	AP $\nearrow$	AP50 $\nearrow$	AP75 $\nearrow$
ClusDet [32]	26.7	50.6	24.7	8.34	7.3	11.91
DMNet [33]	28.2	47.6	28.9	6.84	10.3	7.71
UCGNet [34]	32.8	53.1	33.9	2.24	4.8	2.71
GLASN [31]	32.5	55.8	33	2.54	2.1	3.61
<b>ARSD</b>	<b>35.04</b>	<b>57.9</b>	<b>36.61</b>	<b>-</b>	<b>-</b>	<b>-</b>

### 3.5. Ablation Study

To validate the effectiveness of the additional detection head, different cluster methods, and a different number of sub-regions in detection tasks, this paper conducted extensive experiments on VisDrone2021-DET test-dev.

- (1) **Effect of the large-scale and lightweight network.** The results are from different scales YOLOv5n and YOLOv5m, trained on VisDrone2021-DET, as shown in Figure 6. Within the same computation time, YOLOv5n performs better than YOLOv5m. Therefore, we choose the large-scale YOLOv5n as the base model of ORD.
- (2) **Effect of additional prediction head.** Though experiments show that adding a detection head for small-scale objects makes the GFLOPs increase from 48.1 to 59.1, the performance of an additional detection head is prominent. The experiment is to increase the size of the network to compare different accuracy under different time consumption. As shown in Figure 6, YOLOv5m-AH is YOLOv5m with an additional detection head. The mAP of the net with an additional detection head (blue line, YOLOv5m-AH) is 1.5% higher than without an additional head (green line, YOLOv5m) in the same processing time (3.0 ms). It not only saves the computing power of the hardware, but also improves the mAP considerably in each category.

- (3) **Effect of two-stage structure.** To demonstrate the effect of an additional detection head and two-stage framework, this paper chooses three networks with the same computation time:  $768 \times 768$  YOLOv5m,  $640 \times 640$  YOLOv5m-AH, and our two-stage framework ( $768 \times 768$  YOLOv5n +  $384 \times 384$  YOLOv5m-AH). Based on the results shown in Figure 7, our two-stage framework is more accurate in small-scale categories such as Pedestrian, People, and Bicycle. However, results in large-scale categories such as bus and truck are not as good as the one-stage structure. The reason for this phenomenon is that the increase is caused by the additional small object true positives predicted from sub-regions and the decrease is caused by the false positives predicted from sub-regions that match large ground truth boxes.
- (4) **Effect of FPDCA.** The proposed method performs well on clustering, as indicated by Figure 5. In addition, as shown by lines 1, 2, and 6 in Table 3, when using FPDCA as the SIRSA basic cluster method, the result outperforms K-means and Mean-Shift by 0.7% and 1.2%, respectively, in AP50. This shows that it is useful to consider density information when obtaining clusters. The object detection accuracy indicated by AP and AP50 increases as the number of clusters increases from 2 to 4. However, the gain is subtle while the computational complexity also increases. For example, lines 6 and 9 in Table 3 show that AP and AP50 only improves by 0.97% and 0.6%, respectively, when the number of clusters increases from 3 to 4.



**Figure 6.** The comparison between the mAP of YOLOv5n, YOLOv5m, and YOLOv5m-AH for the same time consumption.

To improve processing speed in terms of FPS (frames per second), ASSA is used to discard the sub-regions by  $1/3$ ,  $1/2$ , and  $0$  from the original set of sub-regions. Considering the balance of computing resources and accuracy, ASSA chooses to discard  $1/3$  of the candidate sub-regions in the inference phase. The processing time is reduced by about a third after discarding  $1/3$  of sub-regions.

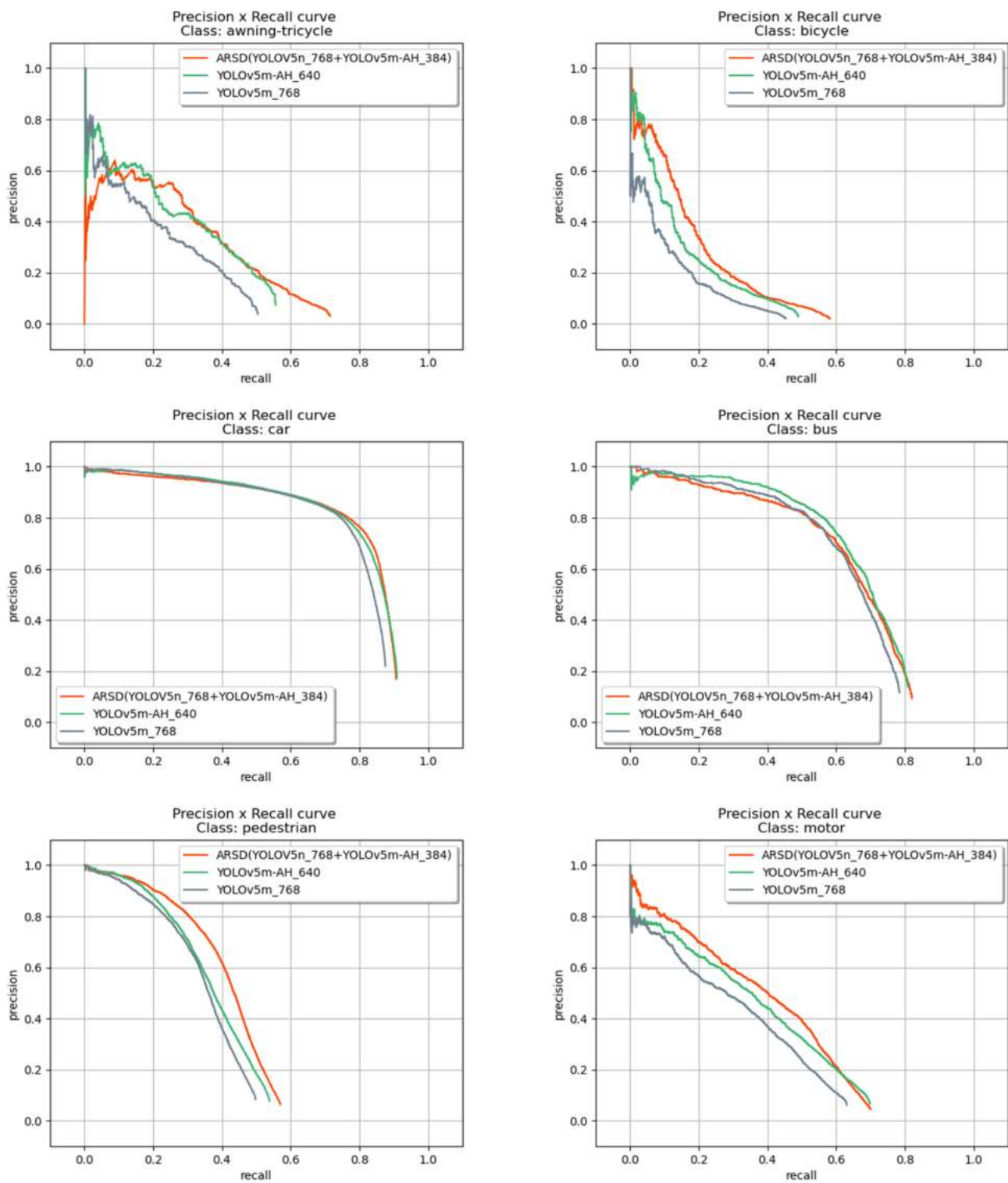
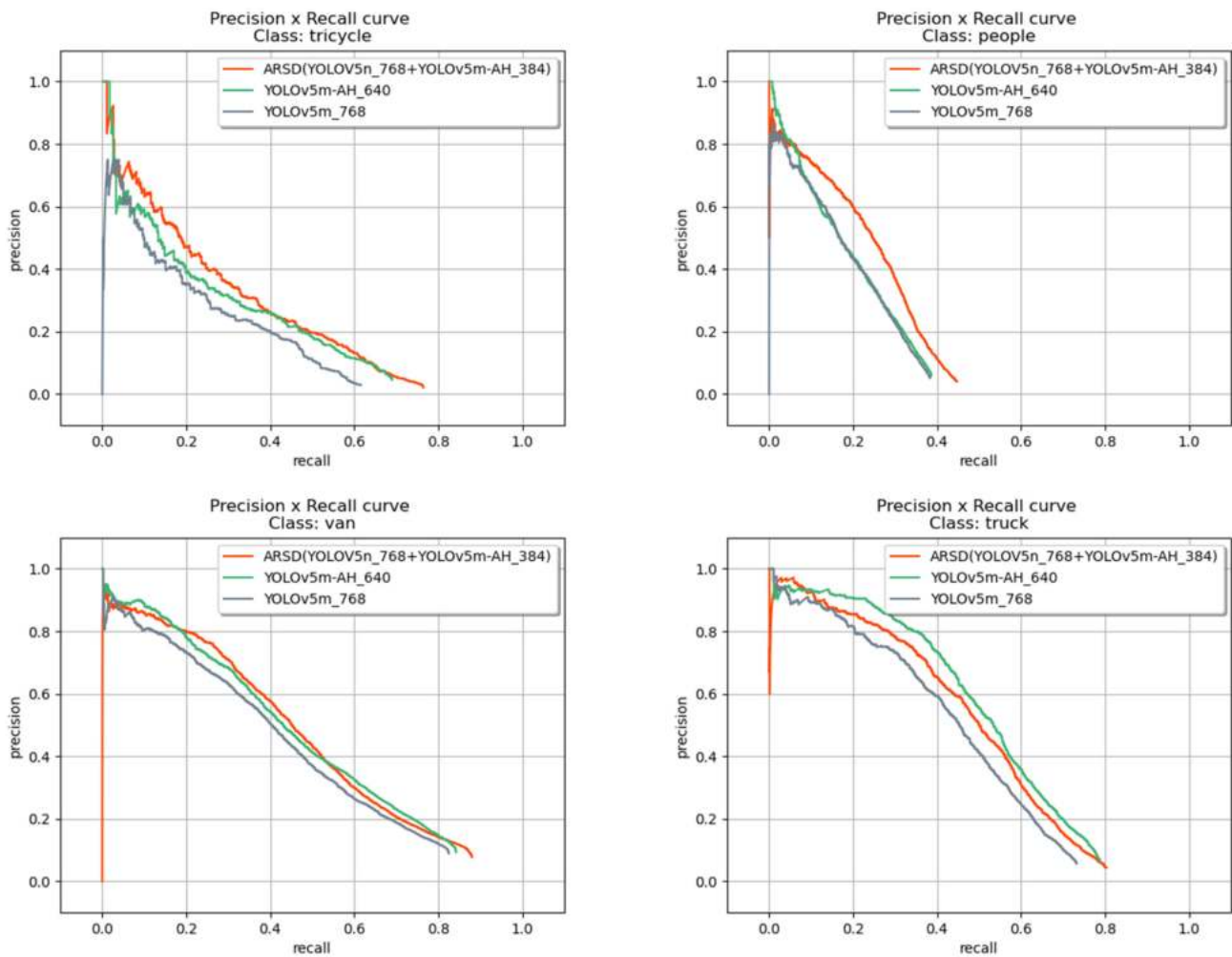


Figure 7. Cont.



**Figure 7.** The comparison of  $768 \times 768$  YOLOv5m,  $640 \times 640$  YOLOv5m-AH, and our method for each category.

**Table 3.** Results with different cluster methods, clusters, and remaining sub-regions. Note: (x) indicates the ratio of sub-regions being discarded.

	Methods	Cluster Methods	Number of Clusters	Candidate Sub-Regions	Remaining Sub-Regions	AP	AP50
1	K-means	K-means	3	4830	4830	22.64	40.6
2	DNSCAN	Mean-Shift	-	5559	5559	22.31	40.1
3	ARSD	FPDCA	2	3220	3220 (0)	20.59	37.76
4	ARSD	FPDCA	2	3220	2146 (1/3)	19.16	35.63
5	ARSD	FPDCA	2	3220	1610 (1/2)	18.33	33.4
6	ARSD	FPDCA	3	4830	4830 (0)	22.93	41.31
7	ARSD	FPDCA	3	4830	3220 (1/3)	21.6	39.6
8	ARSD	FPDCA	3	4830	2415 (1/2)	20.05	37.1
9	ARSD	FPDCA	4	6440	6440 (0)	23.9	41.92
10	ARSD	FPDCA	4	6440	4293 (1/3)	22.32	40.15
11	ARSD	FPDCA	4	6440	3220 (1/2)	21.12	37.86

#### 4. Discussion

The aim is to improve the object detection accuracy for UAV images in high resolution with a lot of small targets in the images. This paper proposes a two-stage object detection framework for UAV images, with extensive performance evaluations conducted. The qualitative results in Section 3.3 and quantitative evaluations in Section 3.4 directly show that the proposed framework performs better than the state-of-the-art detection methods. Compared with GLASN, it improves AP by 2.54% and AP50 by 2.1%. The ablation study

includes four experiments. The first experiment performs tests on different network scales and the results demonstrate that YOLOv5n is the most suitable for ORDN. The second experiment shows that the network with an additional head performs better than others within the same processing time. The third experiment proves that this two-stage framework is more accurate in small-scale categories such as Pedestrians, People, and Bicycle. The final experiment analyses the influence of clustering algorithms and the number of clusters. The novelty mentioned in Section 1 has been verified by the above four experiments. The framework can improve the accuracy of object detection within the same processing time, especially in small-scale objects.

This method should balance the number of clusters and the ratio of the remaining sub-regions among all candidate sub-regions. If a higher number of clusters is set, the calculation time will undoubtedly increase, but the accuracy will also be improved. Our future work includes the study of selecting these parameters more precisely.

## 5. Conclusions

This paper proposes ARSD, an adaptive region selection detection framework, that is more efficient and more effective for UAV image object detection. The main idea is to locate object-dense sub-regions in high-resolution UAV images and feed these sub-regions into the second-stage detector. Finally, the results are merged by WBF. In addition, we developed an adaptive region selection algorithm which consists of the Fixed Points Density-based Clustering Algorithm and the Adaptive Sub-regions Selection Algorithm. FPDCA can generate a fixed number of sub-regions combined with density information, while ASSA can score each sub-region objectively. Additionally, a new detection head is added based on the original detection heads for better small object detection. The evaluations on VisDrone2021-DET datasets have demonstrated the effectiveness and adaptiveness of ARSD. We will extend this work to apply the proposed method to small-scale object detection in remote sensing and search and rescue.

