telecom

MDPI

Article

Hybrid Artificial-Intelligence-Based System for Unmanned Aerial Vehicle Detection, Localization, and Tracking Using Software-Defined Radio and Computer Vision Techniques

Pablo López-Muñoz [1], Luis Gimeno San Frutos [1], Christian Abarca [2], Francisco José Alegre [2], Jose Luis Calle [2] and Jose F. Monserrat [1,*]

[1] Universitat Politècnica de València-iTEAM, 46022 Valencia, Spain; pablomuo@upv.edu.es (P.L.-M.); luigisa1@teleco.upv.es (L.G.S.F.)
[2] Grupo Simetria, 12100 Castellon, Spain; cabarca@simetriagrupo.com (C.A.); fjalegre@simetriagrupo.com (F.J.A.); jlcalle@simetriagrupo.com (J.L.C.)
* Correspondence: jomondel@upv.es

Abstract: The proliferation of drones in civilian environments has raised growing concerns about their misuse, highlighting the need to develop efficient detection systems to protect public and private spaces. This article presents a hybrid approach for UAV detection that combines two artificial-intelligence-based methods to improve system accuracy. The first method uses a software-defined radio (SDR) to analyze the radio spectrum, employing autoencoders to detect drone control signals and identify the presence of these devices. The second method is a computer vision module consisting of fixed cameras and a PTZ camera, which uses the YOLOv10 object detection algorithm to identify UAVs in real time from video sequences. Additionally, this module integrates a localization and tracking algorithm, allowing the tracking of the intruding UAV's position. Experimental results demonstrate high detection accuracy, a significant reduction in false positives for both methods, and remarkable effectiveness in UAV localization and tracking with the PTZ camera. These findings position the proposed system as a promising solution for security applications.

Keywords: UAV; SDR; autoencoders; YOLOv10; tracking; localization

check for updates

Citation: López-Muñoz, P.; San Frutos, L.G.; Abarca, C.; Alegre, F.J.; Calle, J.L.; Monserrat, J.F. Hybrid Artificial-Intelligence-Based System for Unmanned Aerial Vehicle Detection, Localization, and Tracking Using Software-Defined Radio and Computer Vision Techniques. *Telecom* **2024**, *5*, 1286–1308. https://doi.org/10.3390/telecom5040064

Academic Editors: Yichuang Sun, Haeyoung Lee and Oluyomi Simpson

Received: 23 October 2024
Revised: 22 November 2024
Accepted: 2 December 2024
Published: 11 December 2024

Copyright: © 2024 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, due to technological development and cost reduction, unmanned aerial vehicles (UAVs) have experienced exponential growth, making them devices that are now accessible for the personal use of any citizen. The increasing popularity of UAVs can also be attributed to their great versatility, as they offer a wide range of applications, from recreational and informal uses to professional use in tasks such as search and rescue operations, fire control, and other natural disaster operations, as described by Shakhatreh et al., who analyzed the growing use of UAVs in various civilian applications and the associated challenges [1]. Also, there are more applications. For instance, Khosravi and Pishro-Nik designed UAVs for Package Delivery and Network Coverage tasks [2]. Others focused on the monitoring and assessment of agriculture, such as Ju and So, who developed a system using multiple UAVs with a distributed control algorithm for system performance in the agricultural environment [3].

However, this rise in the use of UAVs has also raised significant security concerns, as UAVs can be used maliciously. In recent years, the main issues that can arise concerning the use of UAVs have been analyzed. For instance, the use of UAVs for surveillance purposes can pose risks to individual and intellectual privacy. Some of the malicious uses of UAVs are the espionage of essential and famous individuals and industrial espionage against companies, as explained by Lappa et al., who analyzed the perceptions of forty-three security specialists on the risks and threats to privacy associated with the use of

drones for surveillance [4]. Furthermore, there are other threats associated with UAV use, such as smuggling or even terrorist attacks, as indicated by Yaacoub et al., who analyzed the emerging threats of drone use in various attacks and countermeasures to prevent them [5]. For this reason, in the face of this emerging threat, the detection and neutralization of UAVs have become a priority for governments and security organizations worldwide.

The detection of UAVs is a complex challenge due to the diversity in their size, shape, and modes of operation; each UAV can have a different appearance and methods of functioning and communication, making it challenging to develop a valid and effective method for detecting any type of UAV. For this reason, Khan et al. provided a detailed review and critical analysis of existing techniques for the detection of unauthorized UAVs, such as acoustic detection, detection through computer vision techniques, detection based on RF signals, or the use of radar, pointing out their possible advantages and limitations [6]. Each method has its advantages and disadvantages in detection, and most involve using neural networks to identify the presence of UAVs based on the features extracted by each technique.

Regarding acoustic detection, this method is based on the identification of sound characteristics produced by UAVs when they are in operation, mainly due to the propellers' movement and the motors' noise. Shi et al. developed a detection algorithm using two arrays of four microphones arranged in a tetrahedral shape which could detect and locate amateur UAVs utilizing a time delay of arrival (TDOA) algorithm and a support vector machine (SVM) as a neural network [7]. Acoustic methods are instrumental in environments where UAVs are not visible, although their effectiveness can be affected by background noise and weather conditions.

Radar detection involves emitting signals that bounce off the UAV and return to the receiver to estimate the UAV's distance, speed, or shape. In [8], Semkin et al. implemented a detection and classification method based on radar cross-section (RCS) signatures. At the same time, Ezuma et al. developed a system to detect micro-UAVs using a frequency-modulated continuous-wave (FMCW) radar along with a constant false alarm rate (CFAR) detector with adaptive thresholding to detect the presence of UAVs [9].

RF-signal-based detection involves analyzing the communication signals that UAVs emit and receive during their operation by capturing the RF transmissions between the UAV and its controller. Machine learning techniques are typically used to classify and identify whether the signals are coming from a UAV. An example is the study by Kaplan et al., which detected the presence of UAVs communicating using the FHSS technique in the 2.4 GHz band. This was performed through an initial cyclostationarity method that reduces false positives and a variant of the STFT that extracts the necessary parameters to be fed into the classification algorithms [10].

Lastly, UAV detection using computer vision involves using cameras and advanced algorithms to identify UAVs in real-time images and videos. These computer vision systems are mainly based on convolutional neural networks (CNNs) that are trained with large datasets containing images of UAVs under different conditions. Park et al. analyzed six object detection models to evaluate the accuracy of each in detecting UAVs using a PTZ camera [11].

After conducting an in-depth analysis of each of the existing techniques for UAV detection, it was concluded that none of them is perfect, as they all have some limitations. However, computer vision techniques and RF signal analysis techniques offer the best performance, making them the most effective for UAV detection.

Furthermore, it was thought that, since a single technique may have limitations, using both methods together could enhance the system's effectiveness. This is because the limitations of one process can be compensated for by the other, resulting in a final system with high performance capabilities for UAV detection.

Therefore, this article proposes a hybrid approach integrating a radio spectrum analysis module employing autoencoders for UAV detection and a computer vision module based on the YOLOv10 (You Only Look Once) algorithm. Additionally, a localization

algorithm is developed, where, when two fixed cameras detect a UAV, they indicate the exact position to the PTZ mobile camera, along with a tracking algorithm that allows the PTZ camera to follow the UAV continuously.

The rest of the paper is organized as follows: Section 2 reviews the state of the art of the technologies used in this paper. Section 3 delves into the methodology used to develop the UAV detection system, explaining the system's logical and physical architecture and the algorithms designed for the two modules. Section 4 contains the tests conducted and an analysis of the results, and, finally, Section 5 presents the conclusions and future directions.

2. Related Work

The algorithms used for UAV detection, whether through RF signals or computer vision, have evolved over the years due to the specialization and development of UAVs, which necessitate increasingly novel, effective, and robust algorithms and systems to counter the threat of UAVs being used for malicious purposes.

Regarding the algorithms used for detection through computer vision, object detection algorithms are primarily employed, as these techniques use neural networks to locate and classify objects in an image. This is the goal when using cameras: to identify UAVs from real-time video feeds. These algorithms have continuously improved, always building upon CNNs, which are networks capable of recognizing and learning features and patterns in images [12].

Initially, two-stage detector algorithms were used, including the R-CNN, which has three steps: feature extraction by generating different regions to be analyzed in the image, running a CNN on each of these regions, and finally applying an SVM and a linear regressor to classify the object [13]. An evolution of this algorithm is the Faster R-CNN, which is faster because it generates regions more efficiently by introducing the region proposal network (RPN) and unifying the algorithm's stages [14].

Another object detection technique involves one-stage detectors, which are simpler and faster than two-stage detectors while maintaining a high level of effectiveness and accuracy, making them more suitable for real-time applications. Among them is the Single Shot Multibox Detector (SSD), presented by Liu et al., which achieves very good results in terms of precision for small objects by performing localization and classification tasks in a single step [15].

Finally, the most advanced object detection algorithm currently, also a one-stage detector, is the You Only Look Once (YOLO) algorithm, an algorithm introduced by Redmon et al. where the image is divided into different regions, and bounding boxes are predicted in each region along with their probabilities [16].

The YOLO algorithm has continuously advanced with each version, enhancing detection performance, speed, and accuracy. These improvements have proven valuable in UAV detection applications across different scenarios [17–19].

The version used in this paper, YOLOv10, introduces two major advancements compared to previous versions: a Consistent Dual Assignments mechanism for NMS-free training and an Efficiency–Accuracy-Driven Model Design. These innovations include a lightweight classification head, spatial-channel decoupled downsampling, and rank-guided block design, which enhance both accuracy and efficiency, as explained by Wang et al. [20].

