engineering proceedings

MDPI

Proceeding Paper

Enhanced Drone Detection Model for Edge Devices Using Knowledge Distillation and Bayesian Optimization †

Maryam Lawan Salisu *, Farouk Lawan Gambo, Aminu Musa and Aminu Aliyu Abdullahi

Department of Computer Science, Federal University Dutse, Dutse 720101, Nigeria; farouk4142@gmail.com (F.L.G.); musa.aminu@fud.edu.ng (A.M.); aminuabdullahisumaila@fud.edu.ng (A.A.A.)

* Correspondence: marylsgumel@gmail.com

† Presented at the 5th International Electronic Conference on Applied Sciences, 4–6 December 2024; Available online: https://sciforum.net/event/ASEC2024.

Abstract: The emergence of Unmanned Aerial Vehicles (UAVs), commonly known as drones, has presented numerous transformative opportunities across sectors such as agriculture, commerce, and security surveillance systems. However, the proliferation of these technologies raises significant concerns regarding security and privacy, as they could potentially be exploited for unauthorized surveillance or even targeted attacks. Various research endeavors have proposed drone detection models for security purposes. Yet, deploying these models on edge devices proves challenging due to resource constraints, which limit the feasibility of complex deep learning models. The need for lightweight models capable of efficient deployment on edge devices becomes evident, particularly for the anonymous detection of drones in various disguises to prevent potential intrusions. This study introduces a lightweight deep learning-based drone detection model (LDDm-CNN) by fusing knowledge distillation with Bayesian optimization. Knowledge distillation (KD) is utilized to transfer knowledge from a complex model (teacher) to a simpler one (student), preserving performance while reducing computational complexity, thereby achieving a lightweight model. However, selecting optimal hyper-parameters for knowledge distillation is challenging due to a large number of search space and complexity requirements. Therefore, through the integration of Bayesian optimization with knowledge distillation, we present an enhanced CNN-KD model. This novel approach employs an optimization algorithm to determine the most suitable hyper-parameters, enhancing the efficiency and effectiveness of the drone detection model. Validation on a dedicated drone detection dataset illustrates the model's efficacy, achieving a remarkable accuracy of 96% while significantly reducing computational and memory requirements. With just 102,000 parameters, the proposed model is five times smaller than the teacher model, underscoring its potential for practical deployment in real-world scenarios.

Keywords: UAVs; lightweight model; Object detection; CNN; Knowledge distillation; Bayesian optimization; Edge devices

check for updates

Academic Editor: Eugenio Vocaturo

Published: 4 June 2025

Citation: Salisu, M.L.; Gambo, F.L.; Musa, A.; Abdullahi, A.A. Enhanced Drone Detection Model for Edge Devices Using Knowledge Distillation and Bayesian Optimization. *Eng. Proc.* **2025**, *87*, 71. https://doi.org/10.3390/engproc2025087071

Copyright: © 2025 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

The rise of drones, or unmanned aerial vehicles (UAVs), has transformed various industries by offering advanced technology and convenience. Equipped with sensors, GPS, cameras, and communication systems, drones perform tasks in sectors like surveillance, agriculture, search and rescue, filmmaking, and delivery [1], making them essential for accessing remote areas and capturing high-resolution images [2].

However, the increasing use of drones has also raised serious concerns regarding privacy, security, and public safety [3,4]. Drones can be exploited for malicious activities,

including trespassing, espionage, and even acts of terrorism [4]. To mitigate these risks and to ensure the safe integration of drones into our airspace, it is crucial to develop reliable drone detection and countermeasure systems. This task is particularly challenging in resource-constrained environments, where deploying advanced technology is difficult.

Resource-constrained environments are scenarios where computational power, memory, and energy are limited, making it impractical to deploy conventional, complex UAV detection models [5]. In these situations, developing lightweight and efficient detection systems is essential to enable real-time responses, safeguard critical infrastructure, and ensure public safety.

Machine learning techniques, particularly Convolutional Neural Networks (CNNs), have shown promise in addressing the security challenges posed by drones [6]. These models, which excel at learning features from images, have significantly advanced computer vision, leading to the development of highly effective drone detection systems. However, the computational complexity of deep learning models, such as CNNs and recurrent neural networks (RNNs), often makes real-time detection on resource-constrained devices infeasible [6].

To overcome these challenges, knowledge distillation offers a powerful solution. This machine learning technique involves transferring knowledge from a larger, more complex model (the teacher) to a smaller, more efficient model (the student) [7]. By inheriting the teacher's knowledge, the student model achieves similar performance levels but with reduced computational and memory requirements. Knowledge distillation is particularly effective in compressing neural networks while preserving their predictive power, making it ideal for UAV detection in constrained environments.

However, selecting the optimal hyperparameters for knowledge distillation can be challenging due to the vast search space and complexity involved. Bayesian optimization provides a systematic approach to hyperparameter tuning [8], allowing for the automatic selection of optimal configurations to maximize model performance. By integrating Bayesian optimization with knowledge distillation, this research aims to develop an enhanced CNN-based drone detection model.