**Author Contributions:** Conceptualization, Y.W., Y.H. (Yan Huang) and Y.H. (Yi Han); methodology, Y.W. and Y.Z.; software, Y.W., Y.H. (Yan Huang) and Q.Y.; validation, Y.W. and Y.H. (Yan Huang); formal analysis, Y.W., Y.Z. and Y.H. (Yan Huang); investigation, Y.W., Y.C., Z.L., Z.Y. and Q.L.; resources, Y.W. and Y.H. (Yan Huang); data curation, Y.W., Y.H. (Yi Han) and Z.L.; writing—original draft preparation, Y.W. and Y.H. (Yi Han); writing—review and editing, Y.W., Y.H. (Yi Han), Y.C., Z.L., Z.Y. and Q.L.; visualization, Y.W. and Q.Y.; supervision, Y.H. (Yi Han) and Z.Y.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by a grant from the National Natural Science Foundation of China (Grant No. 61801341). This work was also supported by the Research Project of Wuhan University of Technology Chongqing Research Institute (No. YF2021-06).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The simulation data used to support the findings of this study are available from the corresponding author upon request. The video of the experimental work can be found at the following link: [https://drive.google.com/drive/folders/1eYFxnSaYYEhY\\_M0tdM55oybCA6sMMgzG?usp=sharing](https://drive.google.com/drive/folders/1eYFxnSaYYEhY_M0tdM55oybCA6sMMgzG?usp=sharing) (accessed on 25 August 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Hird, J.N.; Montaghi, A.; McDermid, G.J.; Kariyeva, J.; Moorman, B.J.; Nielsen, S.E.; McIntosh, A.C.S. Use of Unmanned Aerial Vehicles for Monitoring Recovery of Forest Vegetation on Petroleum Well Sites. *Remote Sens.* **2017**, *9*, 413. [CrossRef]
2. Shao, Z.; Li, C.; Li, D.; Altan, O.; Zhang, L.; Ding, L. An Accurate Matching Method for Projecting Vector Data Into Surveillance Video To Monitor And Protect Cultivated Land. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 448. [CrossRef]

3. Shen, Q.; Jiang, L.; Xiong, H. Person Tracking and Frontal Face Capture with UAV. In Proceedings of the IEEE 18th International Conference on Communication Technology (ICCT), Chongqing, China, 8–11 October 2018; pp. 1412–1416.
4. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond Rgb: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 20–32. [CrossRef]
5. Yuan, Z.; Jin, J.; Chen, J.; Sun, L.; Muntean, G.M. ComProSe: Shaping Future Public Safety Communities with ProSe-based UAVs. *IEEE Commun. Mag.* **2017**, *55*, 165–171. [CrossRef]
6. Munawar, H.S.; Ullah, F.; Heravi, A.; Thaheem, M.J.; Maqsoom, A. Inspecting Buildings Using Drones and Computer Vision: A Machine Learning Approach to Detect Cracks and Damages. *Drones* **2021**, *6*, 5. [CrossRef]
7. Kundid Vasić, M.; Papić, V. Improving the Model for Person Detection in Aerial Image Sequences Using the Displacement Vector: A Search and Rescue Scenario. *Drones* **2022**, *6*, 19. [CrossRef]
8. Reckling, W.; Mitasova, H.; Wegmann, K.; Kauffman, G.; Reid, R. Efficient Drone-Based Rare Plant Monitoring Using a Species Distribution Model and AI-Based Object Detection. *Drones* **2021**, *5*, 110. [CrossRef]
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Redmon, J.; Farhadi, A. Yolo9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
11. Redmon, J.; Farhadi, A. Yolov3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
12. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal Speed And Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
13. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
14. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft Coco: Common Objects in Context. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 740–755.
15. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Fei-Fei, L. Imagenet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
16. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
17. Zhu, P.; Wen, L.; Bian, X.; Ling, H.; Hu, Q. Vision Meets Drones: A Challenge. *arXiv* **2018**, arXiv:1804.07437.
18. Kalra, I.; Singh, M.; Nagpal, S.; Singh, R.; Vatsa, M.; Sujit, P.B. Dronesurf: Benchmark Dataset for Drone-Based Face Recognition. In Proceedings of the 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (Fg 2019), Lille, France, 14–18 May 2019; pp. 1–7.
19. Glenn, J. Yolov5 Release v6.1. 2022, 2, 7, 10. Available online: <https://github.com/ultralytics/yolov5/releases/tag/v6.1> (accessed on 25 August 2022).
20. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. Tph-Yolov5: Improved Yolov5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
21. Akyon, F.C.; Altinuc, S.O.; Temizel, A. Slicing Aided Hyper Inference And Fine-Tuning for Small Object Detection. *arXiv* **2022**, arXiv:2202.06934.
22. Zhang, J.; Huang, J.; Chen, X.; Zhang, D. How To Fully Exploit the Abilities of Aerial Image Detectors. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
23. Solovyev, R.; Wang, W.; Gabruseva, T. Weighted Boxes Fusion: Ensembling Boxes From Different Object Detection Models. *Image Vis. Comput.* **2021**, *107*, 104117. [CrossRef]
24. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
25. Hartigan, J.A.; Wong, M.A. Algorithm as 136: A k-Means Clustering Algorithm. *J. R. Stat. Society. Ser. C (Appl. Stat.)* **1979**, *28*, 100–108. [CrossRef]
26. Comaniciu, D.; Meer, P. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [CrossRef]
27. Wang, Y.; Yang, Y.; Zhao, X. Object Detection Using Clustering Algorithm Adaptive Searching Regions In Aerial Images. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 651–664.
28. Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L.; Da Silva, E.A.B. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics* **2021**, *10*, 279. [CrossRef]
29. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
30. Deng, S.; Li, S.; Xie, K.; Song, W.; Liao, X.; Hao, A.; Qin, H. A Global-Local Self-Adaptive Network for Drone-View Object Detection. *IEEE Trans. Image Process.* **2020**, *30*, 1556–1569. [CrossRef] [PubMed]
31. Zhang, P.; Zhong, Y.; Li, X. Slimyolov3: Narrower, Faster and Better for Real-Time UAV Applications. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.

32. Yang, F.; Fan, H.; Chu, P.; Blasch, E.; Ling, H. Clustered Object Detection in Aerial Images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 8311–8320.
33. Li, C.; Yang, T.; Zhu, S.; Chen, C.; Guan, S. Density Map Guided Object Detection in Aerial Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 190–191.
34. Liao, J.; Piao, Y.; Su, J.; Cai, G.; Huang, X.; Chen, L.; Huang, Z.; Wu, Y. Unsupervised Cluster Guided Object Detection in Aerial Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 11204–11216. [CrossRef]

## Article

# Using Classify-While-Scan (CWS) Technology to Enhance Unmanned Air Traffic Management (UTM)

Jiangkun Gong <sup>1</sup> , Deren Li <sup>1</sup>, Jun Yan <sup>1,\*</sup>, Huiping Hu <sup>2</sup> and Deyong Kong <sup>3</sup><sup>1</sup> State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China<sup>2</sup> Wuhan Geomatics Institute, Wuhan 430022, China<sup>3</sup> School of Information Engineering, Hubei University of Economics, Wuhan 430205, China

\* Correspondence: yanjun\_pla@whu.edu.cn; Tel.: +86-027-68778527

**Abstract:** Drone detection radar systems have been verified for supporting unmanned air traffic management (UTM). Here, we propose the concept of classify while scan (CWS) technology to improve the detection performance of drone detection radar systems and then to enhance UTM application. The CWS recognizes the radar data of each radar cell in the radar beam using advanced automatic target recognition (ATR) algorithm and then integrates the recognized results into the tracking unit to obtain the real-time situational awareness results of the whole surveillance area. Real X-band radar data collected in a coastal environment demonstrate significant advancement in a powerful situational awareness scenario in which birds were chasing a ship to feed on fish. CWS technology turns a drone detection radar into a sense-and-alert platform that revolutionizes UTM systems by reducing the Detection Response Time (DRT) in the detection unit.

**Keywords:** automatic target recognition (ATR); classify while scan (CWS); drone detection radar; detection response time (DRT); unmanned air traffic management (UTM)

**Citation:** Gong, J.; Li, D.; Yan, J.; Hu, H.; Kong, D. Using Classify-While-Scan (CWS) Technology to Enhance Unmanned Air Traffic Management (UTM). *Drones* **2022**, *6*, 224. <https://doi.org/10.3390/drones6090224>

Academic Editors: Daobo Wang and Zain Anwar Ali

Received: 17 July 2022

Accepted: 24 August 2022

Published: 27 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Within the past few years, the number of unmanned aircraft systems (UAS), also often referred to as drones, has increased rapidly. They play essential roles in various tasks, including both civil applications and military applications. Drones are often used for reconnaissance, communication, and even attack missions by military clients. In 2020, drones revealed a new war era using the drone and anti-drone war after they achieved brilliant results in the Nagorno-Karabakh conflict [1,2], such as STM Kargu, Bayraktar TB2, IAI Harop, Orbiter 1K, etc. In addition to the military applications, drones, especially small drones, are popular in many civil projects [3], including entertainment photography, sports recording, infrastructure monitoring, precision agriculture, package delivery, rescue operations, remote mapping, and more. It seems that the era of drones will be the near-term future.

The flush drone applications require urgent counter UAS systems (C-UAS). C-UAS are generally designed to eliminate drone targets, and this includes three steps: (1) detection, (2) tracking and identification, and (3) effector [4,5]. The detection unit generally uses radar systems to detect the possible threat, and then operators identify the threat using EO/IR systems; finally, they utilize the effector unit to defeat the threat. Unlike military applications, some civil applications require that a C-UAS solution replace the defeater unit with a management unit. The typical application is airspace management at airports. An effective unmanned aircraft traffic management (UTM) system has been investigated for managing drones flying at low altitudes around airports [3,6].

The concept of UTM systems originates from the current air traffic management (ATM) system supporting flight operations. Most of the detection workload is taken by ground-based radar systems, such as airport traffic control (ATC) radar [7,8]. The primary

identification work is supported by mandatory ACK systems such as the ADS-B system and the TCAS. One of the most challenging problems in designing and applying UTM is that there are neither associated EO/IR systems nor mandatory ADS-B sensors [9]. In this case, the radar system should take the task of identification. If a radar system can automatically detect radar echoes from drones and identify them from other clutters, it can considerably improve the performance of the UTM system. In other words, the drone detection radar needs to be equipped with automatic target recognition (ATR) module.

A radar ATR function is defined as recognizing targets mainly based on radar signals. Primitive target recognition was performed by using the audible representation of the received echoes, and then a trained operator deciphered the information in the sound. Later, radar engineers and scholars conducted design algorithms to do the work automatically. After decades, several schools have recognized radar features that succeed in ATR applications in some cases. Note that the traditional ATR solution contains two procedures, including feature extraction and pattern recognition. Here, we use the feature to classify the interpretations of ATR. The first one is high-range resolution profile (HRRP) technology [10]. Radar transmits ultrawide-band signals and obtains the target profiles in the range direction. The HRRP is a mapping of the shape of the target, and it is processed with a template matching approach in the dataset to obtain the target class. The second one is the micro-Doppler [11]. Micro-Doppler is thought to be the additional Doppler component produced by the micro motions on the target, such as the rotating movements of helicopters' blades, flapping birds' wings, and more. The embedded kinematic/structural information of the target can be measured from the micro-Doppler and then used for registering the target [11]. In addition, the third method could sometimes use tracking information such as speed and trajectory for identifying targets. Although machine learning is popular in recent years [9,12,13], it is a black-box algorithm to process the units of feature extraction and pattern recognition. Since machine learning in ATR applications is an approach over a feature, it is not discussed here. Generally, a machine learning method can process either HRRP data, micro-Doppler data, or tracking data to recognize the target from different background cluttered environments. In addition, all these radar data can be presented via radar images or signals.

Currently, the discussion of applying ATR to drone detection projects or other projects underestimates the value of an ATR algorithm. This paper proposes the higher value of ATR functions, which is the situational awareness (SA) ability using the patented classify while scan (CWS) technology. Section 2 presents the basic introduction of CWS technology and our radar platform. Section 3 describes and analyzes some experimental results, and the application is discussed in Section 4. Finally, our conclusion is presented in Section 5.

## 2. Materials and Methods

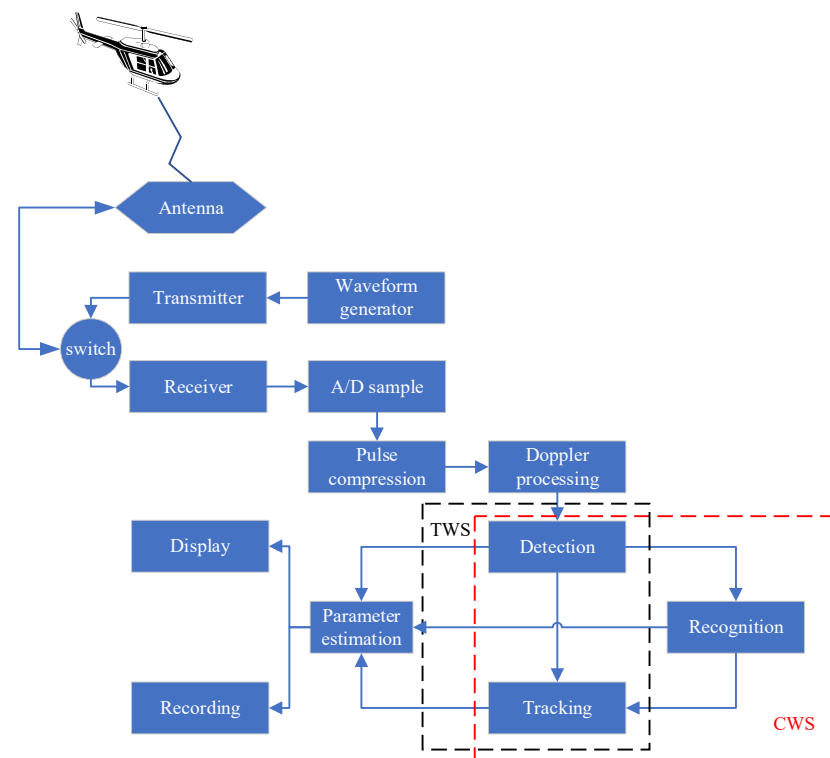
### 2.1. Design Principle

Generally, most ATR functions focus on the “point” information. The “point” characteristics are in both time and space dimensions. Spatially, they often recognize radar signals in a single radar resolution cell, and they also focus on current radar echoes over past and future radar signals temporally. They neither use tracking data of the target nor connect the radar signals in the radar resolution cells around the target. To the best of our knowledge, the track while scanning (TWS) could be probably the only technology that uses the time information of the target.

TWS is a mode of radar operation in which the radar scans the surveillance area while the acquired targets are tracked [14]. The significance of TWS is that the radar provides an overall view of the surveillance area and helps maintain better situation awareness (SA). Radar systems equipped with the TWS function sometimes also claim that they are 4D radar, in which time labels of the target are also regarded as the fourth dimension together with traditional 3D information (i.e., azimuth, elevation angle, and slant range). However, it seems that 4D radar is just a sales-promoting buzzword that has nothing to do with the

fourth dimension in physics. In our opinion, the real 4D radar should include the target attribute given by an ATR function.

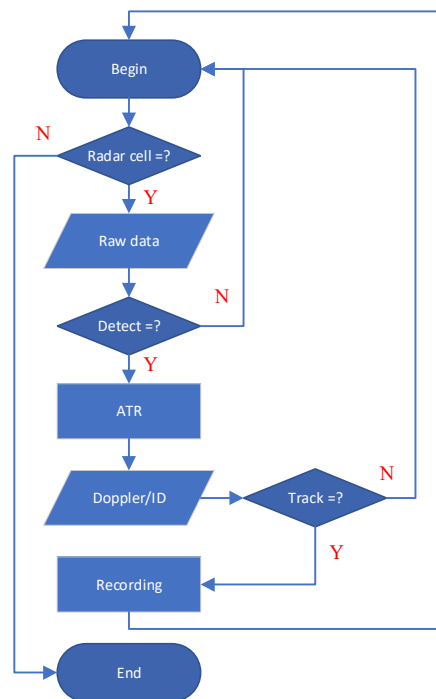
Integrating ATR and TWS is the technology of classify-while-scan (CWS). As shown in Figure 1, CWS processes the raw data in one radar resolution cell and obtains the object's ID. The ID of the target can be either used for recording, displaying on the radar screen, or assisting the tracking unit. With the ID of the target, the tracking unit could easily connect the same ID between the contiguous tracking data. This process can be called track-after-identify (TAI). Then, the CWS function processes radar data in every radar cell and outputs the targets along with the ranged cells in the radar beam. Consequently, the radar beam continues to scan the area and capture the traces of the active targets, and then the whole scenario is presented in a radar display following the scan of the radar beam. Figure 2 demonstrates the basic flowchart of the CWS function.



**Figure 1.** The block diagram of radar equipped with CWS and TWS technology.

The specific algorithms in CWS technology can differ with different methods. For example, the detection methods can include an algorithm based on the signal-to-noise ratio (SNR), the moving target indication (MTI), the algorithm extracting the signal-to-clutter ratio (SCR), and more. ATR algorithms can be diverse, such as HRRP, micro-Doppler, and time-frequency analysis. The only difference exists in the tracking algorithm. Traditional tracking algorithms mainly depend on the correlation of Doppler and trajectory. However, CWS provides ID labels of objects in the radar beam, and then tracking algorithms can use ID labels to improve the correlation process and enhance the tracking accuracy. As shown in Figure 2, our CWS processing algorithm contains several steps, including:

- (1) Assume there are  $N$  radar range cells in one radar beam;
- (2) Extract the radar signals of objects using our SCR detector [15,16];
- (3) Recognize the radar echoes of objects using radar signal signatures;
- (4) Repeat the above detection & recognition algorithms and obtain all the IDs in each radar range cell;
- (5) The radar tracking unit uses the recognized results (i.e., IDs of targets) to track targets.