YOLOv10 demonstrates superior accuracy and efficiency compared to previous YOLO versions and other advanced models, making it highly suitable for real-time detection tasks. As demonstrated by Wang et al. in his article, YOLOv10 achieves the highest accuracy (COCO AP%) with lower latency, which underscores its advantages in both precision and speed for real-time applications.

When it comes to tracking a detected object, Chen et al. presented an algorithm for the dynamic tracking of UAVs using deep learning techniques. Once the object is identified with YOLOv3, a PTZ camera follows the UAV based on its relative position, the PTZ angle, and the coordinates of its position in the image [21].

On the other hand, Sie et al. performed tracking using motion detection techniques with a Kernelized Correlation Filter (KCF) and Discriminative Correlation Filter (DCF). They also achieved 3D localization of a UAV by triangulating its position from the images captured from different viewpoints, successfully determining the object's spatial coordinates [22].

In terms of identification through radio spectrum analysis techniques, there are several methodologies and tools employed, as well as a variety of neural network algorithms, for UAV detection, each with specific characteristics and limitations. Ezuma et al. developed a system capable of detecting and classifying up to 17 types of UAVs in environments with interfering Wi-Fi and Bluetooth signals using a multi-stage detection approach that distinguishes between UAV and interfering device signals and performs classification using neighborhood component analysis (NCA) [23].

Nemer et al. designed a system that employs four hierarchically organized classifiers. In this system, the first classifier is responsible for identifying the UAV presence, the second classifier determines its type, and the last two classifiers are dedicated to identifying the flight mode of the UAV [24]. For their part, AlKhonaini et al. implemented a similar hierarchical classification approach, but used a reinforcement learning algorithm which applies Monte Carlo sampling to estimate the policy gradient, thus achieving an improvement in detection and classification accuracy compared to the previous study [25].

In the field of radio fingerprinting detection, software-defined radio (SDR) is common and frequently used, as it allows the processing of radio-frequency signals through software, without requiring hardware modifications. Xiang et al. proposed a low-cost solution for UAV detection which combines an SDR with an antenna array. Through an amplitude comparison and least squares search method, they succeeded in estimating the angle of the incident signal, although the system had limitations in urban environments due to signal interference [26]. Flak developed a continuous monitoring system in the 2.400 GHz to 2.483 GHz range using a LimeSDR-USB board, which allows the detection of UAV communication signals by distinguishing specific features [27].

The use of autoencoders has also been presented in the literature for UAV detection, as it is an appropriate method for anomaly detection. Jiang et al. developed a novel detection system based on a spatial-temporal autoencoder using a temporal convolution network (TCN) and CNN along with a dynamic threshold that indicates the presence of anomalies based on the correlations of the flight data [28].

Basak et al. developed an autoencoder-based system called DEFEND, which uses RF signals obtained through an SDR as input data and performs two tasks: classifying signals that are known to the system and detecting those that are not. The novel signals, after being detected, are clustered based on the extracted features to become signals known to the system [29].

Contribution of This Paper

The proposed system in this article presents a hybrid approach for detecting intrusive UAVs, combining two complementary detection techniques to ensure effective and rapid detection. The employed techniques are the detection of radio signals produced between the UAV and its controller using autoencoders that can identify anomalies in the analyzed frequency bands, and visual detection through a camera system with YOLOv10, which can locate the UAV in 3D and track it using an autonomous PTZ camera. In summary, the main contributions of this work are as follows:

1. Development of a detection system using autoencoders based on radio signals that can operate in various frequency bands up to 100 MHz of bandwidth, with these bands being selectable to monitor those most likely to be used by intrusive UAVs. Moreover, the autoencoders do not require a database with UAV signals; they only need samples of the frequency bands to be analyzed when there are no UAVs present for training, which simplifies the creation of the database as it does not require data labeling or samples with an operating UAV.

2. Development of a scalable system with fixed cameras plus an autonomous PTZ camera, where all cameras perform detection with the YOLOv10 algorithm trained on a dataset of over 60,000 images obtained from various online sources.
3. Implementation of a triangulation algorithm with fixed cameras for 3D localization of the UAV to provide the exact position to the PTZ camera, in addition to a tracking algorithm through which the PTZ camera continuously follows the UAV once detected, keeping it always at the center of the camera's image to prevent it from being lost.
4. Implementation of the complete system in which both the SDR module and the camera module are in constant communication to indicate when either of them detects a UAV. There is also communication using the Contact ID and SIA protocol with an external alarm-receiving center, which is responsible for deciding the course of action when an intruding UAV is detected.

3. Methodology

This part of the paper will discuss the algorithms designed for the implementation of a hybrid drone detection system based on radio spectrum analysis, computer vision techniques, and localization and tracking procedures. Initially, the hardware used to construct the integral system will be described.

3.1. Hardware

The developed system is based on two computers that are responsible for controlling and managing each module. One controls the SDR for the analysis of RF signals. At the same time, the other is responsible for the computer vision block from the images of the various cameras. A router is needed to connect the system components via Ethernet for communication. This allows communication between the computers through sockets and between cameras through IP addresses, ensuring all devices are on the same local network.

Figure 1 shows all the devices that make up the system. On the left side is the detection module that uses the RF signals, consisting of the SDR and a PC inside an IP67 protection box located next to the cameras, and, on the right side of the figure, is the computer vision module. The router is the central element through which all devices are interconnected via IP addressing or being physically connected via Ethernet. Likewise, the router also has Internet access to communicate with the alarm-receiving center (ARC), which is located in the security company's facilities.

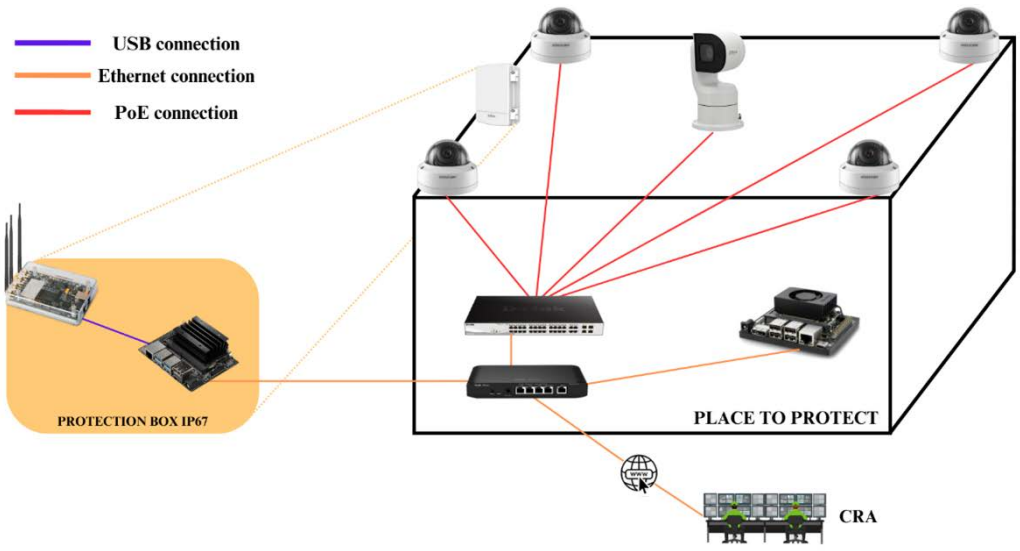


Figure 1. Physical architecture of the system, where the colour of each arrow indicates the type of physical connection between the devices.

The four fixed cameras are the Hikvision DS-2CD2743G0-IZS model (Hikvision Digital Technology Co., Ltd., Hangzhou, China). In contrast, the PTZ camera is the Dahua PTZ1A225-HNR-XA (Zhejiang Dahua Technology Co., Ltd., Hangzhou, China), which is a camera with a 2 MP CMOS sensor capable of 360° horizontal rotation and 180° vertical tilt, featuring a 25× optical zoom. Both camera models have a minimum IP66 rating, making them suitable for outdoor installation, and they support PoE+.

The SDR used for receiving the RF signals is the bladeRF 2.0 micro xA4 (Nuand LLC, Rochester, NY, USA), which offers a frequency range from 47 MHz to 6 GHz and a sampling rate of 61.44 MHz, expandable up to 122.88 MHz, with 2 × 2 MIMO. It has an integrated Analog Devices AD9361 transceiver (Analog Devices, Inc., Wilmington, MA, USA) and an Altera Cyclone V FPGA (Altera Corporation, ahora parte de Intel Corporation, San Jose, CA, USA), enabling real-time transmission and reception for advanced applications. It is physically connected to the computer via USB 3.0.

The two computers used are from the NVIDIA Jetson family and are both powerful platforms specialized in AI development. The NVIDIA device responsible for the SDR module is the Jetson Nano Developer Kit (NVIDIA Corporation, Santa Clara, CA, USA), while the one for the cameras is the Jetson Orin Nano Developer Kit (NVIDIA Corporation, Santa Clara, CA, USA). The latter is considerably more powerful than the former, as the computational cost in computer vision applications is high, requiring a device capable of managing five cameras using YOLOv10.

Finally, to perform IP addressing for the various components, a RG-EG105G-V2 (Ruijie Networks Co., Ltd., Fuzhou, China) router is used, connecting both computers and the network cable that provides Internet to communicate with the ARC over the network. Additionally, the PFS3218-16GT-135 PoE (Zhejiang Dahua Technology Co., Ltd., Hangzhou, China) switch is used to connect all the cameras as they are all PoE.

3.2. Logic Architecture

Before delving into the development of the algorithms for both modules, it is important to understand the general logic of the system and how both modules are related and communicate with each other.

Initially, the system was designed without fixed cameras, only a PTZ camera, so the SDR module was the first step in detection. When it detected a signal that could be coming from a UAV, it would alert the PTZ camera module, which would then begin sweeping the entire area until the UAV was located.