**Figure 2.** The basic flowchart of CWS technology.

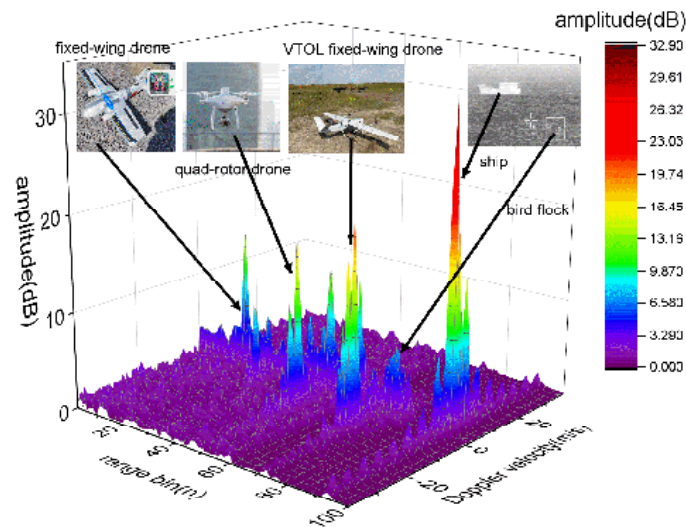
## 2.2. Test Conditions

Here, we demonstrate a drone detection radar system and the detection results using CWS technology. We applied the CWS technology to a new coastal surveillance radar. The pulse-Doppler radar works in the X-band. It is a narrow-band radar with a range resolution of approximately 12 m. The radar is equipped with an active, electronically scanned, phased-array antenna and deployed on a rotating table to achieve 360-degree coverage in the azimuth scan. Moreover, it uses digital beam-forming (DBF) technology to obtain multiple radar beams in the pitch direction every time. The flexibly configured rotating speed of the rotating table is between 2 s and 20 s. The detection response time (DRT) is approximately 30 ms, representing that the lag between an echo return, detection, and eventual display is only 30 ms. It is capable of tracking 1000 targets simultaneously. It can recognize different targets, including birds, drones, vehicles, ships, people, helicopters, and others, and then it presents detection results with graphic icons, which label the recognition results, along with the rotating motion of the scanning beam. The numbers around the icons are tracking numbers.

The test was conducted in a coastal area with a cluttered sea environment. The test area is located on the Yellow Sea coast of Qidong, China. The radar is set on the roof of a 12-m tall building, and the sea was scanned horizontally. The initial goal of this project was to develop a prototype drone detection radar. During the project, an infrared sensor and an optical camera were deployed to support the project and used to confirm the recognition results of the ATR function. This project continued for several months in 2020, and some data were extracted in this paper. The sea scale during the test was about Degree 3~5. The height of the wave was 0.5–1.25 m with Degree 3 and 2.50–4.00 m with Degree 5.

We collected radar signals of different types of drones, including a quad-rotor drone, a fixed-wing drone, and a hybrid Vertical Take-off and Landing (VTOL) fixed-wing drone. They were cooperative targets. Some of their parameters are listed in Table 1. Albatross 1 is a homemade fixed-wing drone with only one pusher blade, DJI Phantom 4 is a famous quad-rotor drone with four lifting blades, and TX25A is a large hybrid VTOL fixed-wing drone with one pusher blade and four lifting blades. Photos of these drones are shown in Figure 3. If we refer to NATO's category of drones [4], Albatross 1 is a Microdrone, DJI

Phantom 4 is a Mini-drone, and TX25A is a Small-drone. In addition, some local fishing ships and birds were also our test targets, and we also collected their radar data.



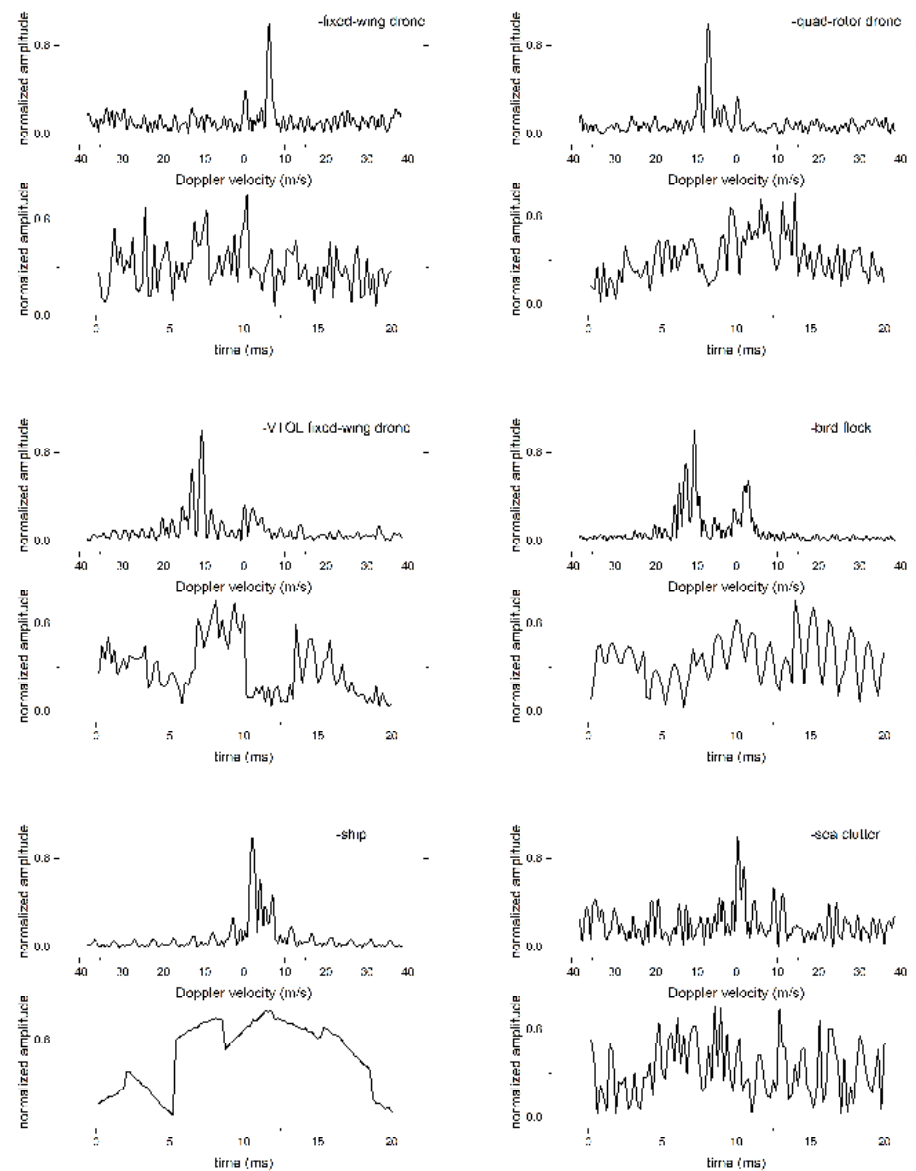
**Figure 3.** The range-Doppler image of radar data in one radar beam containing different objects.

**Table 1.** Parameters of different drones.

Drone Type	Hybrid VTOL Fixed-Wing	Fixed-Wing	Multi-Rotor
Model	TX25A	Albatross 1	Phantom 4
Manufacturer	HarriSkydream Inc.	Homemade	DJI Inc.
Flight weight	26 kg	0.3 kg	1.38 kg
Body size	197 cm	80 cm	40 cm
Wing span	360 cm	108 cm	40 cm
Cruise speed	25 m/s	10 m/s	15 m/s
Rotor number	5	2	4
Blade length	30 cm	10 cm	20 cm
Aero-frame materials	FRP (Fibre reinforces plastic)	EPP (Expanded polypropylene)	PC (Polycarbonate)

### 3. Results

It required several steps from an echo return into the eventual display on the radar display. Figures 3–5 show some examples of the whole procedure, and Figure 6 demonstrates a real radar screenshot using CWS technology. Figure 3 shows the range-Doppler images containing several specific objects in one simulated radar beam. Each radar range sector had twenty bins, where the target was in the center of the range sector. The range resolution of a radar bin is 12 m, and the range of a range sector is 240 m. To demonstrate the detection results, we manually jointed five sectors from different radar beams into the detection result in Figure 3. Most clutters were approximately 0 Hz. Since the detection ranges of those targets were different, the amplitudes of the signals were incomparable. Nevertheless, the Doppler signals in each spectrum still differ from each other. We use the SCR detector to extract radar signals of targets in cluttered backgrounds [15]. The SCR detector performs superior to the SNR detector, reducing missed and false alarms when detecting and tracking drones.



**Figure 4.** Raw time & frequency data of different objects.

The time-frequency characteristics of different objects provide useful signatures for the ATR module. Figure 4 abstracts the raw radar of objects in Figure 5 and plots the time-frequency distributions of each object. The spectra were obtained using conventional Fourier transforms. We normalized the signal amplitudes to remove the interferences posed by the detection ranges. The radar dwell time in the time domain was 20 ms. There were no patterns in the time domains. In addition to the body Doppler, the micro-Doppler also appeared in the spectra, and they seem to have patterns. Then, a patented ATR algorithm is used to analyze the time-frequency characteristics of radar signals from different objects and obtain the recognized results of the targets.

The radar tracking units used the recognized results in each radar beam to enhance the tracking data of every target. Figure 5 shows the tracking data of our targets in Figures 3 and 4. These data were real data collected on different test dates. The height information was calculated using DBF technology; therefore, the accuracy was limited. Nevertheless, the trace of each target still described the moving kinematics of each target. Generally, the mean Doppler velocity values of the fixed-wing drone, quad-rotor drone, VTOL fixed-wing drone, bird flock, and ship were approximately 7.34 m/s, 6.76 m/s, 15.22 m/s, 12.75 m/s, and 3.36 m/s, respectively. The ship has the slowest moving speed,

while the VTOL drone has the fastest moving velocity. In addition, the measured height numbers of those targets were approximately 670.07 m, 272.68 m, 100.05 m, 20.81 m, and 0 m, respectively. These measured characteristics were following the natural flight parameters of these objects.

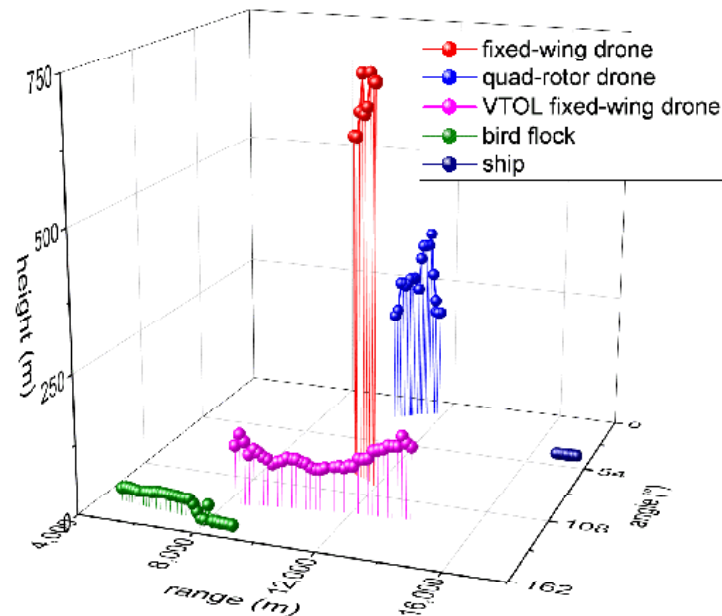


Figure 5. The tracking data of different moving targets.

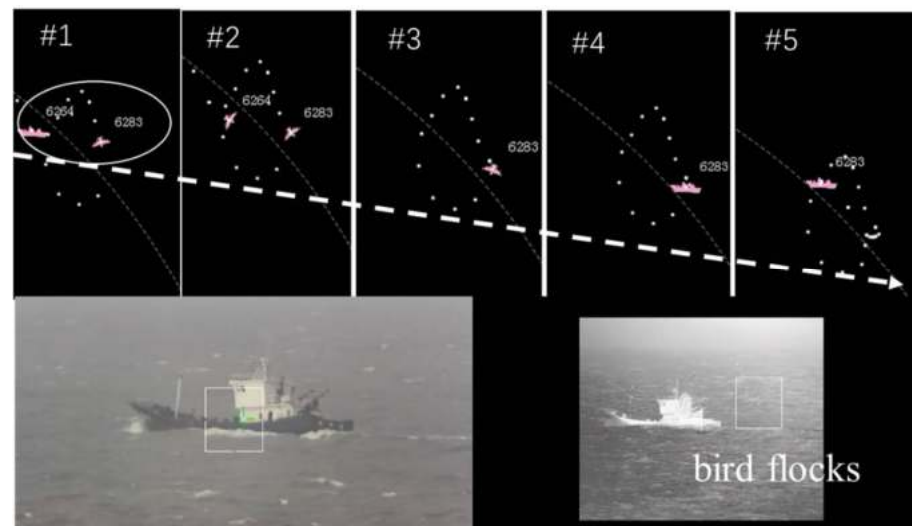


Figure 6. A real scenario sensed by situation awareness using the CWS technology, where the birds were chasing the ship to feed on fish.

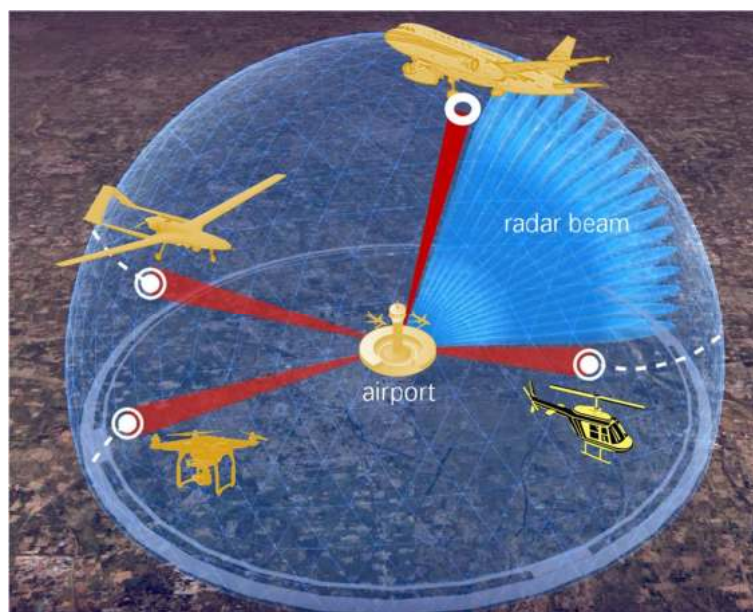
#### 4. Discussion

The most significant benefit provided by CWS technology is situational awareness (SA). Figure 6 provides a typical example of an application of situational awareness where birds chased ships to pick up fish awakened by the ship's propellers. The recognized results were marked using graphic icons, including ships, helicopters, drones, and birds. The thin white dotted lines are the tracking traces of objects. The numbers around the icons were the tracking numbers of objects. CWS enhanced the radar to become a WYSIWYG (What You See Is What You Get) system or a real-time sense-and-alert system. Moreover, the whole scenario in the surveillance area was captured and updated with new data. Note

the tracking birds of No. 6264 were flying around the ship of No. 6283; thus, radar echoes from birds were interfering with the detection of the ship. When they appeared in the same radar cell, the technology recognized that these data were sometimes birds while the others were the ship. The birds were flying around the ship because they habitually chase ships in the sea to feed on the fish, which were awakened by the rotating propellers of the ship. In other words, SA provided by CWS described a habit of sea birds.

UTM is a safety system that helps ensure the newest entrant, a variety of drones, into the skies does not collide with buildings, larger aircraft, or one another. UTM is different from the ATM functioned by the airport agencies in some respects. First, UTM cannot use the airport traffic control (ATC) radar systems used by the current ATM system. ATC radar systems are primarily designed to detect and track large, fast-moving aircraft flying in high airspace. Compared to large aircraft, drones generally have small RCS values and slow-moving speeds. In particular, the ATC radar systems were reported to turn to identify and retain targets that move consistently, remain visible from sweep to sweep, and have a ground speed of at least 15.432 m/s, and, as a result, they failed to detect the flock of Canadian geese in the aircraft accident of flight 1549 on 15 January 2009 [17]. The flight characteristics of drones make them potentially undetectable by traditional ATC radar systems. Second, drones may not be equipped with signal transmission sensors, such as RID broadcast receivers and ADS-B. This means that ground-based detection systems could be critical sensors to support the UTM system.