However, the system has been improved with the addition of fixed cameras. Now, when two fixed cameras simultaneously detect the UAV, they send the exact position to the PTZ camera, and, once it detects the UAV, it begins to track it. The main objective of the system is for the PTZ camera to identify the UAV and continuously follow it, allowing the security company using this system to always know the UAV's location. For this reason, the PTZ camera can detect the UAV through two methods:

1. Through sweeping. This situation occurs either when the SDR module detects the UAV or when it is detected by only one fixed camera. In both cases, the UAV's position is unknown, so the PTZ camera begins to sweep its surroundings to detect it.
2. After being located by two fixed cameras. If two or more fixed cameras detect the UAV, they send its position directly to the PTZ camera, which then moves directly to the UAV's location without needing to sweep.

Furthermore, once the PTZ camera locates the UAV, it tracks it. However, if the UAV disappears from the camera's view at any point because it has flown away or moved too quickly for the camera to follow, the PTZ camera begins a post-loss sweeping period to relocate it. If the UAV is not found after the sweeping period and no alarm is received from another camera or the SDR, the UAV is considered to have left the area.

Finally, when the UAV is detected, an alert is sent to the security company's ARC so they are aware of the presence of a UAV in the protected area. Once the UAV is located and

tracked by the PTZ camera, videos of the UAV will be periodically sent to the ARC, along with a message when the UAV has disappeared.

Communication with the ARC is conducted through the Data Transport Layer Security (DTLS) protocol, which creates a secure and robust communication link between the detection system, acting as the client, and the ARC, acting as the server. The alarms and videos themselves are sent using the SIA DC-09-2021 protocol (Security Industry Association, Silver Spring, MD, USA). On the other hand, communication between the two modules is carried out via sockets programmed in Python, just like the rest of the system.

3.3. SDR Module

In the development of the algorithm for the detection of drones through radio signals using artificial intelligence, it was necessary to consider some conditions that have influenced the development of the algorithm and the selection of the neural networks to be used.

Firstly, the system needed to be a generalist to detect any UAV, regardless of the type, the communication and control protocols it uses, or the frequency bands it operates on. This is because the various UAV detection algorithms in the literature focus on detecting UAVs that use only specific protocols or detect only a single frequency band.

Secondly, the system must be plug and play, meaning its installation should not require testing or prior training at the deployment site. This implies that the selected detection technique should not require an extensive and varied database to be created at the installation site, as would be necessary with supervised methods such as CNNs. These methods require collecting specific spectral data under different environmental conditions and with different UAV models.

For these reasons, autoencoders were chosen. These constitute an unsupervised learning technique that is trained only with unlabeled spectrum samples when no UAVs are present. This means there is no need to manually generate a database at the deployment site, ensuring the system meets the plug-and-play condition. Moreover, autoencoders allow for the creation of a general algorithm, which will be explained later, that is capable of detecting all types of UAVs as required.

The developed algorithm consists of two parts that run in parallel. The first part is responsible for receiving and processing the RF signals obtained with the SDR, and the second part analyzes the samples obtained to identify using the autoencoders whether there is an intruding UAV based on the radio signals.

3.3.1. Reception and Processing of Data from the SDR

The first part is based on GNU Radio, and its purpose is to capture signals from different frequency bands, calculate their power, and store them in text files, which will be used in the second block for detection. Figure 2 shows the structure followed in GNU Radio, consisting of four GNU Radio blocks. The first two blocks are native to the software, while the subsequent two have been created based on the project's requirements.

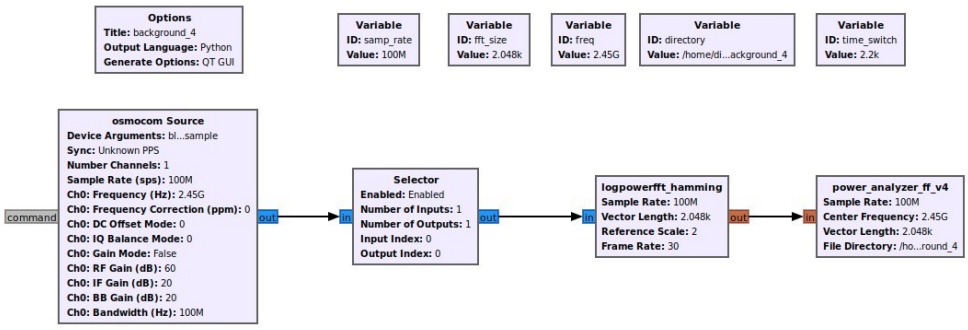


Figure 2. GNU Radio schematic used to receive signals from the bladeRF SDR and process them for input to the detection system.

The first GNU block, Osmocom Source, is responsible for receiving complex signals from the bladeRF SDR, where parameters such as the bandwidth, frequency, and gain of the receiving system are set. The second block, Selector, is used to block data transmission and is only employed when changing the system's frequency. If it is not blocked, there is a risk of data being stored incorrectly.

The third block receives the complex signals from the SDR and calculates the power in dBm by applying a Hamming window. Finally, the fourth block stores the calculated power of that frequency band in a text file, creating a separate file for each analyzed band.

Thus, the system first monitors a series of frequency bands with a bandwidth of 100 MHz, then calculates the power at 2048 points within that band, and, finally, stores those values in a text file that is updated each time values for that frequency band are received.

For the tests conducted, three frequency bands commonly employed in UAV usage are analyzed: 950 MHz, 2.450 GHz, and 5.80 GHz. In this way, three different files are generated, containing various samples obtained from each of the frequencies. Each sample consists of 2048 power values, which, together, represent 100 MHz of that particular frequency band. These samples of 2048 power values are the data input to the autoencoders for detection.

3.3.2. Detecting Algorithm Using Autoencoders

This second part involves analyzing the samples obtained in the first block by inputting them into autoencoders to determine if a UAV is present in any of the analyzed frequency bands.

Autoencoders are used as anomaly detectors within the frequency bands. When the system is launched for the first time after installation, it begins collecting data from the bands to be analyzed, which will be used to train the autoencoders. These samples correspond to the signal power in the absence of a UAV, allowing the training of a specific autoencoder for each frequency band. This data collection and training process is automatic, and, once it is complete, the detection system starts, along with the camera module. If the system shuts down for any reason, it will resume directly with the detection process upon restarting as the training will already have been completed.

Since the training data correspond to the signal power in the absence of a UAV, when inference is performed on the signals constantly received from each band, if the input data do not contain UAV signals, the autoencoders, trained on such data, will reconstruct the signal with low error. However, if the input data contain UAV signals, which were not present during training, the reconstruction error will be high, indicating the possible presence of a UAV.

Figure 3 demonstrates this. In the example on the left, the spectrum of the 2.45 MHz band and the reconstruction of the band after passing through the autoencoders are shown. The reconstruction is similar to the input signal, resulting in a low error, as shown in the black spectrum, indicating no UAV in this band. In contrast, in the example on the right, the autoencoder is unable to reconstruct the input signals, resulting in a high reconstruction error, indicating the potential presence of a UAV. The figure was generated in Python 3.8 using the Matplotlib library, employing the power vectors extracted from GNU Radio and the power vectors obtained after processing them through the autoencoders.

Based on this, the system continuously analyzes each frequency band for anomalies that could indicate the presence of a UAV. An alert is sent to the camera module only if anomalies are detected in the same section of the frequency band in two consecutive readings. An anomaly is considered to be present when the reconstruction error of the signal in that part of the band exceeds a certain threshold.

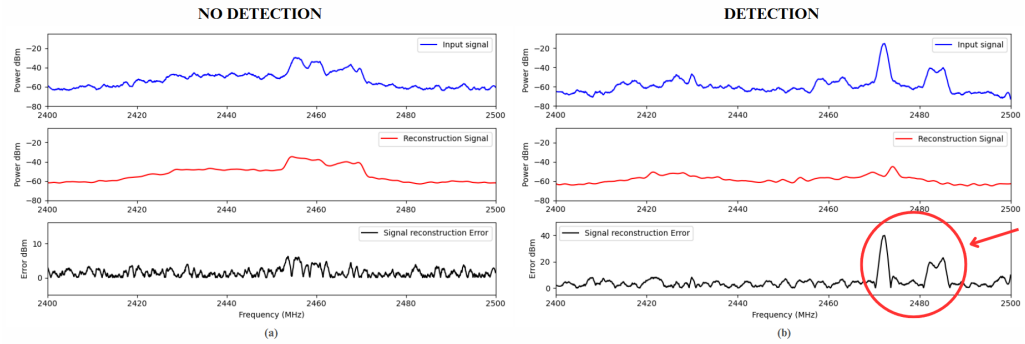


Figure 3. Examples of the performance of the autoencoder-based detection algorithm: (**a**) shows a situation of no detection as the autoencoder can reconstruct the input signal properly, (**b**) shows a detection in the part surrounded by the red circle, because the reconstruction error there is very high, as the autoencoder has not been able to reconstruct the input signal, indicating the possible presence of a UAV.

The requirement of two consecutive readings with anomalies is due to the high likelihood of false positives caused by interfering signals in the frequency bands. An anomaly detected in a single reading could result from a momentary interference and may not necessarily indicate the presence of a UAV. Therefore, by requiring anomalies to be recurrent in two consecutive readings, the possibility of false positives is reduced, improving detection accuracy.

The following part explains the steps the algorithm takes to determine if a UAV is present in a frequency band after two consecutive data readings.

1. Sample Reception: N samples of a specific frequency are received. These samples are fed into the autoencoder, producing N corresponding reconstructions.
2. Reconstruction Error Calculation: For each sample and its reconstruction, the following takes place:
   - Both the sample and its reconstruction are divided into M segments.
   - The reconstruction error is calculated for each segment individually.
   - It is checked whether the reconstruction error in any segment exceeds a predefined threshold.
   - The segments m in sample n where the error exceeds the threshold are recorded.
3. Storing Results: This process is repeated for all N samples, storing the segments where the reconstruction error exceeds the threshold for each sample.
4. Segment Analysis: For each segment, the percentage of samples that have exceeded the initial reconstruction error threshold is calculated.
5. Anomaly Detection: If the percentage in any given segment exceeds a second detection threshold, an anomaly is considered to be present in that segment of the analyzed frequency.
6. Detection Confirmation: The process is repeated when N new samples of the same frequency are received. If an anomaly is detected in the same segment of the frequency in this second iteration, the presence of a UAV is confirmed.

Figure 4 graphically illustrates step 2, "Reconstruction Error Calculation", from the previously described process, while Figure 5 shows the steps of "Storing Results", "Segment Analysis", and "Anomaly Detection".

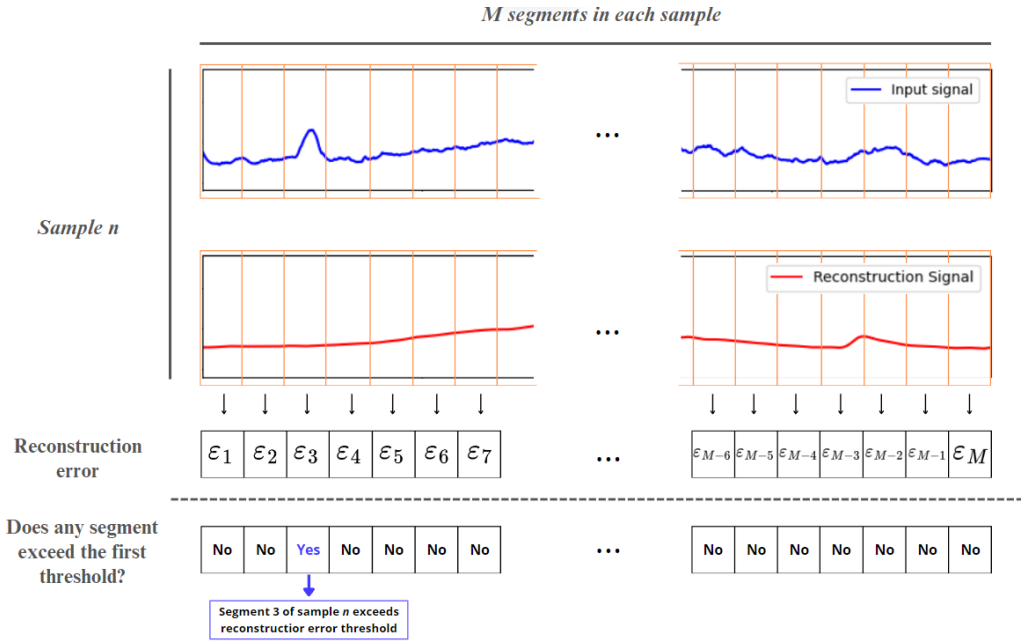


Figure 4. Reconstruction Error Calculation process performed with each sample received.

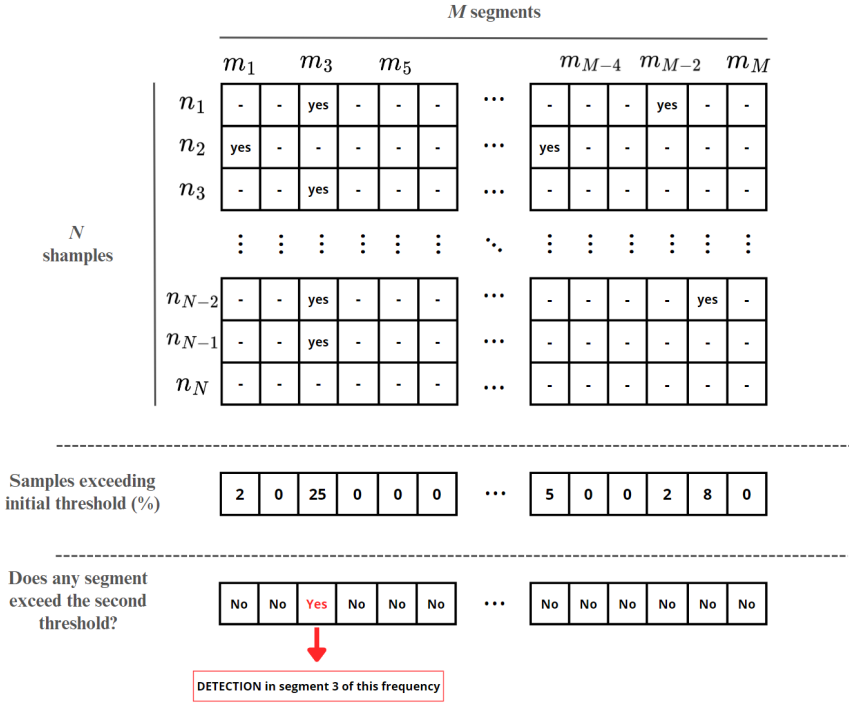


Figure 5. Steps of Storing Results, Segment Analysis, and Anomaly Detection in the autoencoder-based detection process.

Thus, autoencoders are utilized to identify the presence of potential intrusive UAVs by analyzing and comparing the RF signals received from each band with the training data of the band obtained in the absence of UAVs. This approach ensures that the system does not require the manual creation of a complex database, wherein UAVs would need to be flown in the installation area in order for UAV signals to be captured for the database. Moreover, with this approach using autoencoders, the algorithm is capable of identifying all types of UAVs.

The architecture of the developed autoencoders can be seen in Figure 6 and is composed of seven layers, where the input layer has 2048 values, corresponding to the length of the power samples, and the dimensionality is reduced to only 32 neurons. Each layer has a ReLU activation function, except for the output layer, which uses a linear activation function.

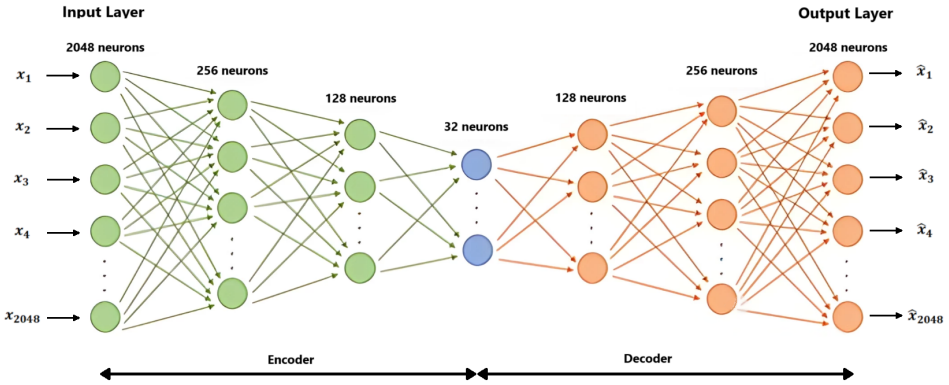


Figure 6. Architecture of autoencoders used for drone detection. The green part, enconder, represents the compression of the input data. The blue is the latent space, which is the representation of compressed data, and the orange, decoder, represents the reconstruction up to the original dimension.

For the training process, the input data are passed through a Savitzky–Golay (Savgol) filter, and the model is trained for 400 epochs using the Adam optimizer with an adaptive learning rate of 0.001 that varies based on the model's validation performance. The model's loss function is the Mean Absolute Error (MAE), which evaluates accuracy by measuring the average difference between the autoencoder outputs and the original inputs. The MAE is suitable for this model as it provides a direct measure of the magnitude of the prediction error in the same units as the original data, facilitating the use of a threshold for detection.

3.4. Computer Vision Module

3.4.1. YOLOv10

As previously explained, all the cameras in the system use YOLOv10 for UAV detection, utilizing a custom model trained with a dataset comprising the Birds vs. Drones dataset [30], Maciej and Marek UAV dataset [31], and the GSV-Cities dataset [32]. From these three sources, 60,178 images were used, divided into 48,521 for training and 11,657 for validation.

The reason for using such an extensive dataset is to represent as many real-world scenarios as possible. It includes images of UAVs of different sizes and types, at various distances and under different weather conditions, and in diverse flight environments. There are also background images and elements that could potentially confuse detection, such as birds.

The training was carried out using the YOLOv10-S model, a small version that balances speed and accuracy. It is the most powerful version that could be trained using the created dataset and the selected hyperparameters given that the computer used for training did not have sufficient computational capacity for more complex architectures.

The training was set to 400 epochs with a batch size of 32 and employed data augmentation techniques. The training process lasted approximately three days and yielded very good results when analyzing the metrics extracted by YOLOv10, as shown in Figure 7, achieving very high precision and recall values.