Similar to the ATC in the ATM system, a drone detection radar system is the key role of UTM. Nevertheless, not all drone detection radar systems are suitable. Here, we insist that a 4D radar must detect an object and obtain its 3D position and 1D attribution, provided by an ATR algorithm. Figure 7 demonstrates a typical application, where 4D radar can enhance the real-time situational awareness at a simulated airport. In this way, the drone detection radar can detect and track different types of drones (e.g., quad-rotor drone, VTOL drone, unmanned-helicopter, etc.) over other large aircraft or bird clutters and then support a UTM system. ATR turns the radar data of objects into meaningful IDs of targets. These knowledge-based IDs could also be used for other applications, such as human-machine distributed situational awareness [9]. SA may be a key in UTM applications.



**Figure 7.** Using CWS to enhance UTM at an airport.

Generally, the performance of an ATR algorithm can be described in four tiers:

(1) Tier DETECTION: extracting radar signals of a target in background clutter, such as whether the radar echoes are from a target or a clutter;

(2) Tier CLASSIFICATION: classifying the class of the target, such as whether the radar echoes are from a drone or a bird;

(3) Tier IDENTIFICATION: identifying the type of the target in one class, such as whether the radar echoes are from a fixed-wing drone or a multi-rotor drone;

(4) Tier DESCRIPTION: describing the model of the target, such as whether the radar echoes are from a DJI Phantom 4 drone or a DJI S800 drone.

There are many types of aircraft at airports, so the ATR function should achieve the Tier IDENTIFICATION to support the UTM application (e.g., the example in Figure 7). As a higher tier of ATR function, CWS technology provides valuable situational awareness to support the UTM system. The performance of CWS technology must consider three factors, including the detection range, processing speed, and confidence factor. First, no matter what the algorithm of a CWS function is, it should be independent of the detection range of a target. Otherwise, the drone detection radar could not cover enough airspace at airports. Second, the processing speed determines the DRT. There is always a lag between an echo return and eventual display on the radar screen. The lag is sometimes called the DRT, which contains the time spent on communication and processing. Ideally, DRT should be at a level of milliseconds to fulfill a WYSIWYG system. Practically, the DRT should be shorter than the update interval between the current radar beam and the next radar beam. Last, but not least, the confidence factor determines the recognized result of CWS technology. Unlike the identification probability, the confidence factor describes the trust level of the recognized results calculated by the ATR algorithm of the CWS technology. Besides, the trajectory prediction is a key module in the ATM/UTM system [18–23], which can use the predicted trajectory to reduce the DRT in the tracking unit. In contrast, our CWS technology shortens the DRT in the detection unit. If the CWS technology is deployed in the radar system, it can provide the location of a target for the tracking unit using trajectory prediction algorithms, faster. In total, the shorter DRT and a higher confidence factor of a good CWS technology can result in a better UTM system.

## 5. Conclusions

There is a need for drone detection radar systems to support UTM systems at the airport. Here, we propose CWS technology to improve the detection performance of drone detection radar systems and enhance UTM systems. The CWS processes radar data of each radar cell in the radar beam using the ATR algorithm and then obtains all targets' recognized results. It reduces the DRT in the detection unit of the radar. Moreover, the recognized results can be used to track targets using the TAI algorithm in a surveillance area and then obtain real-time situational awareness (SA) results. The future work will investigate the performance of the TAI algorithm, and the performance of the radar situational awareness (SA) with the CWS and TAI algorithms. With situational awareness, drone detection radar has become a WYSIWYG, bringing revolutionary performance to UTM systems.

**Author Contributions:** Conceptualization, J.Y.; methodology, J.G.; software, D.K.; validation, J.Y.; formal analysis, J.G.; investigation, J.Y.; resources, D.L.; data curation, J.Y.; writing—original draft preparation, J.G.; writing—review and editing, H.H.; visualization, H.H.; supervision, J.Y.; project administration, D.L.; funding acquisition, D.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received some support by the Natural Science Foundation of Hubei Province (General Program: 2021CFB309).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Some of the data presented in this study may be available on request from the corresponding author. The data are not publicly available due to the internal restriction of the research group.

**Acknowledgments:** We appreciate both the testers during the collection of the data, and we also want to thank the authors whose photographs are reproduced in this study.


**Conflicts of Interest:** The authors declare that they have no conflict of interest.

## References

1. Semercioglu, H. The New Balance of Power in the Southern Caucasus in the Context of the Nagorno-Karabakh Conflict in 2020. *Res. Stud. Anatolia J.* **2021**, *4*, 49–60.
2. Hecht, E. Drones in the Nagorno-Karabakh War: Analyzing the Data. *Mil. Strateg. Mag.* **2022**, *7*, 31–37.
3. Jiang, T.; Geller, J.; Ni, D.; Collura, J. Unmanned Aircraft System Traffic Management: Concept of Operation and System Architecture. *Int. J. Transp. Sci. Technol.* **2016**, *5*, 123–135. [CrossRef]
4. Wellig, P.; Speirs, P.; Schuepbach, C.; Oechslin, R.; Renker, M.; Boeniger, U.; Pratisto, H. Radar Systems and Challenges for C-UAV. In Proceedings of the 2018 19th International Radar Symposium (IRS), Bonn, Germany, 20–22 June 2018; pp. 1–8.
5. Watts, A.C.; Ambrosia, V.G.; Hinkley, E.A. Unmanned Aircraft Systems in Remote Sensing and Scientific Research: Classification and Considerations of Use. *Remote Sens.* **2012**, *4*, 1671–1692. [CrossRef]
6. Chin, C.; Gopalakrishnan, K.; Egorov, M.; Evans, A.; Balakrishnan, H. Efficiency and Fairness in Unmanned Air Traffic Flow Management. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 5939–5951. [CrossRef]
7. Nakamura, H.; Matsumoto, Y.; Suzuki, S. Flight Demonstration for Information Sharing to Avoid Collisions between Small Unmanned Aerial Systems (Suass) and Manned Helicopters. *Trans. Jpn. Soc. Aeronaut. Space Sci.* **2019**, *62*, 75–85. [CrossRef]
8. Taylor, J.W.; Brunins, G. Design of a New Airport Surveillance Radar (ASR-9). *Proc. IEEE* **1985**, *73*, 284–289. [CrossRef]
9. Pérez-Castán, J.A.; Pérez-Sanz, L.; Bowen-Varela, J.; Serrano-Mira, L.; Radisic, T.; Feuerle, T. Machine Learning Classification Techniques Applied to Static Air Traffic Conflict Detection. *IOP Conf. Ser. Mater. Sci. Eng.* **2022**, *1226*, 12019. [CrossRef]
10. Tait, P. *Introduction to Radar Target Recognition*; Institution of Electrical Engineers: London, UK, 2006; ISBN 9781849190831.
11. Chen, V.C. *The Micro-Doppler Effect in Radar*; Artech House: Norwood, MA, USA, 2011; ISBN 9781608070572/1608070573.
12. Raval, D.; Hunter, E.; Hudson, S.; Damini, A.; Balaji, B. Convolutional Neural Networks for Classification of Drones Using Radars. *Drones* **2021**, *5*, 149. [CrossRef]
13. Barbaresco, F.; Brooks, D.; Adnet, C. Machine and Deep Learning for Drone Radar Recognition by Micro-Doppler and Kinematic Criteria. In Proceedings of the 2020 IEEE Radar Conference (RadarConf20), Florence, Italy, 21–25 September 2020; pp. 1–6.
14. Bair, G.L.; Zink, E.D. Radar Track-While-Scan Methodologies. In Proceedings of the IEEE Region 5 Conference, 1988: ‘Spanning the Peaks of Electrotechnology’, Colorado Springs, CO, USA, 21–23 March 1988; pp. 32–37.
15. Gong, J.; Yan, J.; Li, D. The Radar Detection Method Based on Detecting Signal to Clutter Ratio (SCR) in the Spectrum. In Proceedings of the 2019 Photonics & Electromagnetics Research Symposium - Spring (PIERS-Spring), Rome, Italy, 17–20 June 2019; pp. 1876–1882.
16. Gong, J.; Li, D.; Yan, J.; Hu, H.; Kong, D. Comparison of Radar Signatures from a Hybrid VTOL Fixed-Wing Drone and Quad-Rotor Drone. *Drones* **2022**, *6*, 110. [CrossRef]
17. National Transportation Safety Board. *Aircraft Accident Report: Loss of Thrust in Both Engines After Encountering a Flock of Birds and Subsequent Ditching on the Hudson River*; National Transportation Safety Board: Washington, DC, USA, 2010.
18. Wang, X.; Musicki, D.; Ellem, R.; Fletcher, F. Efficient and Enhanced Multi-Target Tracking with Doppler Measurements. *IEEE Trans. Aerosp. Electron. Syst.* **2009**, *45*, 1400–1417. [CrossRef]
19. Pang, Y.; Zhao, X.; Hu, J.; Yan, H.; Liu, Y. Bayesian Spatio-Temporal Graph Transformer Network (B-STAR) for Multi-Aircraft Trajectory Prediction. *Knowledge-Based Syst.* **2022**, *249*, 108998. [CrossRef]
20. Pang, Y.; Zhao, X.; Yan, H.; Liu, Y. Data-Driven Trajectory Prediction with Weather Uncertainties: A Bayesian Deep Learning Approach. *Transp. Res. Part C Emerg. Technol.* **2021**, *130*, 103326. [CrossRef]
21. Xu, Z.; Zeng, W.; Chu, X.; Cao, P. Multi-Aircraft Trajectory Collaborative Prediction Based on Social Long Short-Term Memory Network. *Aerospace* **2021**, *8*, 115. [CrossRef]
22. Zeng, W.; Chu, X.; Xu, Z.; Liu, Y.; Quan, Z. Aircraft 4D Trajectory Prediction in Civil Aviation: A Review. *Aerospace* **2022**, *9*, 91. [CrossRef]
23. Pang, B.; Zhao, T.; Xie, X.; Wu, Y.N. Trajectory Prediction with Latent Belief Energy-Based Model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 11814–11824.

## Article

# A Modified YOLOv4 Deep Learning Network for Vision-Based UAV Recognition

Farzaneh Dadrass Javan <sup>1,2,\*</sup> , Farhad Samadzadegan <sup>2</sup>, Mehrnaz Gholamshahi <sup>3</sup> and Farnaz Ashatari Mahini <sup>2</sup>

<sup>1</sup> Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, 7522 NB Enschede, The Netherlands

<sup>2</sup> School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 1439957131, Iran; samadz@ut.ac.ir (F.S.); f.ashtari@ut.ac.ir (F.A.M.)

<sup>3</sup> Department of Electrical and Computer Engineering, Faculty of Engineering, Kharazmi University, Tehran 1571914911, Iran; mehrnazgholamshahi.khu@gmail.com

\* Correspondence: f.dadrassjavan@utwente.nl

**Abstract:** The use of drones in various applications has now increased, and their popularity among the general public has increased. As a result, the possibility of their misuse and their unauthorized intrusion into important places such as airports and power plants are increasing, threatening public safety. For this reason, accurate and rapid recognition of their types is very important to prevent their misuse and the security problems caused by unauthorized access to them. Performing this operation in visible images is always associated with challenges, such as the small size of the drone, confusion with birds, the presence of hidden areas, and crowded backgrounds. In this paper, a novel and accurate technique with a change in the YOLOv4 network is presented to recognize four types of drones (multirotors, fixed-wing, helicopters, and VTOLs) and to distinguish them from birds using a set of 26,000 visible images. In this network, more precise and detailed semantic features were extracted by changing the number of convolutional layers. The performance of the basic YOLOv4 network was also evaluated on the same dataset, and the proposed model performed better than the basic network in solving the challenges. Compared to the basic YOLOv4 network, the proposed model provides better performance in solving challenges. Additionally, it can perform automated vision-based recognition with a loss of 0.58 in the training phase and 83% F1-score, 83% accuracy, 83% mean Average Precision (mAP), and 84% Intersection over Union (IoU) in the testing phase. These results represent a slight improvement of 4% in these evaluation criteria over the YOLOv4 basic model.

**Citation:** Dadrass Javan, F.; Samadzadegan, F.; Gholamshahi, M.; Ashatari Mahini, F. A Modified YOLOv4 Deep Learning Network for Vision-Based UAV Recognition. *Drones* **2022**, *6*, 160. <https://doi.org/10.3390/drones6070160>

Academic Editors: Daobo Wang and Zain Anwar Ali

Received: 3 June 2022

Accepted: 24 June 2022

Published: 27 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** convolutional neural network CNN; YOLO deep learning; drone; UAV; drone detection; drone recognition

## 1. Introduction

Drones are actively used in a variety of fields, including recreational, commercial, security, crisis management, and mapping [1,2]. They are also used in combination with other platforms such as satellites in resource management, agriculture, and environmental protection [3–5]. However, the negligent and the malicious use of these flying vehicles poses a great threat to public safety in sensitive areas such as government buildings, power plants, and refineries [6,7]. For this reason, it is important to recognize drones to prevent them from entering critical infrastructure or ensuring security in large locations such as stadiums [8].

In this study, the recognition of four types of drones was investigated. Conventional drone detection technologies include the use of various sensors such as radar (radio detection and ranging) [9], Lidar (Light Detection and Ranging) [10], acoustic [11], and thermal sensors [12]. In these methods, first, the presence or the absence of the drone in the scene is checked and then the drone type recognition process is performed [13,14]. However, the

application of these types of sensors has always been associated with problems such as higher costs and higher energy consumption [15]. In contrast, visible images do not have these problems and are widely used for object recognition and semantic segmentation due to their high resolution [16]. On the other hand, the use of visible images also introduces problems such as light changes within the imagery, the presence of occluded areas, and a crowded background, which necessitates the application of an efficient and comprehensive method for recognition.

Recent advances in deep convolutional neural networks and the appearance of more improved hardware make it possible to use visual information to recognize objects with higher accuracy and speed [17]. Unlike conventional drone detection technologies, the nature of deep learning networks is to perform drone recognition simultaneously. By classifying inputs into several classes, these networks determine the presence, absence, image location, and type of drone class [18]. Among neural networks, the convolutional neural network (CNN) is one of the most important representatives of image recognition and classification. In this network, the input data enters the convolutional layers. The convolution operation is then performed using the network kernel to find similarities. Finally, feature extraction is performed using the resulting feature map [19]. There are different types of convolutional neural networks available such as R-CNN (Region-based CNN) [20], SPPNet (Spatial Pyramid Pooling Network) [21], and Faster-RCNN [22]. In these networks, due to the application of convolutional operations, more features are extracted than in conventional object detection methods and better speed and accuracy are achieved in recognizing objects. The extracted features are essentially descriptors of objects, and as the number of these features increases, object recognition is performed with higher accuracy. In these networks, the proposed regions are first defined using region proposal networks (RPNs) [23]. Then, convolutional filters are applied to these regions, and the extracted features are obtained as the result of the convolutional operation [22]. In other deep learning methods such as SSD (Single Shot MultiBox Detector) [24] and YOLO [25], the image is generally explored, which results in higher accuracy and speed in object recognition as compared to the basic methods [25]. The reason for the higher speed in these methods is the architecture is simpler than in region-based methods. The YOLO network is a method for detecting and for recognizing an object based on CNNs. The YOLO network predicts bounding box coordinates and class probabilities for these boxes, considering the whole image. The fourth edition of the YOLO Network is the YOLOv4 Deep Learning Network, which performs better than previous versions in terms of speed and accuracy [26]. However, the YOLOv4 deep learning network may not be able to overcome some challenges, such as the small size of the drone in different images [16]. In this study, this network could not recognize the drone in some of the challenging images. These challenges include confusing some drones with birds due to their small size, and the presence of drones in crowded backgrounds and hidden areas. Therefore, the YOLOv4 deep learning network was modified to better overcome the challenges of recognizing flying drones. The change in the architecture of this network is the main innovation in this article. Also, 4 types of multirotors, fixed-wings, helicopters, and VTOLs (Vertical Take-Off and Landing) were recognized. Given the need to recognize each type of UAV in different applications, the study of this topic can be considered as another innovation of this paper.

### *1.1. Challenges in Drone Recognition*

Drone recognition is always fraught with challenges. Some of the important challenges in this regard are discussed.

#### *1.1.1. Confusion of Drones and Birds*

Due to the physical characteristics of drones, they can easily be confused with birds in human eyes. This problem is more challenging when using drones in maritime areas due to the presence of more birds. The similarity between drones and birds and their distinction from each other is shown in Figure 1.