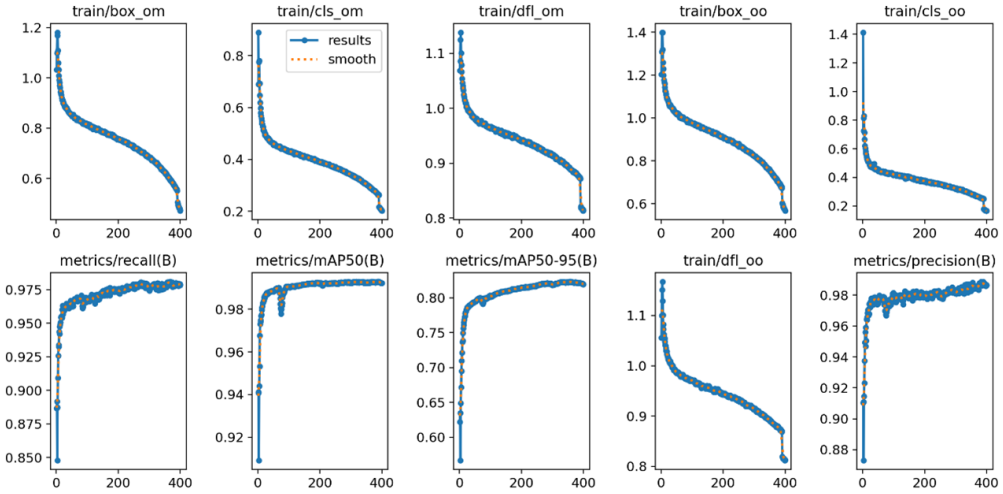


Figure 7. Results of the model trained with YOLOv10.

3.4.2. Tracking Algorithm

The UAV tracking process is conducted using the PTZ camera once the UAV is detected, automatically adjusting the camera's orientation to keep the UAV continuously located, thanks to the developed algorithm, which is based on three main functions.

The first function involves adjusting the camera's zoom to optimize the UAV's size relative to the total image frame. This is achieved by analyzing the percentage of the UAV's area compared to the total area based on the camera's current zoom, as illustrated in Figure 8, and then adjusting the zoom to ensure the UAV has a specified size in the frame.

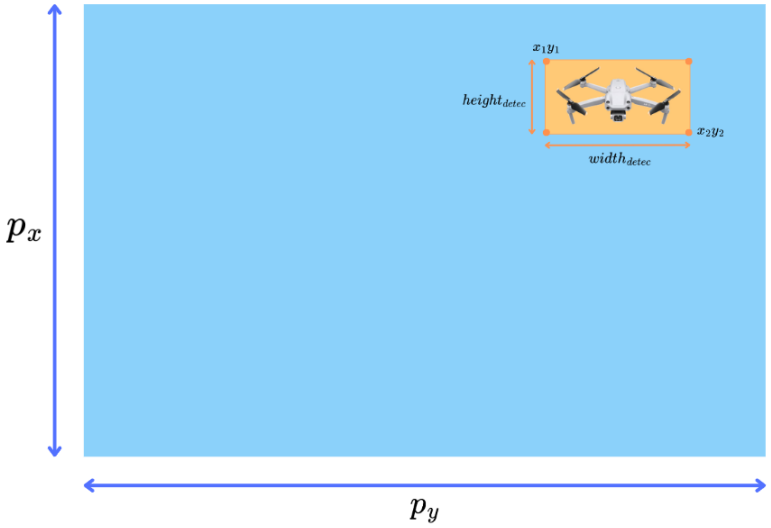


Figure 8. Parameters used for zoom adaptation from UAV detection.

The established value for proper operation is to ensure that the detection area represents between 0.5% and 1.2% of the screen area. If the UAV's area is above or below these threshold values, the camera's zoom is adjusted accordingly.

These limits ensure that the UAV is large enough to be detected by YOLOv10 while also being small enough relative to the screen to prevent it from quickly disappearing from the camera's view due to its rapid movements. If the UAV is too large on the screen, it is more likely to move out of the camera's field of view easily.

The second function is used to calculate the field of view (FoV) of the PTZ camera based on the current zoom level and knowing the camera's specifications. It first establishes a simplified linear relationship between the zoom level (0 to 25) and the focal length (4.8 mm to 120 mm) using Equation (1), which computes the focal length $f(z)$ from the zoom level z:

$$f(z) = f_{min} + (f_{max} - f_{min}) * \frac{z}{zomm_{max}} \tag{1}$$

Then, the field of view (FoV) is calculated using the focal length and the camera sensor's size, as shown in Equation (2):

$$FoV = 2 * arctan\left(\frac{d}{2 * f(z)}\right) \tag{2}$$

where d is the sensor dimension representing the horizontal FoV width and the vertical FoV height. The sensor size values are assumed to be 5.7 mm $\times$ 2.88 mm, based on the manufacturer's information.

Finally, the third function calculates the relative movement that the camera must make on the pan and tilt axes to keep the UAV at the center of the camera view. This function is based on trigonometry applied to a circle and its chord, where the chord represents the deviation in the image. The calculation is performed independently for both axes of the screen (horizontal and vertical).

First, the camera's FoV is needed from the current zoom level, calculated using the previous function, as the FoV defines the angle of view (θ) that delimits the chord used in the calculations. To simplify the trigonometric calculations, a circle with a radius of $r = 1$ is used, allowing the chord length to depend only on the angle θ. Second, it is necessary to calculate the distance from the center of the circle to the chord c_0 using Equation (3), as illustrated in Figure 9a.

$$c_0 = cos\left(\frac{\theta}{2}\right) * r \tag{3}$$

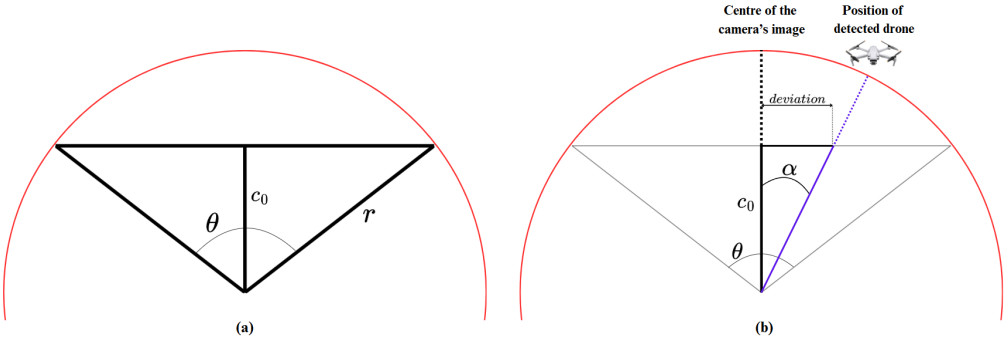


Figure 9. Representation of the trigonometry used to calculate c_0 in (**a**) and *deviation* in (**b**), where, the black dotted line refers to the centre of the image over the centre of the circle and the purple dotted line to the position where the UAV is located.

Once this value is known, any deviation of the UAV from the center of the image can be calculated, allowing us to determine α, which represents the angular correction that the camera must apply to center the UAV. The *deviation* is the difference calculated between the midpoint of the image and the center of the detection on the axis. Figure 9b illustrates how the angular correction is derived from the deviation of the UAV detection from the center, and Equation (4) provides the value:

$$\alpha = arctan\left(\frac{deviation}{c_0}\right) \tag{4}$$

Compared to existing tracking systems, which often rely on fixed zoom levels or less dynamic orientation adjustments, this algorithm provides a significant improvement in responsiveness and accuracy. Its reliance on real-time data and camera specifications allow it to operate effectively without requiring extensive computational resources, making it a robust and efficient solution for UAV tracking.

The key innovation is the dynamic zoom adjustment, which continuously analyzes the UAV's size in the frame to balance visibility and coverage, preventing the UAV from becoming too small to detect or moving out of view. Additionally, the algorithm calculates the camera's FoV in real time based on the zoom level, leveraging the camera's optical specifications to fine-tune its orientation with precision. Lastly, to ensure smooth and accurate tracking, continuous recalibration is carried out based on trigonometric angular correction.

3.4.3. Localization Algorithm

The localization of the UAV is performed using fixed cameras. If two or more cameras detect the UAV, they send its exact position to the PTZ camera using an approach based on the intersection of the detection vectors. Figure 10 shows how localization is carried out when two cameras simultaneously detect a UAV.

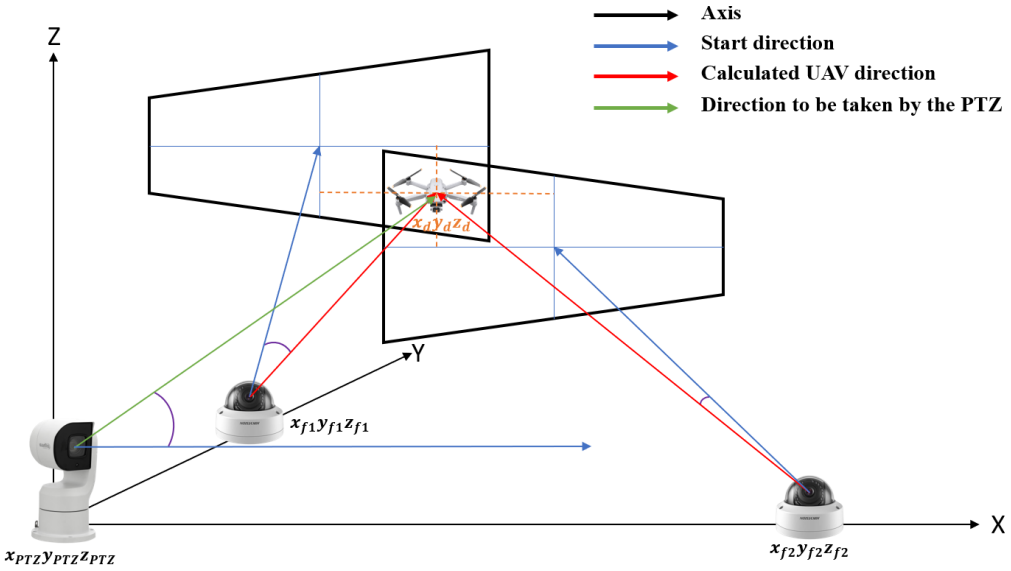


Figure 10. Schematic for UAV position estimation using two fixed cameras.

Each fixed camera has a "start direction", which is the direction in which the camera is oriented. When a UAV is detected, the "calculated UAV direction" is computed, indicating the precise orientation of the UAV relative to the camera that detected it. These calculated directions are three-dimensional vectors obtained from the start direction of the camera and its deviation to the central point of the detection box, calculating the deviation angle in terms of pan (horizontal) and tilt (vertical) within a range of 0° to 360°.

A non-linear least squares optimization method finds the lines' intersection points representing the fixed cameras' detection directions. The objective function to minimize is the sum of the squared distances from an estimated point in three-dimensional space to each of the lines defined by the detection directions of the cameras. Mathematically, this is represented by Equation (5):

$$D^2 = \left\| \vec{r}_i - \vec{p} - \left(\frac{(\vec{r}_i - \vec{p}) * \vec{d}_i}{\vec{d}_i * \vec{d}_i} \right) * \vec{d}_i \right\|^2 \tag{5}$$

$\vec{r}_i$ is the origin point of camera i, while $\vec{p}$ is the estimated point in space representing the UAV's position and $\vec{d}_i$ is the detection direction of camera i. This calculation is based on

projecting the estimated point onto each of the lines and measuring the Euclidean distance between the point and its projection.

Minimization of this function is performed using the Sequential Least Squares Programming (SLSQP) algorithm from the SciPy library. The goal is to find the values of x, y, z that minimize the following function (6):

$$\min_{\vec{p}} \sum_{i} D_i^2 \tag{6}$$

Once the UAV's position is estimated, the vector connecting the PTZ camera's position to the UAV's position is calculated. This vector is then transformed into pan and tilt units using Equations (7) and (8), which normalize the calculated angles to an appropriate range for controlling the PTZ camera:

$$pan = \frac{atan2(y, x)}{\pi} \tag{7}$$

$$tilt = \frac{arcsin(z)}{2\pi} \tag{8}$$

The innovation in the localization method using fixed cameras lies in its combination of simplicity and precision in determining the UAV's position. Each camera computes the UAV's orientation based on deviations from its initial direction, resulting in three-dimensional vectors. The intersection of these vectors from multiple cameras is then calculated using a non-linear least squares optimization method. This approach minimizes the Euclidean distance between the estimated point and the vectors, ensuring precise localization.

This method enhances accuracy without requiring complex hardware setups like multiple PTZ cameras. Additionally, it operates in real time and integrates seamlessly with the PTZ camera, directly providing positional data to adjust its focus and tracking. The novelty also lies in its robust mathematical foundation, which employs the SLSQP algorithm, ensuring efficient and accurate computation, even in dynamically changing scenarios.

4. Results Discussion

To evaluate the performance of the developed algorithms and the overall system, various tests were conducted to cover as many scenarios as possible. These tests were performed in different spaces using different types of UAVs and at various times of the day to assess the response of the SDR module and the cameras to the presence of an intrusive UAV, thereby determining the system's effectiveness in terms of detection.

The testing process was also crucial for analyzing the system's limitations and identifying which parameters or situations lead to successful or unsuccessful detection. Factors such as the distance to the UAV, its speed, and the electromagnetic interference in the testing environment were considered.

In most of the tests conducted, the results were analyzed separately, examining the performance of each module individually, as communication between the two modules has never posed a problem. The primary challenge of the project was to ensure that both the SDR module with the autoencoders and the computer vision module accurately detect UAVs, as well as perform localization and tracking functions appropriately.

Each module presented its challenges. For radio-frequency signals, it was necessary to determine whether the algorithm based on autoencoders is effective and how parameters such as the distance of the UAV, which affects the signal strength received by the SDR, impact performance. Additionally, it was important to examine if the algorithm's effectiveness is compromised when operating in environments with numerous interfering signals in the frequency bands being analyzed.

For the camera module, the main challenges were related to assessing the detection performance based on the UAV's distance and speed, determining whether the localization system can correctly triangulate the UAV's position and send the precise location to the PTZ camera, and, finally, evaluating if the PTZ camera can successfully track the UAV.

4.1. SDR Module Results

The first and primary aspect to analyze in the behavior of the SDR module's algorithm was its ability to detect signals from intrusive UAVs. For this purpose, tests were conducted in different environments where a UAV was flown at various distances to determine if the system can indicate the presence of a UAV.

Figure 11 shows the results of all the tests conducted, illustrating the detection success rate as a function of the distance to the UAV. The success rate is simply the percentage of times the UAV was detected by the algorithm out of the total number of times the UAV was flown at that distance.

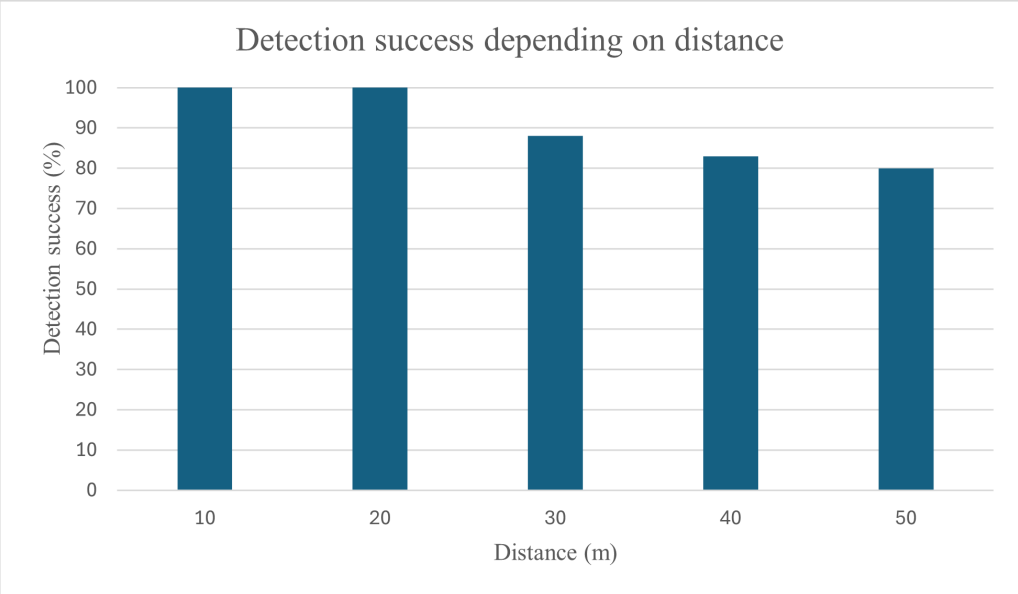


Figure 11. Detection test results using the SDR module as a function of distance from the UAV.

The results show that the UAV detection system achieved a performance of practically 100% at distances of 10 and 20 m, indicating high accuracy and reliability at close ranges. However, as the distance increased, the detection efficiency decreased. This trend can be explained by the signal strength received by the SDR; the further away the UAV is, the lower the received power, making it more challenging for the algorithm to identify anomalies in the received signals.

Nonetheless, the results for these longer distances are still favorable, with successful detection rates of 89% and 83% at distances of 30 and 40 m, respectively, dropping to the lowest value of 80% at a distance of 50 m.

It is important to note that the effectiveness of the algorithm is significantly influenced by the dataset used to train the autoencoders. Generally, the larger the dataset, the better the detection algorithm will perform, as it will have been trained with a greater number of samples. This is particularly crucial in highly saturated bands like those used by Wi-Fi, where there are many interfering signals. A more extensive dataset would include more samples of all these non-UAV signals.

Ideally, when installing the system, the algorithm should collect data from the spectrum of each band for an entire day. This would provide samples of the radio spectrum's state in each band when no UAV is present at different times of the day, which is important as the signals present may vary throughout the day. For these tests, the training was

conducted with data collected over periods of less than two hours, so, with more data, detection could be more effective.

The second aspect analyzed was the average time taken to detect the UAV as a function of distance. As mentioned earlier, greater distances can make detection more challenging due to the weaker signal strength received, which also results in increased detection time.

Figure 12 illustrates this situation, clearly showing that, as the distance increases, the time required for the algorithm to identify the presence of the UAV also increases.

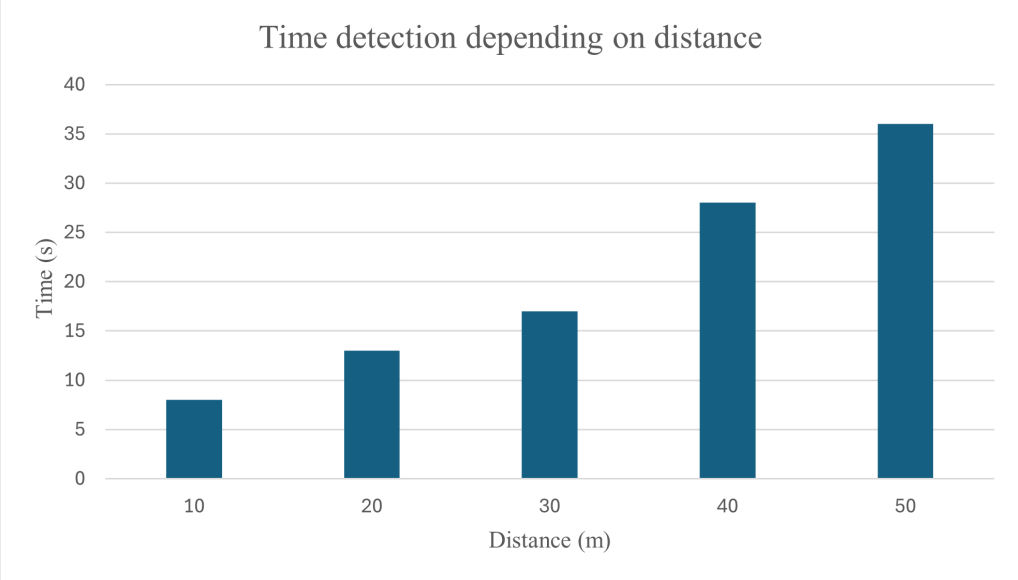


Figure 12. Time required for UAV detection as a function of distance.