**Figure 1.** Challenges related to confusion with birds in drone recognition.

### 1.1.2. Crowded Background

As it appears from Figure 2, the presence of drones in areas with crowded backgrounds and similar environments has made them more difficult to recognize due to the inability to isolate the background. A crowded background refers to conditions such as the existence of clouds, dust, fog, and fire in the sky.



**Figure 2.** Challenges related to a crowded background in drone recognition.

### 1.1.3. Small Drone Size

The small size of drones makes them difficult to see at longer distances and difficult to quickly and accurately recognize, or they are possibly recognized as birds. Furthermore, the presence of a swarm of UAVs at different scales makes the recognition process more challenging. Figure 3 illustrates some examples of the presence of small drones at different scales.



**Figure 3.** Challenges related to the presence of small drones at different scales.

Drone recognition is always fraught with challenges. For this reason, it is necessary to use a fast, accurate, robust, and efficient method to overcome the challenges and to correctly recognize drones.

## 2. State of the Art Work

Recently, the use of drones has become increasingly popular, and they have been applied to various scientific and commercial purposes in different fields of photogrammetry, surveying, agriculture, natural disaster management, and so on [27]. There are different types of drones, each used for a specific purpose; and, in terms of design technology, application, and physical characteristics, they can be divided into four types: multirotor, helicopter, VTOL, and fixed-wing [28–31]. They are also divided into two scenarios in terms of operation manner in the environment. In the first scenario, drones operate individually, while in the second scenario they fly in combination with others, which are normally known as a swarm of UAVs [32–35].

Because of the enormous potential applications of each type of UAV in meeting the needs of society, the possibility of their misuse has become a major concern for communities. Over the past decade, much of the research has focused on finding efficient and accurate

techniques for the recognition of different types of UAVs [12,17,36]. However, sometimes drone recognition is difficult because they are normally flying in challenging environments. Therefore, the recognition of UAVs requires advanced techniques that can recognize them as they fly individually or in swarm mode.

### 3. Related Works

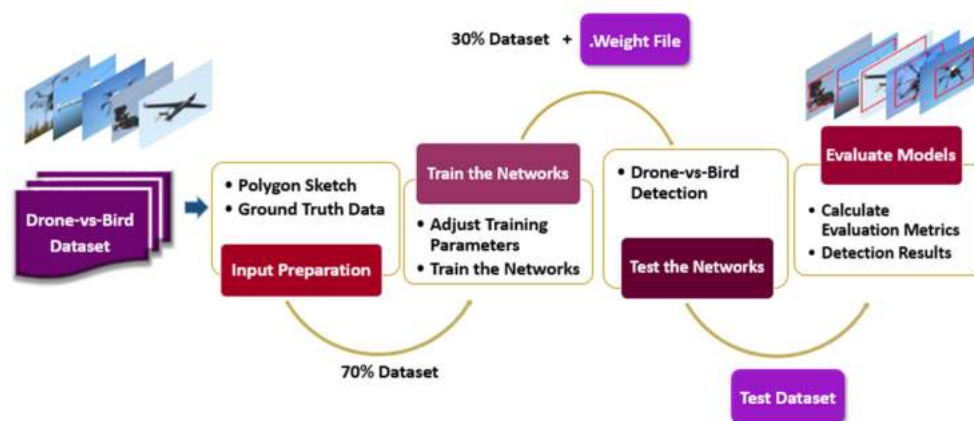
Due to the increasing development of deep neural networks in visual applications, these networks are also used widely for the recognition of objects in visible images [36–38]. In 2019, Nalamati et al. used a collection of visible images to detect small drones and solve their detection challenges. In this work, different CNN-based architectures were used, such as SSD [22], Faster-RCNN with ResNet-101 [22], and Faster-RCNN [22] with Inceptionv2. Based on the results, the R-CNN network with ResNet-101 performs the best in training and testing [39]. In 2019, Unlu et al. used an independent drone detection system, using the YOLOv3 Deep Learning Network. One of the advantages of this system is its cost-effectiveness due to the limited need for GPU memory. This study can detect drones of a small size and at a minimal distance, but it cannot recognize the types of drones [40]. In 2020, Mahdavi et al. detected a drone using a fisheye camera, and three methods of classification were applied: convolutional neural network (CNN), support vector machine (SVM), and nearest-neighbor. The results showed that CNN, SVM, and nearest-neighbor have total accuracy of 95%, 88%, and 80%, respectively. Compared with other classifiers with the same experimental conditions, the accuracy of the convolutional neural network classifier was satisfactory. In this study, only the detection of drones without considering their types and challenges has been investigated [41]. In 2020, Behera et al. detected and classified drones in RGB images using the YOLOv3 network, and they achieved a mAP of 74% after 150 epochs. In this article, only drones were detected at various distances, and the issue of drone recognition and its distinction from birds was not discussed [42]. In 2020 Shi et al. proposed a detection process of the low-altitude drone based on the YOLOv4 deep learning network. They then compared the YOLOv4 detection result with the YOLOv3 and the SSD networks. In this study, the YOLOv4 network performed better than the YOLOv3 and the SSD networks in detecting, recognizing, and identifying three types of drones in terms of mAP and detection speed, achieving 89% mAP [43]. In 2021, Tan Wei Xun et al. detected and tracked a drone using the YOLOv3 deep learning network. In their study, the NVIDIA Jetson TX2 was used to detect drones in real-time. The results of this method show that the proposed YOLOv3 network detects drones of three sizes: small, medium, and large, with an average confidence score of 88% and a confidence score between 60% and 100% [44]. In 2021, Isaac-Medina et al. detected and tracked drones using a set of visible and thermal images and four deep learning network architectures. In this paper, the deep learning networks Faster RCNN, SSD, YOLOv3, and DETR (DEtection TRansformer) are used. Based on the results, all the studied networks were able to detect a small drone at a far distance. But the YOLOv3 deep learning network generally leads to better accuracy (up to 0.986 mAP) and the RCNN network performed better in detecting small drones (up to 0.77 mAP) [45]. In 2021 Singha et al. developed an automatic drone detection system using YOLOv4. They used a dataset of drones and birds to detect drones, and then evaluated the model on two types of drone videos. The results obtained in this study for detecting two types of multirotor drones are: mAP 74.36%, F1-score 0.79, recall 0.68, and precision 0.95 [46]. In 2021, Liu et al. examined three object detection methods, such as YOLOv3, YOLOv4, RetinaNet, and FCOS (Fully Convolutional One-stage Object Detector) networks, on visible image data. To get great accuracy in drone detection, the pruned YOLOv4 model is used to build a sparser, flatter network. The application of the method has improved the detection of small drones and high-speed drones. The pruned YOLOv4, with a pruning rate of 0.8 and a 24-layer pruning, achieved a mAP of 90.5%, an accuracy of 22.8%, a recall of 12.7%, and a processing speed of 60%. However, the challenges of crowded backgrounds, hidden areas, and surveys of multiple drone types have not yet been addressed [16]. In 2022, Samadzadegan et al. detected and recognized drones using YOLOv4 Deep Networks

in visible images [47]. This network can recognize multirotor and helicopters directly, and it can differentiate between drones and birds with a mAP of 84%, an IoU of 81%, and an accuracy of 83%. In this paper, the challenges related to recognition have been well addressed, but this method is limited to detecting and to recognizing only two drone types, such as multirotor and helicopter, and it has not detected other types [47].

In this study, to achieve higher accuracy in solving the challenges of drone type recognition in visible images, a modified YOLOv4 network is proposed. Drone recognition challenges include the drone's far distance from the camera, a crowded background, unpredictable movements, and the drone's resemblance to birds. As an independent approach, the proposed modified YOLOv4 deep learning network architecture is capable of recognizing birds and four types of drones: multirotors, fixed-wings, helicopters, and VTOLs. To show the improved results of the new model, its performance is also compared with the base YOLOv4 network.

#### 4. Methodology

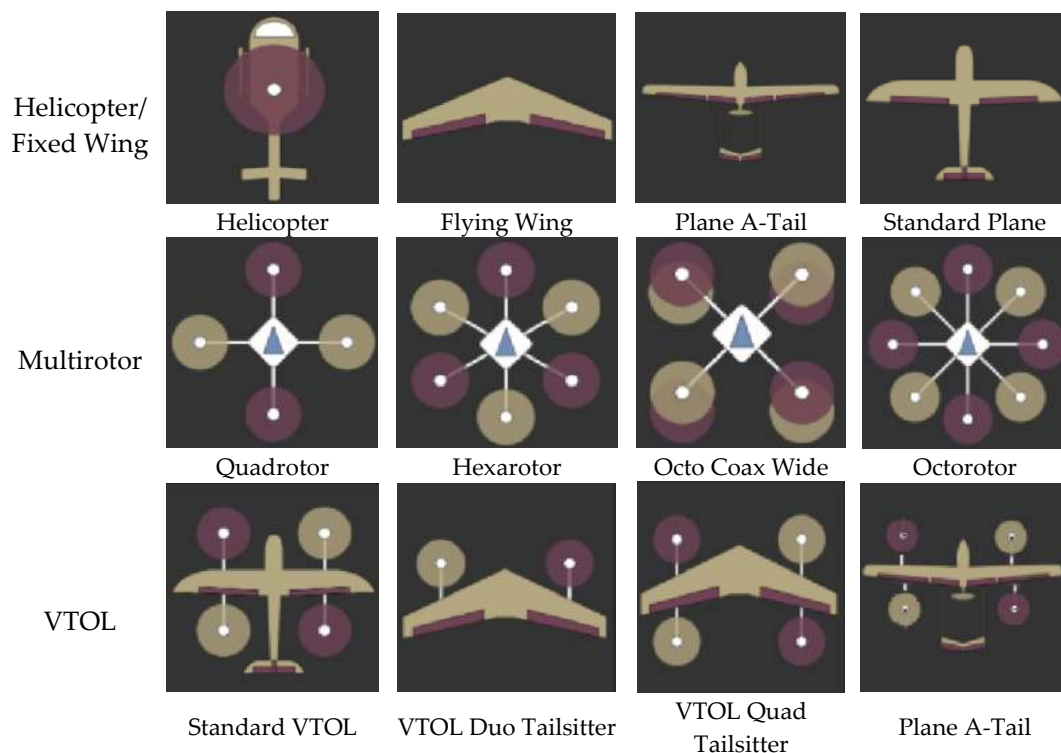
In this study, a modified network based on the latest version of the YOLO network is proposed. The steps to recognize bird species and four different types of drones are presented in Figure 4. In the first step, the input data was prepared to be ready to enter the proposed network. In the second step, the model was trained to recognize the drones, and the weight file obtained for the testing phase was generated. In the third step, the network was tested to observe how it worked; and, in the last step, the proposed deep learning network was evaluated using evaluation metrics.



**Figure 4.** Recognition process using implemented modified network.

##### 4.1. Input Preparation

Drones are generally divided into four categories: multirotors, fixed-wings, VTOLs, and helicopters. Multirotors are mainly developed in different structures such as octorotor, octo coax wide, hexarotor, and quadrotor. The fixed-wing drone can also be one of the types of fixed wing, plane a-tail, and standard plane, and the VTOL drones are the combination of both the previous versions, including four types of standard VTOL, VTOL duo tailsitter, VTOL quad tailsitter, and plane a-tail. Because of the similarity of the behavior of birds at long distances, this group of datasets belongs to the fifth category of network input data. The schematic drawings of each drone are presented in Figure 5. Thus, in this study, the data was labeled into a total of five classes and prepared for the training phase.



**Figure 5.** Different types of drones as Helicopters, Fixed-Wings, Multirotors, and VTOLs [48].

One strategy for preparing input data is to draw a rectangular bounding box around the object. In this study, the polygon sketch was used to label the drone dataset. Then, the best rectangle containing the object was fitted to the polygon. As shown in Figure 6, drawing the best rectangular bounding box around the drone results in the accurate extraction of the pixels containing the object.



**Figure 6.** The conversion of the polygon to the best-fitted rectangle.

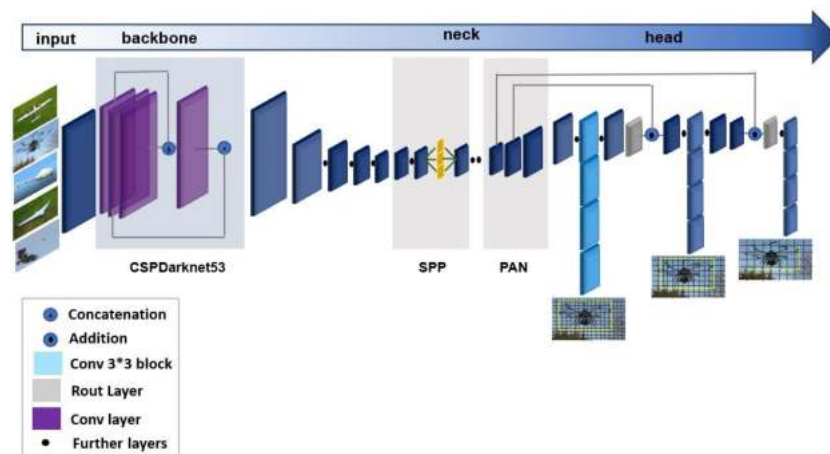
Finally, the bounding box central coordinates, the class number, and its width and its height are normalized in the range of  $[0, 1]$ , and they are introduced to the next step in the proposed method.

#### 4.2. Train the Networks

The YOLOv4 Deep Learning Network is selected as the drone vs. bird recognition network because of its advantages in this area. In addition, this network was modified to improve the performance of the basic network and to better address challenges.

##### 4.2.1. YOLOv4 Deep Learning Network Architecture

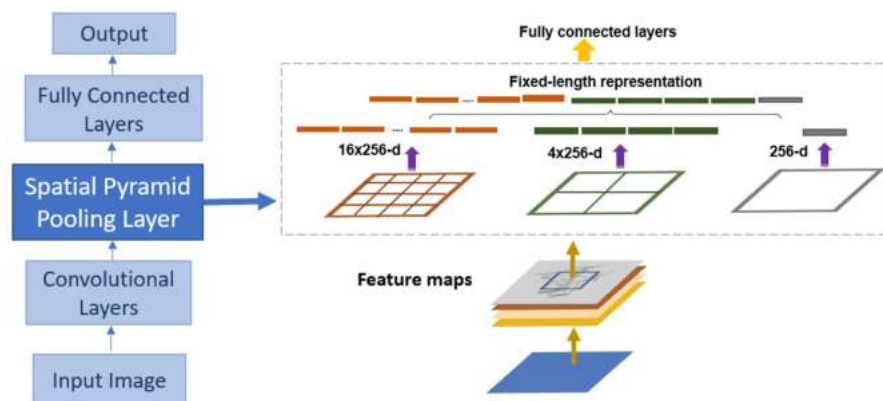
According to Figure 7, the YOLOv4 network consists of four main parts: the input of the network, backbone (feature map extractor), neck (feature map collector), and head (results of recognition).



**Figure 7.** The basic YOLOv4 Deep Learning Network Architecture.

The backbone is responsible for extracting the feature map from the previously entered data [26]. In the implemented model, a Darknet53-based network, CSP Darknet-53 [49], is used to extract the feature. CSPDarknet53 enhances the learning process of CNN. The CSPDarknet53 overhead integration pyramid section is connected to improve the receiver field and to differentiate very important context features. Extracting better features leads to increasing the accuracy of detecting drones, recognizing their type, and differentiating them from birds. In the structure of the YOLOv4 network, there are several convolution layers after the backbone. In convolution layers, internal multiplication and feature extraction operations are performed using the obtained feature maps. In this network, a  $3 \times 3$  convolutional layer is used after the backbone layer to extract more detailed and accurate features using the Mish activity function. The reason for using the Mish activity function is that this function also considers negative values, solving the problem of overfitting with precise regulatory effects.

The neck receives the feature map created in the backbone stage. This helps to add a layer between the backbone and the head. It consists of a modified Spatial Pyramid Pooling (SPP) and a modified Path Aggregation Network (PAN), both of which are used to gather information to improve accuracy [21,50]. Spatial Pyramid Pooling is an integration layer that removes the constraint of the fixed size of the network input. This layer consists of three pooling, with sizes  $256-d$ ,  $4 \times 256-d$ ,  $16 \times 256-d$ , and (Figure 8). The SPP layer receives the feature map created from the previous convolution layer. It combines features and produces fixed-length outputs, it then connects to fully connected layers, and then enters the improved PAN network. This network is used to aggregate parameters for different detector surfaces instead of feature pyramid networks (FPNs) to detect the object used in YOLOv3.



**Figure 8.** Spatial Pyramid Pooling (SPP) block.

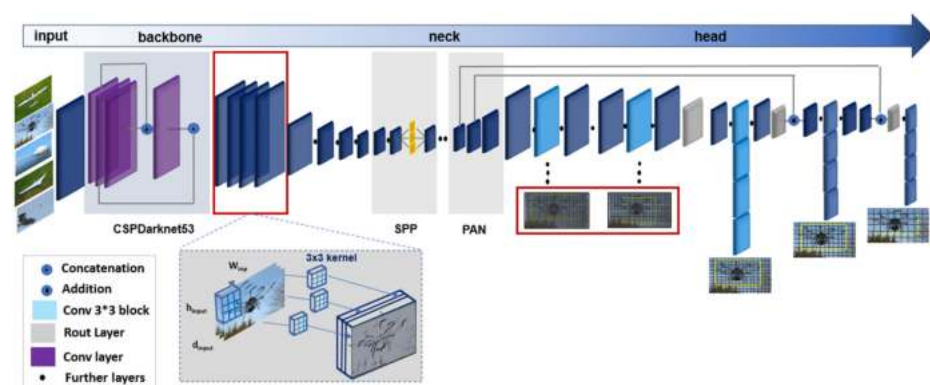
The head part is used to classify and to locate the predicted boxes of the proposed network. At this level, the probabilities and the bounding box coordinates ( $x$ ,  $y$ , height, and width) are given. This part uses the YOLOv3 deep learning network architecture, so the output is a tensor containing bounding box coordinates and class probabilities [51].

Bag-of-specials and bag-of-freebies are techniques used in the YOLOv4 algorithm to increase accuracy during and after training [26]. Bag-of-freebies helps to improve recognition during training, without increasing the inference time. This feature uses techniques such as data augmentation and drop blocks. Bag-of-freebies uses CutMix, data augmentation, and DropBlock techniques to increase and to regularize the data. It also uses techniques, such as grid sensitivity elimination, CIOU-loss, CmBN, self-adversarial training, and use of several anchors for one ground truth in detection techniques. Bag-of-specials are techniques that slightly increase the inference time and change the architecture. This feature uses methods such as non-maximum suppression, multi-input residual connections, and cross-stage partial connections (CSP) in the backbone. Additionally, in detection, it uses SPP-block, PAN-path aggregation, mish activation, SAM-block, and DIOU-NMS. Both technologies and their features are useful for training and testing the networks [26].

This network is not suitable for various applications, such as recognition of small objects and objects in cluttered backgrounds and hidden areas. Therefore, in this article, the YOLOv4 network architecture was modified to increase the accuracy of drone recognition and to improve the performance of the model to overcome challenges, which is one of the innovations in this article.

#### 4.2.2. The Modified YOLOv4 Deep Learning Network Architecture

In the modified YOLOv4 network, according to Figure 9, three convolutional layers were added to the basic YOLOv4 network architecture after the backbone. Convolutional layers are useful for extracting features from images because these layers deal with spatial redundancy by weight sharing. The addition of these three layers makes the extracted features more exclusive and informative, and it reduces redundancy. This is primarily due to the repeated cascaded convolutions and to information compression by subsampling layers. By reducing redundancy, the network displays a compressed feature about the content of the image. Consequently, by increasing the depth of the network's convolutional layers, more accurate semantic features are extracted, recognition accuracy is increased, and overcoming existing challenges is facilitated [52,53].



**Figure 9.** Recognition process using modified YOLOv4 network.

In addition, two large-scale convolutional layers were added to the head. In this part, a convolutional layer with a size of  $3 \times 3$  and another with a size of  $1 \times 1$  was used. This  $1 \times 1$  convolutional layer reduced the final depth as well as the volume of network computation. This change balances the modified network to recognize large and small long-range targets.

The implemented network can recognize drones better than the basic YOLOv4 network at long distances with small sizes and against crowded backgrounds and hidden areas. This powerful object recognition model uses a single GPU to provide accurate object recognition. For training these two networks, 70% of the entire dataset was used. The two networks were trained under the same conditions, i.e., with the same data set, the same training parameters, and the same number of iterations. Once the model implementation and training phase are complete, the testing and evaluation phase of the network begins based on the selected evaluation metrics.

#### 4.3. Test the Networks

After completing the network training, the network testing process began with 30% of the whole dataset. This stage was to select the best bounding box with the object and to evaluate the network's performance in recognizing drones and birds. The proposed learning network defines a bounding box around the detected drones. Since the drones in the dataset have different sizes and shapes, the proposed model creates multiple bounding boxes to recognize them. However, to select the most suitable bounding box out of the others, the non-maximum suppression (NMS) algorithm must be used [54]. This algorithm was used to remove bounding boxes with lower confidence scores and to select the best drone and bird box. As it appears from Figure 10, the green box is the best box that contains a drone, the other two bounding boxes also cover part of the drone, and they are candidates for drone recognition. In this algorithm, the bounding box with the highest confidence score is selected first and then the boxes with higher overlap with the selected box are removed. This process is continued until no representative boxes contain the drone. The same process is performed for the bird and, finally, the best bounding box is selected.



**Figure 10.** Proposed non-maximum suppression (NMS) box selection method.

The steps for performing the non-maximum suppression (NMS) algorithm can be summarized as follows:

- Select the predicted bounding box with the highest confidence level;
- Calculate  $IoU$  (the intersection and overlap of the selected box and other boxes) (Equation (1));

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (1)$$

- Remove boxes with an overlap of more than the default  $IoU$  threshold of 7% with the selected box;
- Repeat steps 1–3.

In Figure 11, the above algorithm was run twice, and the green boxes were selected as the final bounding boxes containing the drone and the bird.

		True Class	
		Drone	Bird
Predicted Class	Drone	True Positive (TP)	False Positive (FP)
	Bird	False Negative (FN)	True Negative (TN)

**Figure 11.** Schematic view of the confusion matrix.

#### 4.4. Evaluation Metrics

To better understand how the proposed model works, the model was evaluated using mAP, IoU, accuracy, recall, F1-score, and precision.

The mAP is one of the most important evaluation metrics. In this paper, mAP was used as an evaluation metric to recognize drones and birds. It can be claimed that mAP is responsible for comparing the predicted bounding boxes by the network with the ground truth box [19].

The other important parameter that is the key to obtaining recall, F1-score, and precision measures is the confusion matrix. Computing confusion matrix can provide valuable information on the performance of the modified model and the presence of various errors [19]. Therefore, the values of recall, F1-score, and precision can be calculated from the confusion matrix using the values true negative (TN), false negative (FN), true positive (TP), and false positive (FP). The structure of the confusion matrix is illustrated in Figure 11.

The IoU shows the connection between the ground truth bounding box and the predicted bounding box. If the intersection value of these bounding boxes is above the default threshold value of 0.7, the classification is performed correctly (TP, true positive). On the other hand, if the IoU value is below 0.7, it is misdiagnosed (FP, false positive), and if these bounding boxes do not overlap with each other, it is considered a false negative (FN).

Precision means the percentage of positive predictions among predicted classes determined to be positive [19]. Precision, F1-score, and recall metrics are calculated separately for each class of multirotor, helicopter, fixed-wing, VTOL, and bird.

The recall value shows the percentage of positive predictions among all data in the positive class. The F1-score is the mean of values for accuracy and precision, and it can indicate the validity of the classification process. This metric works well for imbalanced data because it takes into account the FN and the FP values [19].

Another metric examined in this study is the overall accuracy of the model [19]. This metric shows the performance of the model in recognizing drones and birds.

## 5. Experiments and Results

To assess the performance of proposed networks in recognizing drones and distinguishing them from birds, the steps of implementing the YOLOv4 deep learning network and the proposed modified network are presented in this section. Moreover, the types of the applied dataset, the network evaluation over different types of images, and the recognition challenges are discussed.

### 5.1. Data Preparation

To begin the training phase, a set of 26,000 visible images were first prepared that included various bird species and four types of drones such as multirotors, helicopters, fixed-wing, and VTOLs (examples are shown in Figure 12). The use of these four drone types in the collected dataset is another innovation of this study. Public images and videos

were used to create the dataset. Approximately 70% of these images were used for training and 30% for testing the network. In both phases, the number of images used is the same in each class. To prepare these images, first, a polygon around the target was determined. This task was handled using an efficient and useful Computer Vision Annotation Tool (CVAT) and the input images were categorized into five classes. Based on these boxes, the best rectangles containing the drone were then selected and passed to the network. In this dataset, multirotors are classified in the first class, helicopters in the second class, fixed-wing aircraft in the third class, VTOL in the fourth class, and birds in the fifth class.



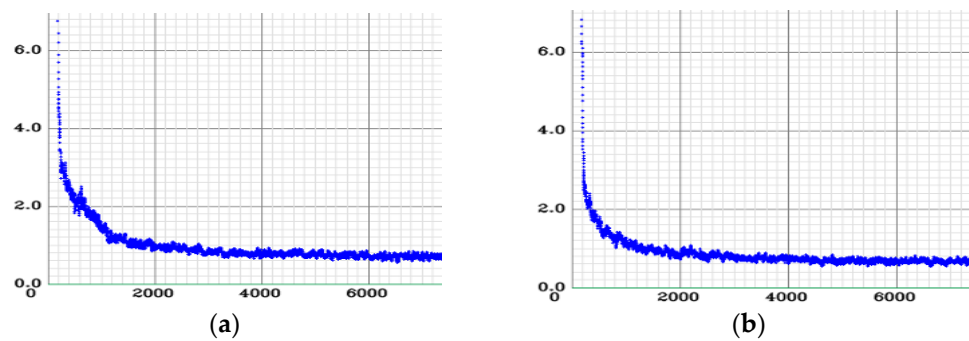
**Figure 12.** Included in the drone dataset.

### 5.2. Model Implementation and Training Results

The modified YOLOv4 network and the basic YOLOv4 network were trained using an Nvidia GeForce MX450 Graphics Processing Unit (GPU) hardware with 30,000 iterations. To train these two networks, the settings in the configuration file were changed as follows:

- The input image size was set to  $160 \times 160$ ;
- The subdivision and the batch parameters were changed to 1 and 64, respectively (these settings were made to avoid errors due to lack of memory);
- The learning rate was changed to 0.0005;
- The step parameter was changed to 24,000 and 27,000, (with 80% and 90% of the number of iterations, respectively);
- The size of the filter in three convolutional layers near the YOLO layers was changed to 30 according to the number of classes.

The basic network used for training is the widely used Darknet Framework [55]. Depending on the hardware used, a CUDA (Compute Unified Device Architecture) Toolkit version 10.0, CUDNN (CUDA Deep Neural Network library) version 8.2, Visual Studio 2017, and OpenCV version 4.0.1 are used. During the training, a graph of the number of iterations and loss was plotted as shown in Figure 13. After 30 k iterations (about 3–4 days), the modified implemented method and the basic YOLOv4 model achieved a loss of 0.58 and 0.68, respectively. The obtained values show that the detection loss of the proposed network was lower than that of the basic network, indicating the better performance of the proposed model in the training stage.



**Figure 13.** The loss graph in the training process; (a) The YOLOv4 network. (b) The modified YOLOv4 network.

### 5.3. Evaluation of the YOLOv4 and Modified YOLOv4 Models

The modified YOLOv4 and the basic YOLOv4 networks were precisely evaluated using accuracy, mAP, recall, precision, and F1-score in recognizing the drones. Tables 1 and 2 show the evaluation result of the proposed model and the basic YOLOv4 network. Accordingly, the performance of both models was compared with the same dataset, with an IoU default threshold of 0.07 and the same number of iterations. The recognition performance of the modified model was performed in five classes: multi-rotor, helicopter, fixed-wing, VTOL, and birds; and, the evaluation metrics of precision, recall, F1-score, accuracy, mAP, and IoU were increased. These results indicate the better performance of the modified model than the basic YOLOv4 model. In this network, the accuracy was 0.83%, the mAP was 83%, and the IoU was 84%, which was 4% better than the basic model.

**Table 1.** Evaluation results of the basic and Modified YOLOv4 networks.

Dataset	Model	Num of Images	Precision %	Recall %	F1-Score %
Bird	YOLOv4	1570	81	87	84
	Modified YOLOv4		87	90	89
Fixed Wing	YOLOv4	1570	88	70	78
	Modified YOLOv4		88	77	82
Helicopter	YOLOv4	1570	81	73	77
	Modified YOLOv4		88	73	80
Multirotor	YOLOv4	1570	77	90	83
	Modified YOLOv4		79	90	84
VTOL	YOLOv4	1570	72	77	74
	Modified YOLOv4		74	83	78
Total	YOLOv4	7850	80	79	79
	Modified YOLOv4		83	83	83

**Table 2.** Total evaluation results of the basic and Modified YOLOv4 networks.

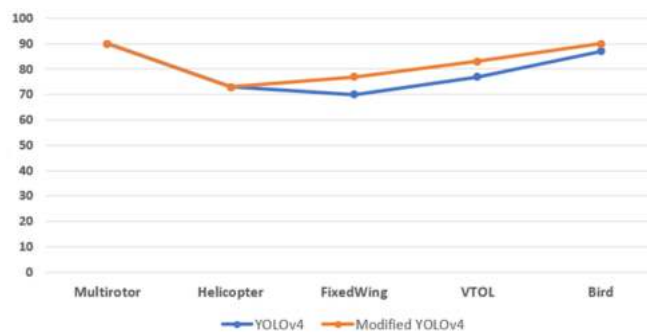
Dataset	Model	Num of Images	Accuracy %	mAP %	IoU %
Total	YOLOv4	7850	79	79	80
	Modified YOLOv4		83	83	84

The comparison of precision, recall, F1-score, and total mAP in all five classes in the two implemented models are graphically presented in Figure 14. The precision was improved in four classes (multirotor, helicopter, VTOL, and bird) and unchanged in the fixed-wing class. In general, these results show the improvement of model performance in

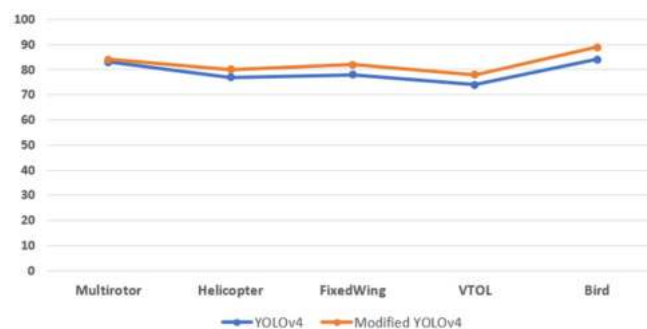
the proposed network. By comparing the recall according to Figure 15, its improvement is observed in three classes, and there is no change in the other two classes. This means that the ability and the performance of the second model in recognition were improved. As it appears from Figure 16, the F1-score increased compared to the basic model of YOLOv4, and this also shows the better performance of the modified YOLOv4 network.