At a distance of 10 m, the detection time was minimal because, as previously explained, determining the presence of a UAV requires detecting an anomaly in the same part of the frequency band in two consecutive readings of the same band. Therefore, the time needed to perform two readings of the same band is approximately what was observed for the 10 m distance.

As the distance increases, the detection time also increases because the UAV is not always detected in two consecutive readings. If the UAV is not detected in one reading, at least two more readings are needed to make the detection, thereby increasing the time required. The greater the distance, the weaker the signal received, making it more likely that detecting the UAV will take longer.

This approach is designed to minimize the number of false positives that the system might produce. By requiring two consecutive readings with anomalies in the same part of the band, the system nearly eliminates false positives, though it makes detection more challenging.

During testing, the algorithm's resistance to false positives was also analyzed by running the system to monitor the frequency bands without flying any UAVs, to check if the UAV alarm would trigger when there was none. The results were highly positive, as very few false positives occurred. The few false positives observed happened in environments with a lot of interference, such as an office, where the presence of numerous wireless devices increased the likelihood of false positives. Nevertheless, the system still proved to be quite robust.

4.2. Computer Vision Module Results

The first aspect to analyze was whether the trained model detects UAVs effectively. In the YOLOv10 training graphs, the results appeared highly promising, but these were obtained using static images. To determine if the model truly performs well for the

developed system, it was essential to test it with real-time images in environments where the system could be deployed.

For this reason, Figure 13 illustrates the success rate when a UAV was flown at different distances from the camera. The goal was to verify whether the model detects UAVs properly and to analyze how distance affects detection.

As can be seen from the results, the overall detection performance was very good. The detection rate was always above 75% for distances up to 50 m. At closer distances, the UAV was always detected, with the percentage decreasing as the distance increased, but strong results were maintained throughout.

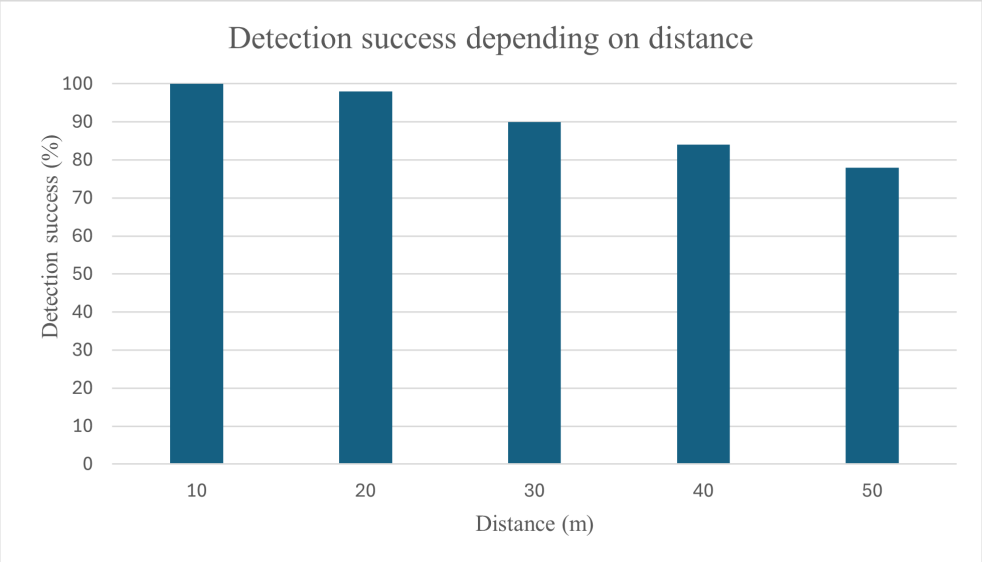


Figure 13. Detection test results using the computer vision module as a function of distance from the UAV.

It is important to note that the results for this parameter, as well as for the others, were obtained from performing different tests in various environments to ensure the reliability of the results. This prevented skewed data that might arise from testing in the same location repeatedly.

It is also important to indicate that the tests were conducted under conditions where the UAV could be detected, avoiding flights in darkness or in scenarios with no visibility where even a human observer would not be able to identify the UAV. Nonetheless, the results were very positive considering the characteristics of the UAVs used, which were small devices that can easily blend in with the environment.

The next aspect to analyze was the effectiveness of the tracking algorithm, evaluating whether the PTZ camera can continuously follow the UAV as it moves through the environment. Once again, the impact of distance on the tracking process was analyzed.

Figure 14 presents the results of the tracking algorithm. As can be observed, the success rate no longer followed the same trend with respect to distance. The best results were obtained at intermediate distances, with the tracking success rate being above 90% for distances of 20 and 30 m and around 83% for 40 m.

The poorest results were obtained at the shortest and longest distances, but both cases have a clear explanation. At shorter distances, the quality of tracking decreases because the angular velocity needed to follow the same movement of the UAV is greater compared to longer distances. A linear displacement of the UAV at close range results in a more significant angular change for the camera, which can make tracking more challenging.

At a distance of 50 m, the tracking effectiveness decreased because the model struggled more to detect the UAV, as seen in Figure 14. At that distance, the detection of the UAV by the camera was not consistent, which affected the tracking algorithm's performance. When

the UAV is not detected, the algorithm lacks the necessary visual reference, reducing the smoothness of the tracking.

Consequently, it can be concluded that the tracking algorithm functions optimally as long as the UAV is detected. The reduction in the tracking rate at greater distances is primarily due to the loss of continuous detection of the UAV, indicating that the smoothness of tracking heavily relies on the model's ability to consistently detect the UAV.

Lastly, in terms of localization, the goal was to verify if the algorithm can accurately send the exact position of the UAV to the PTZ camera. For these tests, the distance was not a factor to consider since, if the algorithm is correctly developed, once the UAV is detected by the fixed cameras, the position will be triangulated independently of the distance.

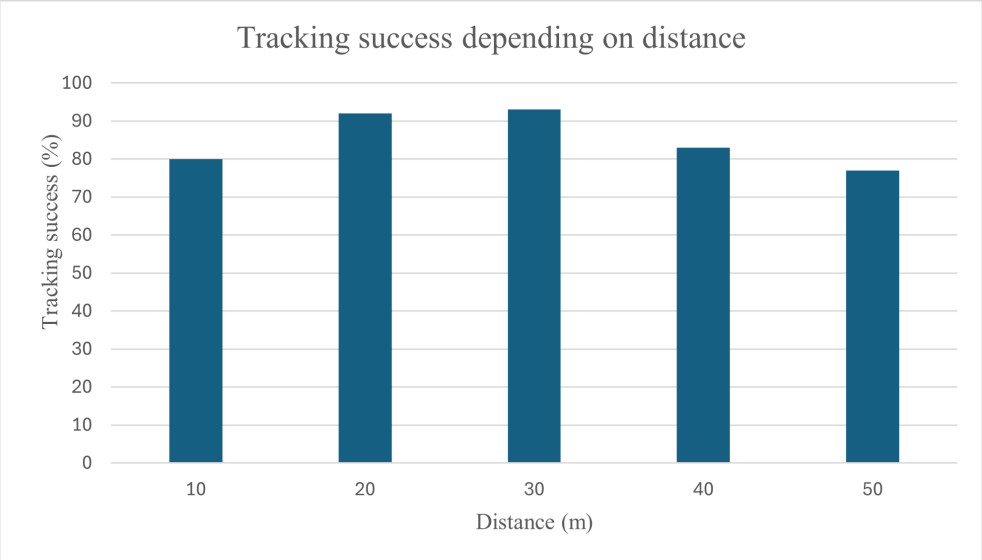


Figure 14. Tracking test results using the computer vision module as a function of distance from the UAV.

The only aspect affected by distance is detection, as the UAV must be detected by at least two cameras to be localized. Thus, at very long distances, it might be more challenging for both cameras to locate the UAV. However, as soon as the UAV is detected simultaneously by both cameras, they will send the position.

In the various tests conducted, it was demonstrated that the localization process works correctly. In all cases where the UAV was detected by two cameras, the PTZ camera moved to the UAV's position and was able to detect it.

Figure 15 provides an example of what was displayed on the NVIDIA Jetson Orin during the localization tests. As shown in Figure 15a, the two fixed cameras simultaneously detected the UAV while the PTZ camera was pointed in a different direction. Figure 15b shows the frame where the PTZ camera rotated to the exact position of the UAV after triangulating its position. The time elapsed between both moments corresponded to the time it took for the camera to complete the rotation, which was less than one second.

It is essential to highlight that, for localization to be effective and accurate, the calibration of the cameras' positions and orientations must be precise. When installing the system, several parameters must be entered into the system before it is launched, including the exact positions of the cameras and the direction in which their lenses are pointing. If these parameters are not accurately input, the localization process will fail because the algorithm will triangulate the UAV's position based on incorrect camera positions.



Figure 15. Vision of the cameras in the localization tests. In (**a**), it is shown that the fixed cameras simultaneously detected the UAV, while in (**b**) it is shown that the PTZ camera had already located the UAV after being provided with its exact position.

5. Conclusions

Throughout this article, we have thoroughly explained the two UAV detection methods developed as a potential solution to the growing security concerns stemming from the misuse of UAVs in recent times.

The two developed methods are designed to complement each other, creating a more effective and comprehensive detection system using artificial intelligence algorithms, making this system a novel and intelligent solution. We designed the first detection method based on radio-frequency signals, which are captured and processed by the SDR BladeRF 2.0 micro xa4 (Nuand LLC, Rochester, NY, USA), along with an NVIDIA Jetson Nano Developer Kit (NVIDIA Corporation, Santa Clara, CA, USA) responsible for running the detection algorithm.