**Figure 14.** The comparison between the precision values in the basic and the modified YOLOv4 models.

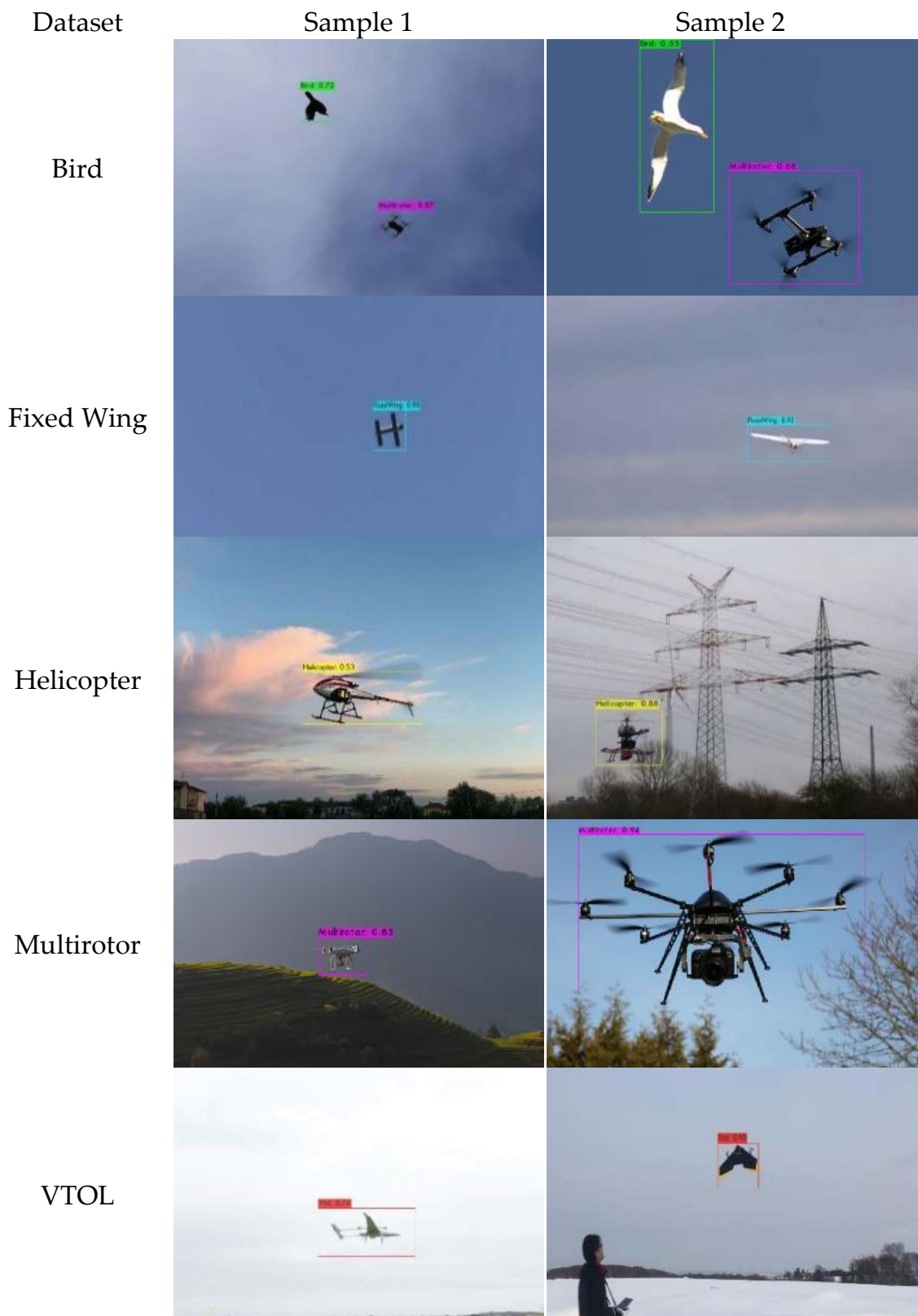


**Figure 15.** The comparison between the recall values in the basic and the modified YOLOv4 models.

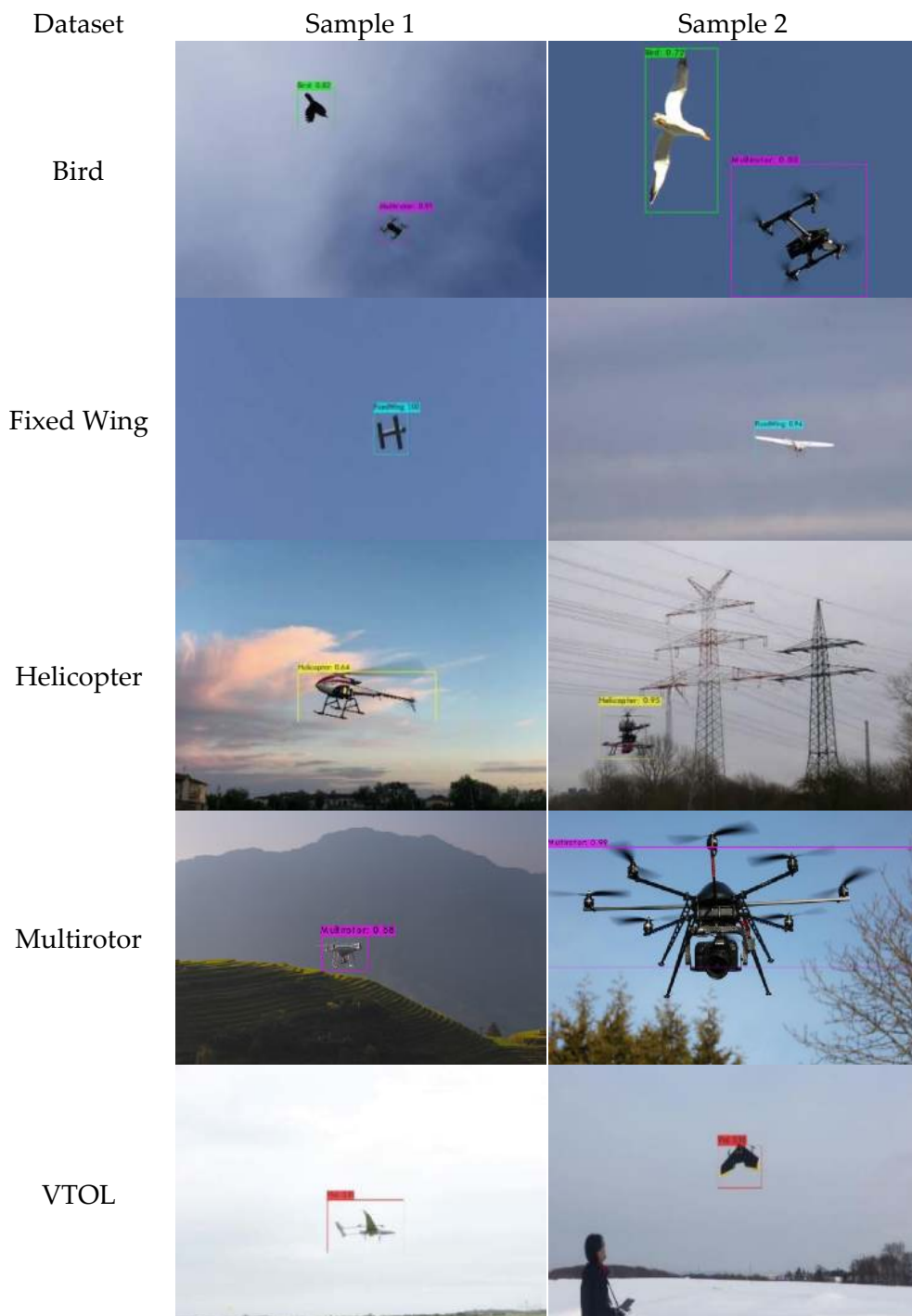


**Figure 16.** The comparison between F1-score in the basic and the modified YOLOv4 models.

Figure 17 shows the results of drone recognition in the basic YOLOv4 network and Figure 18 shows these results with the modified YOLOv4 network. A comparison of the results of the two networks shows an improvement in the accuracy of the bounding boxes and the class probabilities in the modified network. On the other hand, the recognition of four types of multirotor (octotorotor, octo coax wide, hexarotor, and quadrotor), three types of fixed-wing (flying wing, plane a-tail, and standard plane), four types of VTOLs (standard VTOL, VTOL duo tailsitter, VTOL quad tailsitter, and plane a-tail), one type of helicopter, and their discrimination from birds were improved in the proposed model.



















**Figure 17.** Some samples of drone-vs-bird recognition results with a basic YOLOv4 network.



**Figure 18.** Some samples of drone-vs-bird recognition results with a modified YOLOv4 network.

#### 5.4. Addressing the Challenges in the Modified YOLOv4 Model

In this study, the YOLOv4 network was modified to improve the recognition results of four types of drones and to differentiate them from birds concerning existing challenges, such as the smaller size of the drones, crowded backgrounds, loss of scalability, and similarity with birds. Figure 19 shows the performance of both the basic and the modified models in overcoming the challenges.

Challenge	Model	Sample 1	Sample 2
(a) Confusion with bird	YOLOv4		
	Modified YOLOv4		
(b) Crowded backgrounds	YOLOv4		
	Modified YOLOv4		
(c) Small drone size	YOLOv4		
	Modified YOLOv4		
(d) Loss of scalability	YOLOv4		
	Modified YOLOv4		

**Figure 19.** Samples of handling challenges in drone recognition.

The first challenge is to recognize the drone and to distinguish it from the bird; the basic model is unable to recognize the bird in some samples and the class probability values are lower than for the modified model. The modified YOLOv4 model can recognize drones and distinguish them from birds by extracting smaller and more accurate features, and it has demonstrated this ability in the evaluated samples (Figure 19). The second challenge studied is the presence of drones in a crowded background, where the modified model can recognize targets with a high prediction probability. In some cases, the base model is unable to recognize the drone or it has a lower probability than the modified model. The third challenge is the small size of the drones and their placement at long ranges, where the modified model performs better than the base model. The modified model can accurately recognize all small drones in the test images. This ability is due to the use of more convolutional layers at the head of the network. The change in network structure does not remarkably affect the execution time of the network training algorithm. The training of the proposed network takes approximately 3 h longer than that of the first network. Finally, the challenge of the detectability swarm of UAVs at different scales is investigated. In this challenge, the base model is unable to recognize drones in the images in some cases, and it generally has lower class probability than the modified model.

Figure 20 illustrates several samples of different drone types in crowded environments with lighting conditions and different weather. As the results show, the modified YOLOv4 network is capable of recognizing different drone types under complex and challenging conditions.



**Figure 20.** Several samples of drone recognition in different light and weather conditions and against a crowded background. (a) YOLOv4. (b) Modified YOLOv4.

## 6. Discussion

The evaluation metric examined in this study is confusion matrix, IoU, mAP, accuracy, precision, recall, and F1-score. The values obtained from these criteria in the modified network are as follows: 79% precision, 84% F1-score, and 90% recall for the multirotor; 88% precision, 82% F1-score, and 77% recall for the fixed-wing; 88% precision, 80% F1-score, and 73% recall for the helicopter; 74% precision, 78% F1-score, and 83% recall

for VTOL; and 87% precision, 89% F1-score, and 90% recall for the bird. Based on the basic YOLOv4 network results, it can be said that the proposed network had an average improvement of 3.4% in precision, 3.2% in the recall, and 3.4% in F1-score values. The overall accuracy, indicating the correct classification of the input data set into five classes, was also investigated. This criterion increased by 4% in the modified network compared to the basic network. The mAP also increased by 4% in the modified network compared to the basic YOLOv4 network. This rate indicates an improvement in the classification of the proposed model in the five classes. Finally, the IoU metric was used to calculate the overlap value of the predicted bounding box with the ground truth bounding box. This metric also increased by 4% in the modified network, indicating the higher accuracy of the predicted bounding box.

In recent years, the use of artificial intelligence and deep learning methods has become one of the most popular and useful methods in object recognition. In 2021, Tan Wei Xun et al. detected and tracked a drone using the YOLOv3. The results of this method show that the proposed YOLOv3 network detects drones of different sizes with a confidence level between 60 and 100% [44]; also this year, drone detection and tracking were performed by Isaac-Medina et al. using a set of visible and thermal images and four deep learning network architectures, including RCNN, SSD, YOLOv3, and DETR. According to the results of this study, it can be said that the YOLOv3 deep learning network performs better than the other models [45]. One of the problems with these studies was the inability to detect small drones over long distances. In addition, an automatic drone detection system using YOLOv4 was developed by Singha et al. In this system, a collection of drone-vs-bird images were used to detect the drone and its evaluation was tested using a video dataset. The results of this work include the high accuracy of the YOLOv4 network [46]. One of the limitations of this study was the lack of recognition of the different types of drones and the lack of challenging images. Moreover, Liu et al. applied three object detection methods, such as YOLOv3, YOLOv4, RetinaNet, and FCOS networks on the drone dataset. To build a scattered, flat network and to get great accuracy in drone detection, the pruned YOLOv4 model was used, which improved the accuracy of drone detection at high speed. However, the study did not address the challenges of crowded background images, hidden areas, and the distance of the drone from the camera [16]. Additionally in 2022, Samadzadegan et al. recognized two types of drones (multirotors and helicopters) and birds using the YOLOv4 deep learning network. However, this paper did not recognize two other drone types such as VTOL and fixed-wing drones, which could be more dangerous than the previous two types. In addition, a rectangular bounding box was used in the data preparation, which resulted in the input of additional information and reduced the accuracy of recognition [47].

Previous studies have focused on drone detection, and a small number of these studies have examined the recognition of different types of drones. Moreover, the challenges of drone recognition, such as the small size of the drone, crowded background, hidden areas, and confusion with birds, have not been comprehensively addressed in these studies. Therefore, it can be said that the unauthorized presence of drones in challenging environments and their inaccurate recognition in sensitive infrastructures is still one of the most important problems in ensuring public safety. The main goal of this research is to recognize four types of drones and to differentiate them from birds at far distances despite challenges, such as the small size of the drone, a crowded background, and the presence of hidden areas.

In this study, the YOLOv4 network was modified to improve drone recognition challenges. A set of visible images with different types of drones and birds in different environments at near and far distances were collected to recognize four types of drones and to differentiate them from birds. Two convolutional layers were added to the head of the YOLOv4 to solve the challenges of small drone recognition. For example, in Figure 20 there are examples of small drones that the basic model was unable to recognize in some cases. However, the modified model recognized them; and, in other cases they operated

with less accuracy than the modified network. This result shows the improvement of the current network compared to the basic network for recognizing small drones. Furthermore, to increase the accuracy of feature extraction and precise recognition of drones and birds, three convolutional layers were added after the backbone layer. Adding these layers to the architecture of the basic YOLOv4 network did not change the training time of the network, and it took only a few hours longer than training the basic network. By applying these changes in the network architecture and using extensive datasets, the proposed method was able to recognize all drone types and bird species in challenging environments. In Figure 19, rows (a), (b), and (d) provide examples of difficult images and swarms of UAVs, all of which have higher recognition accuracy in the modified network than in the basic network. Figure 20 also contains other challenging examples where the proposed network performs better.

## 7. Conclusions

As has been noted, UAV recognition in various situations is a complex process; the usual methods and even conventional deep learning network methods do not work well in some cases. In this study, the basic YOLOv4 network was used to recognize drone types and to differentiate them from bird species. To increase recognition accuracy and to better address existing challenges, a novel and modified model of this network was proposed. To train, test, and evaluate these two networks, a collection of 26,000 visible image datasets including four types of UAVs (multirotor, fixed-wing, helicopter, and VTOL) and birds were collected. The comparison of these two models was done using mAP, confusion matrix, IoU, precision, accuracy, F1-score, and recall evaluation metrics. With the modified YOLOv4 model, we achieved 84% IoU, 83% mAPs, and 83% accuracy, which is better than the basic model, and it solved the challenges well. In the future, real-time identification with onboard systems can be studied in addition to drone recognition. Background removal algorithms can also be used to make labeling input data easier and faster. In addition to the multirotor types used, Tri-rotors can also be used to complete the dataset. Additionally, other deep learning networks can be used to compare their results with the results of this modified network.

**Author Contributions:** All authors contributed to the study conception and design. F.D.J. contributed to supervision, reviewing, and validation. F.S. is a drone expert involved in the conceptualization, methodology, and editing of the draft. F.A.M. contributed to programing, visualization, computer vision concepts, writing, and editing. M.G. is involved in software, deep learning concepts, data collection and preparation, and drafting. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors have no conflict of interest to disclose.