The developed algorithm is based on autoencoders, which, as discussed, are unsupervised neural networks frequently used for anomaly detection. In this project, multiple autoencoders were trained, each for a specific frequency band to be analyzed. The input data consisted of 2048 power values that made up the 100 MHz bandwidth of each band.

A validation system was developed based on the autoencoders, in which a set of conditions must be met to confirm the presence of a UAV, aiming to balance detection time with a reduction in false positives.

The second method involves a module comprising four fixed cameras and a PTZ camera, connected via IP addresses to an NVIDIA Jetson Orin Nano responsible for managing them. Each camera detects UAVs using a custom-trained YOLOv10 model with over 60,000 images, showing excellent performance in detecting UAVs even at long distances.

The computer vision module also includes the function of tracking the UAV continuously. An algorithm was developed for tracking, ensuring the PTZ camera keeps the UAV centered in the image. Additionally, a localization system for the UAV was implemented, allowing the PTZ camera to track the UAV once two or more fixed cameras simultaneously detected it, calculating the exact position and sending it to the PTZ for tracking.

Both modules are interconnected and communicate in real time to ensure UAV detection. The developed system also includes a communication protocol using SIA DC-09-2021 (Security Industry Association, Silver Spring, MD, USA), allowing the system to alert a remote monitoring station (CRA) when a UAV is detected, and periodically sending videos that enable them to monitor the UAV's presence and take appropriate actions.

The results indicate that both modules performed well in terms of detection, achieving high success rates even at distances beyond 40 m. The system's primary goals are to detect efficiently while minimizing false positives. This is achieved in the camera module by implementing techniques that filter out a significant portion of false positives. In the SDR module, with an adequate database, the percentage of false positives remains minimal.

Both localization and tracking algorithms functioned appropriately, accurately locating the UAV when it was detected by two cameras, and the tracking algorithm successfully maintained continuous UAV tracking.

Future improvements could mainly focus on expanding the datasets for both modules and upgrading the hardware to more powerful devices. With a more extensive and varied database, YOLOv10 detection could be further improved, enhancing accuracy and reducing false positives. In the SDR module, a more comprehensive initial dataset of non-UAV signals used to train the autoencoders would result in better signal reconstruction when no UAV is present, further improving detection accuracy.

Regarding hardware, this has been the main limitation of the system, as both NVIDIA devices were operating at their maximum capacity. With a more powerful device for the SDR module, more frequency bands could be analyzed, and more samples could be processed per band, significantly improving system performance. In the camera module, a more powerful device would allow for smoother camera operation, better PTZ tracking of the UAV, and the ability to incorporate more fixed cameras.

In this sense, the Research and Development department of the Simetria Group will work on the improvement points indicated above to optimize the performance of the system and broaden the detection spectrum.

Author Contributions: Conceptualization, J.F.M. and J.L.C.; methodology, P.L.-M., F.J.A. and C.A.; software, P.L.-M. and L.G.S.F.; supervision, J.F.M. and C.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Red.Es.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors Christian Abarca, Francisco José Alegre and Jose Luis Calle are employed by Grupo Simetria. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]
2. Khosravi, M.; Pishro-Nik, H. Unmanned Aerial Vehicles for Package Delivery and Network Coverage. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–5.

3. Ju, C.; Son, H.I. Multiple UAV Systems for Agricultural Applications: Control, Implementation, and Evaluation. *Electronics* **2018**, *7*, 162. [CrossRef]
4. Lappa, D.; Fessakis, G.; Karampelas, P. Recognizing the Threats of Drone Surveillance. A Case Study. In Proceedings of the 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), Nafplio, Greece, 26–29 June 2022; pp. 1–5.
5. Yaacoub, J.P.; Noura, H.; Salman, O.; Chehab, A. Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet Things* **2020**, *11*, 100218. [CrossRef] [PubMed]
6. Khan, M.A.; Menouar, H.; Eldeeb, A.; Abu-Dayya, A.; Salim, F.D. On the Detection of Unauthorized Drones—Techniques and Future Perspectives: A Review. *IEEE Sens. J.* **2022**, *22*, 11439–11455. [CrossRef]
7. Shi, Z.; Chang, X.; Yang, C.; Wu, Z.; Wu, J. An Acoustic-Based Surveillance System for Amateur Drones Detection and Localization. *IEEE Trans. Veh. Technol.* **2020**, *69*, 2731–2739. [CrossRef]
8. 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; pp. 223–224.
9. Ezuma, M.; Ozdemir, O.; Anjinappa, C.K.; Gulzar, W.A.; Guvenc, I. Micro-UAV Detection with a Low-Grazing Angle Millimeter Wave Radar. In Proceedings of the 2019 IEEE Radio and Wireless Symposium (RWS), Orlando, FL, USA, 20–23 January 2019; pp. 1–4.
10. Kaplan, B.; Kahraman, I.; Ekti, A.R.; Yarkan, S.; Görçin, A.; Özdemir, M.K.; Çirpan, H.A. Detection, Identification, and Direction of Arrival Estimation of Drone FHSS Signals with Uniform Linear Antenna Array. *IEEE Access* **2021**, *9*, 152057–152069. [CrossRef]
11. Park, J.; Kim, D.H.; Shin, Y.S.; Lee, S. A comparison of convolutional object detectors for real-time drone tracking using a PTZ camera. In Proceedings of the 2017 17th International Conference on Control, Automation and Systems (ICCAS), Jeju, Republic of Korea, 18–21 October 2017; pp. 696–699.
12. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
13. Bharati, P.; Pramanik, A. Deep learning techniques—R-CNN to mask R-CNN: A survey. *Comput. Intell. Pattern Recognition. Adv. Intell. Syst. Comput.* **2020**, *999*, 657–668.
14. Zhang, C.; Xu, X.; Tu, D. Face Detection Using Improved Faster RCNN. *arXiv* **2018**, arXiv:1802.02142.
15. 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 (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
16. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
17. Aydin, B.; Singha, S. Drone Detection Using YOLOv5. *Eng* **2023**, *4*, 416–433. [CrossRef]
18. Kim, J.-H.; Kim, N.; Won, C.S. High-Speed Drone Detection Based On Yolo-V8. In Proceedings of the ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–2.
19. Zhai, X.; Huang, Z.; Li, T.; Liu, H.; Wang, S. YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection. *Electronics* **2023**, *12*, 3664. [CrossRef]
20. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
21. Chen, S.; Yang, H.; Zhang, A.; Chen, B.; Shu, P.; Xiang, J.; Lin, C. UAV Dynamic Tracking Algorithm Based on Deep Learning. In Proceedings of the 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China, 3–5 December 2021; pp. 482–485.
22. Sie, N.J.; Srigrarom, S.; Huang, S. Field Test Validations of Vision-Based Multi-Camera Multi-Drone Tracking and 3D Localizing with Concurrent Camera Pose Estimation. In Proceedings of the 2021 6th International Conference on Control and Robotics Engineering (ICCRE), Beijing, China, 16–18 April 2024; pp. 139–144.
23. Ezuma, M.; Erden, F.; Kumar Anjinappa, C.; Ozdemir, O.; Guvenc, I. Detection and Classification of UAVs Using RF Fingerprints in the Presence of Wi-Fi and Bluetooth Interference. *IEEE Open J. Commun. Soc.* **2020**, *1*, 60–76. [CrossRef]
24. Nemer, I.; Sheltami, T.; Ahmad, I.; Yasar, A.U.-H.; Abdeen, M.A.R. RF-Based UAV Detection and Identification Using Hierarchical Learning Approach. *Sensors* **2021**, *21*, 1947. [CrossRef] [PubMed]
25. AlKhonaini, A.; Sheltami, T.; Mahmoud, A.; Imam, M. UAV Detection Using Reinforcement Learning. *Sensors* **2024**, *24*, 1870. [CrossRef] [PubMed]
26. Xiang, B.; Meng, H.; Zhou, J.; Liu, Y.; He, L.; Luo, Z.; Zeng, C. UAV Detection Research Based on A Low-Cost SDR. In Proceedings of the 2020 International Conference on Microwave and Millimeter Wave Technology (ICMMT), Shanghai, China, 20–23 September 2020; pp. 1–3.
27. Flak, P. Drone Detection Sensor with Continuous 2.4 GHz ISM Band Coverage Based on Cost-Effective SDR Platform. *IEEE Access* **2021**, *9*, 114574–114586. [CrossRef]
28. Jiang, G.; Nan, P.; Zhang, J.; Li, Y.; Li, X. Robust Spatial-Temporal Autoencoder for Unsupervised Anomaly Detection of Unmanned Aerial Vehicle with Flight Data. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–14. [CrossRef]
29. Basak, S.; Rajendran, S.; Pollin, S.; Scheers, B. Autoencoder Based Framework for Drone RF Signal Classification and Novelty Detection. In Proceedings of the 2023 25th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Republic of Korea, 19–22 February 2023; pp. 218–225.

30. Walia, H. Birds vs. Drone Dataset. 2021. Available online: https://www.kaggle.com/datasets/harshwalia/birds-vs-drone-dataset (accessed on 9 September 2024).
31. Pawełczyk, M.; Wojtyra, M. Real World Object Detection Dataset for Quadcopter Unmanned Aerial Vehicle (UAV). *IEEE Access* **2020**, *8*, 174394–174409. [CrossRef]
32. Ali-bey, A.; Chaib-draa, B.; Giguère, P. GSV-Cities: Toward appropriate supervised visual place recognition. *Neurocomputing* **2022**, *513*, 194–203. [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.