## References

1. Mueller, M.; Smith, N.; Ghanem, B. *A Benchmark and Simulator for UAV Tracking*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9905, pp. 445–461.
2. Wu, M.; Xie, W.; Shi, X.; Shao, P.; Shi, Z. Real-time drone detection using deep learning approach. In Proceedings of the International Conference on Machine Learning and Intelligent Communications, Hangzhou, China, 6–8 July 2018; Springer: Berlin/Heidelberg, Germany, 2018.
3. Bansod, B.; Bansod, B.; Singh, R.; Thakur, R.; Singhal, G. A comparison between satellite based and drone based remote sensing technology to achieve sustainable development: A review. *J. Agric. Environ. Int. Dev.* **2017**, *111*, 383–407.
4. Orusa, T.; Orusa, R.; Viani, A.; Carella, E.; Borgogno Mondino, E. Geomatics and EO Data to Support Wildlife Diseases Assessment at Landscape Level: A Pilot Experience to Map Infectious Keratoconjunctivitis in Chamois and Phenological Trends in Aosta Valley (NW Italy). *Remote Sens.* **2020**, *12*, 3542. [CrossRef]

5. Chiu, M.; Xu, X.; Wei, Y.; Huang, Z.; Schwing, A.; Brunner, R.; Khachatrian, H.; Karapetyan, H.; Dozier, I.; Rose, G.; et al. Agriculture-Vision: A Large Aerial Image Database for Agricultural Pattern Analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.
6. Anwar, M.Z.; Kaleem, Z.; Jamalipour, A. Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2526–2534. [CrossRef]
7. Sathyamoorthy, D. A Review of Security Threats of Unmanned Aerial Vehicles and Mitigation Steps. *J. Def. Secur.* **2015**, *6*, 81–97.
8. Yaacoub, J.-P.; Noura, H.; Salman, O.; Chehab, A. Security Analysis of Drones Systems: Attacks, Limitations, and Recommendations. *Internet Things* **2020**, *11*, 100218. [CrossRef]
9. Semkin, V.; Yin, M.; Hu, Y.; Mezzavilla, M.; Rangan, S. Drone Detection and Classification Based on Radar Cross Section Signatures. In Proceedings of the 2020 International Symposium on Antennas and Propagation (ISAP), Osaka, Japan, 25–28 January 2021.
10. Haag, M.U.D.; Bartone, C.G.; Braasch, M.S. Flight-test evaluation of small form-factor LiDAR and radar sensors for sUAS detect-and-avoid applications. In Proceedings of the 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), Sacramento, CA, USA, 25–29 September 2016.
11. Svanstrom, F.; Englund, C.; Alonso-Fernandez, F. Real-Time Drone Detection and Tracking With Visible, Thermal and Acoustic Sensors. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2020.
12. Andraši, P.; Radišić, T.; Muštra, M.; Ivošević, J. Night-time Detection of UAVs using Thermal Infrared Camera. *Transp. Res. Procedia* **2017**, *28*, 183–190. [CrossRef]
13. Nguyen, P.; Ravindranatha, M.; Nguyen, A.; Han, R.; Vu, T. Investigating Cost-effective RF-based Detection of Drones. In Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use, Singapore, 26 June 2016; pp. 17–22.
14. Humphreys, T.E. *Statement on the Security Threat Posed by Unmanned Aerial Systems and Possible Countermeasures*; Oversight and Management Efficiency Subcommittee, Homeland Security Committee: Washington, DC, USA, 2015.
15. Drozdowicz, J.; Wielgo, M.; Samczynski, P.; Kulpa, K.; Krzonkalla, J.; Mordzonek, M.; Bryl, M.; Jakielaszek, Z. 35 GHz FMCW drone detection system. In Proceedings of the 2016 17th International Radar Symposium (IRS), Krakow, Poland, 10–12 May 2016.
16. Liu, H.; Fan, K.; Ouyang, Q.; Li, N. Real-time small drones detection based on pruned yolov4. *Sensors* **2021**, *21*, 3374. [CrossRef]
17. Seidaliyeva, U.; Alduraibi, M.; Ilipbayeva, L.; Almagambetov, A. Detection of loaded and unloaded UAV using deep neural network. In Proceedings of the 2020 4th IEEE International Conference on Robotic Computing (IRC), Taichung, Taiwan, 9–11 November 2020.
18. Ashraf, M.; Sultani, W.; Shah, M. Dogfight: Detecting Drones from Drones Videos. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 7063–7072.
19. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
20. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2013.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *Computer Vision—ECCV 2014*; Springer International Publishing: Cham, Switzerland, 2014.
22. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; MIT Press: Cambridge, MA, USA, 2015; Volume 1, pp. 91–99.
23. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
24. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Volume 9905, pp. 21–37.
25. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
26. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
27. Chaurasia, R.; Mohindru, V. Unmanned aerial vehicle (UAV): A comprehensive survey. In *Unmanned Aerial Vehicles for Internet of Things (IoT) Concepts, Techniques, and Applications*; Wiley: New Delhi, India, 2021; pp. 1–27.
28. Gu, H.; Lyu, X.; Li, Z.; Shen, S.; Zhang, F. Development and experimental verification of a hybrid vertical take-off and landing (VTOL) unmanned aerial vehicle (UAV). In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017.
29. Cai, G.; Lum, K.; Chen, B.M.; Lee, T.H. A brief overview on miniature fixed-wing unmanned aerial vehicles. In Proceedings of the IEEE ICCA, Xiamen, China, 9–11 June 2010.
30. Kotarski, D.; Piljek, P.; Pranjić, M.; Grlj, C.G.; Kasać, J. A Modular Multirotor Unmanned Aerial Vehicle Design Approach for Development of an Engineering Education Platform. *Sensors* **2021**, *21*, 2737. [CrossRef] [PubMed]
31. Cai, G.; Chen, B.M.; Lee, T.H.; Lum, K.Y. Comprehensive nonlinear modeling of an unmanned-aerial-vehicle helicopter. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI, USA, 18–21 August 2008.
32. Qin, B.; Zhang, D.; Tang, S.; Wang, M. Distributed Grouping Cooperative Dynamic Task Assignment Method of UAV Swarm. *Appl. Sci.* **2022**, *12*, 2865. [CrossRef]

33. Shafiq, M.; Ali, Z.A.; Israr, A.; Alkhamash, E.H.; Hadjouni, M. A Multi-Colony Social Learning Approach for the Self-Organization of a Swarm of UAVs. *Drones* **2022**, *6*, 104. [CrossRef]
34. Ali, Z.A.; Han, Z.; Masood, R.J. Collective Motion and Self-Organization of a Swarm of UAVs: A Cluster-Based Architecture. *Sensors* **2021**, *21*, 3820. [CrossRef]
35. Xu, C.; Zhang, K.; Jiang, Y.; Niu, S.; Yang, T.; Song, H. Communication Aware UAV Swarm Surveillance Based on Hierarchical Architecture. *Drones* **2021**, *5*, 33. [CrossRef]
36. Li, Y. Research and application of deep learning in image recognition. In Proceedings of the 2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 21–23 January 2022; IEEE: Piscataway, NJ, USA, 2022.
37. Pathak, A.R.; Pandey, M.; Rautaray, S. Application of deep learning for object detection. *Procedia Comput. Sci.* **2018**, *132*, 1706–1717. [CrossRef]
38. Deng, L.; Yu, D. Deep Learning: Methods and Applications. In *Foundations and Trends® in Signal Processing*; Now Publishers Inc.: Hanover, NH, USA, 2014; Volume 7, pp. 197–387.
39. Nalamati, M.; Kapoor, A.; Saqib, M.; Sharma, N.; Blumenstein, M. Drone Detection in Long-Range Surveillance Videos. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–6.
40. Unlu, E.; Zenou, E.; Riviere, N.; Dupouy, P.-E. Dupouy Deep learning-based strategies for the detection and tracking of drones using several cameras. *IPSN Trans. Comput. Vis. Appl.* **2019**, *11*, 7. [CrossRef]
41. Mahdavi, F.; Rajabi, R. Drone Detection Using Convolutional Neural Networks. In Proceedings of the 2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), Mashhad, Iran, 23–24 December 2020; IEEE: Piscataway, NJ, USA, 2020.
42. Behera, D.K.; Raj, A.B. Drone detection and classification using deep learning. In Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; IEEE: Piscataway, NJ, USA, 2020.
43. Shi, Q.; Li, J. Objects Detection of UAV for Anti-UAV Based on YOLOv4. In Proceedings of the 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Weihai, China, 14–16 October 2020; IEEE: Piscataway, NJ, USA, 2020.
44. Xun, D.T.W.; Lim, Y.L.; Srigrarom, S. Drone detection using YOLOv3 with transfer learning on NVIDIA Jetson TX2. In Proceedings of the 2021 2nd International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 20–22 January 2021; IEEE: Piscataway, NJ, USA, 2021.
45. Isaac-Medina, B.K.; Poyser, M.; Organisciak, D. Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark. *arXiv* **2021**, arXiv:2103.13933.
46. Singha, S.; Aydin, B. Automated Drone Detection Using YOLOv4. *Drones* **2021**, *5*, 95. [CrossRef]
47. Samadzadegan, F.; Javan, F.D.; Mahini, F.A.; Gholamshahi, M. Detection and Recognition of Drones Based on a Deep Convolutional Neural Network Using Visible Imagery. *Aerospace* **2022**, *9*, 31. [CrossRef]
48. Roche, R. QGroundControl (QC). 2019. Available online: <http://qgroundcontrol.com/> (accessed on 20 May 2022).
49. Wang, C.; Liao, H.M.; Wu, Y.; Chen, P.; Hsieh, J.; Yeh, I. CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020.
50. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
51. Redmon, J.; Farhadi, A. Yolo3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
52. Nanculef, R.; Radeva, P.; Balocco, S. Training Convolutional Nets to Detect Calcified Plaque in IVUS Sequences. In *Intravascular Ultrasound*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 141–158.
53. Wang, L.; Lee, C.-Y.; Tu, Z.; Lazebnik, S. Training deeper convolutional networks with deep supervision. *arXiv* **2015**, arXiv:1505.02496.
54. Hosang, J.; Benenson, R.; Schiele, B. Learning non-maximum suppression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
55. Redmon, J. Darknet: Open Source Neural Networks in C. 2013–2016. Available online: <http://pjreddie.com/darknet/> (accessed on 20 May 2022).

## Article

# Anti-Occlusion UAV Tracking Algorithm with a Low-Altitude Complex Background by Integrating Attention Mechanism

Chuanyun Wang <sup>1,\*</sup>, Zhongrui Shi <sup>2</sup>, Linlin Meng <sup>1</sup>, Jingjing Wang <sup>3</sup>, Tian Wang <sup>4</sup>, Qian Gao <sup>1</sup>  
and Ershen Wang <sup>5</sup>

<sup>1</sup> College of Artificial Intelligence, Shenyang Aerospace University, Shenyang 110136, China; menglinlin@stu.sau.edu.cn (L.M.); gaoqian@buaa.edu.cn (Q.G.)

<sup>2</sup> School of Computer Science, Shenyang Aerospace University, Shenyang 110136, China; shizhongrui@stu.sau.edu.cn

<sup>3</sup> China Academic of Electronics and Information Technology, Beijing 100041, China; wangjingjing@cetcloud.com

<sup>4</sup> Institute of Artificial Intelligence, Beihang University, Beijing 100191, China; wangtian@buaa.edu.cn

<sup>5</sup> School of Electronic and Information Engineering, Shenyang Aerospace University, Shenyang 110136, China; wes2016@sau.edu.cn

\* Correspondence: wangcy0301@sau.edu.cn

**Abstract:** In recent years, the increasing number of unmanned aerial vehicles (UAVs) in the low-altitude airspace have not only brought convenience to people's work and life, but also great threats and challenges. In the process of UAV detection and tracking, there are common problems such as target deformation, target occlusion, and targets being submerged by complex background clutter. This paper proposes an anti-occlusion UAV tracking algorithm for low-altitude complex backgrounds by integrating an attention mechanism that mainly solves the problems of complex backgrounds and occlusion when tracking UAVs. First, extracted features are enhanced by using the SeNet attention mechanism. Second, the occlusion-sensing module is used to judge whether the target is occluded. If the target is not occluded, tracking continues. Otherwise, the LSTM trajectory prediction network is used to predict the UAV position of subsequent frames by using the UAV flight trajectory before occlusion. This study was verified on the OTB-100, GOT-10k and integrated UAV datasets. The accuracy and success rate of integrated UAV datasets were 79% and 50.5% respectively, which were 10.6% and 4.9% higher than those of the SiamCAM algorithm. Experimental results show that the algorithm could robustly track a small UAV in a low-altitude complex background.

**Keywords:** unmanned aerial vehicle; target tracking; attention mechanism; anti-occlusion; location prediction

**Citation:** Wang, C.; Shi, Z.; Meng, L.; Wang, J.; Wang, T.; Gao, Q.; Wang, E. Anti-Occlusion UAV Tracking Algorithm with a Low-Altitude Complex Background by Integrating Attention Mechanism. *Drones* **2022**, *6*, 149. <https://doi.org/10.3390/drones6060149>

Academic Editors: Daobo Wang and Zain Anwar Ali

Received: 11 May 2022

Accepted: 14 June 2022

Published: 16 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, with the rapid development of the UAV industry and the continuous improvement of artificial intelligence, UAVs have been widely used in public security, disaster relief, photogrammetry, news broadcasts, travel, and other fields, bringing great convenience to production and social life. However, the increasing number of UAVs in low-altitude airspace and the frequent occurrence of various illegal flight incidents have brought great threats and challenges to aviation flight, security, confidentiality protection and privacy protection [1].

In order to detect UAV in the low-altitude airspace as early and as far as possible using computer vision, it is often necessary to implement the long-distance detection and tracking of UAVs [2], which cause small imaging sizes and weak signals [3]. At the same time, the flight altitude of UAV in low-altitude airspace is very low, often only dozens to hundreds of meters, and the surrounding environment of this altitude is relatively complex, such as trees, buildings, and walls, which may lead to the visual tracking of UAV being interfered

by strong clutter, occlusion, and other factors, resulting in tracking drift and loss. Therefore, it is important and urgent to find a robust tracking algorithm against background clutter interference and occlusion for UAV tracking in low-altitude airspace.

On the basis of combining existing UAV visual tracking technology and referring to the network structure of ATOM [4], this paper proposes an anti-occlusion target tracking algorithm by integrating the SeNet [5] attention mechanism to solve the complex background and occlusion problems during tracking, which achieved good performance. First, the SeNet attention mechanism was introduced into the original feature extraction network to enhance the extracted features, which effectively improved the performance of subsequent tracking process. Second, an occlusion-sensing model was designed to judge the state of the target. Lastly, the LSTM [6] trajectory prediction network was used to predict the UAV position according to the target state. This study was verified on the OTB-100 [7], GOT-10k [8] and integrated UAV datasets. Experimental results show that the proposed algorithm could effectively reduce the influence of low-altitude complex environments on the target and robustly track a UAV.

The main contributions of this paper are:

1. In order to solve the problem of UAVs in the low-altitude airspace being easy to be submerged in complex background clutter, the SeNet attention mechanism was used in the backbone to improve the correlation between feature channels, enhance the feature of the target, and reduce the influence of background clutter.
2. In order to solve the occlusion problem in low-altitude airspace during flight, an occlusion judgment mechanism is proposed to judge whether the target is occluded.
3. When the target is occluded, the LSTM trajectory prediction network is used to predict the flight trajectory of the aircraft, so as to achieve robust tracking and stop the template update to improve tracking accuracy.

## 2. Related Works

Existing target tracking algorithms can be roughly divided into two categories: One is the traditional target tracking method based on correlation filtering [9–13], which uses the response diagram between the template frame and the detection frame after Fourier transform to determine the target of the detection frame. The other is the deep-learning target tracking method based on a convolutional neural network [14–17], which obtains the features of the target by convolutional operation on the images of the template and detection frames, and then obtains the tracking target by similarity matching. This section reviews the related work of researchers in recent years.

### 2.1. Algorithm Based on Correlation Filter

The tracker based on a discriminant correlation filter (DCF) can effectively use limited data and enhance the training set by using all shifts of local training samples in the learning process. The method based on DCF trains the least-squares regression to predict the target confidence score by using the characteristics of cyclic correlation and fast Fourier transform (FFT) in the learning and detection steps [18]. Mosse [9] was the first pioneering work to propose correlation filter for tracking that uses a random affine set of samples from a single initial frame transformation to construct a minimal output sum of the squares' filter. KCF reduces storage and calculation [6] by several orders of magnitude by diagonalizing the cyclic data matrix with discrete Fourier transform. The periodic assumption of KCF also introduces an unnecessary boundary effect, which seriously reduces the quality of tracking model. SRDCF introduces a spatial regularization component in the learning process in order to reduce the boundary effect, which punishes them according to the spatial position of the correlation filter coefficients [19]. In addition, there are several excellent trackers based on correlation filters, such as STRCF [20] and ECO [8]. They usually divide tracking into two stages: feature extraction and target classification, so end-to-end training is not possible. The objective function in the target classification module in ATOM [13] is based on the mean square error, like the discriminant correlation filtering method, but it is established