The goal is to explore and harness knowledge distillation's potential to improve UAV detection in resource-constrained environments. By applying these techniques, the research creates a lightweight UAV detection model (LDDm-CNN) that balances efficiency, resource optimization, and detection accuracy. The significance of this work extends to various applications, including border security, wildlife conservation, critical infrastructure protection, and emergency response, where effective UAV detection is crucial.

2. Literature Review

2.1. Related Works

Traditional UAV detection methods like radar, photoelectric, radio frequency, and acoustic detection face challenges such as interference, blind spots, short range, and high costs. While photoelectric detection offers potential for real-time UAV feature extraction, it is hindered by weak image semantics and environmental interference [9].

Research in RF-based detection focuses on analyzing radio frequency signals to identify drones. This approach leverages distinct RF signatures, such as micro-Doppler effects and unique transmission patterns, which require specialized deep learning architectures to extract meaningful features from the signal data. Notable studies by Al-Sa'd et al. [3] present a study that systematically collects, analyzes, and records raw RF signals of different drones under various flight modes, such as off, on and connected, hovering, flying, and video recording. Meanwhile, with 98.9% accuracy using wavelet scattering transforms on steady-state RF signals, the results of RF-based drone detection using machine learning

and SqueezeNet CNNs demonstrate the potential of using both transient and steady-state RF signals for UAV classification, stimulating further research into resilient detection systems and data mining techniques [10]. These studies highlight the potential of applying machine learning to RF data for drone detection. However, they typically lack systematic hyperparameter optimization and do not detail the application of knowledge distillation within these techniques.

Visual-based drone detection remains a dominant approach due to the availability of high-resolution cameras and the rich information content in visual data. These methods typically employ convolutional neural networks (CNNs) designed to capture intricate spatial features present in image data. For example, Balachandran and S Sarath [11] address the issues faced by small, low-flying UAVs and provide a Generative Adversarial Network (GAN)-based technique for UAV recognition that uses the Pix2Pix framework. By generating a paired dataset of synthetic and actual photos from the drone vs. bird dataset, the proposed method outperforms existing UAV identification techniques, even in unfavorable weather situations, achieving a recall of 94%, precision of 93%, and F1-Score of 93%. Unlu [12] presents an autonomous drone detection and tracking system utilizing a static wide-angle camera and a lower-angle camera on a rotating turret. However, Schuman [13] proposes a UAV detection framework that begins with an initial detection phase using background subtraction techniques to identify potential targets, followed by a classification phase in which a dedicated CNN-based classifier categorizes the detected objects. Mahdiavi and Rajabi [14] address drone detection using a fisheye camera. It compares CNN, SVM, and nearest neighbor methods, showing CNN's superior accuracy (95%) in contrast to SVM (88%) and nearest neighbor (80%).

Acoustic detection methods analyze the sound patterns generated by drones, such as engine noise and rotor sounds. This modality requires unique preprocessing techniques and network architectures, such as recurrent neural networks (RNNs) or temporal convolutional networks, to effectively handle time-series audio data. Several works for acoustic modality were proposed in the past, such as an acoustic surveillance system for drone detection and localization with continuous availability, featuring novel feature extraction methods and TDOA estimation algorithms for enhanced accuracy [15]. A novel acoustic flight trajectory estimation method that integrates Doppler-based estimation of flight parameters and Direction of Arrival (DOA) estimation using a single array of acoustic sensors was seen in the literature [16]. Nevertheless, an advanced acoustic camera made of a 120-element microphone array and a video camera can detect amateur drones, with detection ranges spanning from 150 to 290 m and covering areas from 7 to 26 hectares, depending on the drone type. The system's beam-forming capabilities outperform commercial shotgun microphones, providing improved directivity, which is advantageous for real-time decision-making by operators and integration with other detection systems [17]. Prior work in this domain has shown promise but similarly falls short in addressing systematic parameter tuning and integrating knowledge transfer strategies.

Radar-based methods detect drones by analyzing reflected radio frequency signals and extracting motion-based features, such as micro-Doppler signatures. The radar successfully spotted the nano-drone and extracted its micro-Doppler signature using an STFT, demonstrating its effectiveness [18]. However, other research focuses on IAA as a promising technique with the potential to significantly improve drone detection using ground-based surveillance radars [19]. The advent of deep learning has revolutionized computer vision and offers potential solutions to UAV detection challenges, but radar-based studies often lack systematic hyperparameter optimization and rarely incorporate knowledge distillation to improve the efficiency and scalability of their models.

Deep learning methods autonomously learn UAV features, enabling effective feature extraction and improving detection accuracy. A novel classification approach utilizing the skinny pattern and iterative neighborhood component analysis (INCA) feature selector achieved impressive accuracy rates of up to 99.72% with the Fine kNN classifier [20], while a state-of-the-art convolutional neural network (CNN)-based surveillance system, CNN-SSDI, for UAV detection and type identification achieved remarkable accuracy rates of 99.8% for UAV detection and 94.5% for type identification, outperforming existing methods [21]. The difficulty of recognizing, in surveillance videos, small drones, which are comparable to birds and have complicated backgrounds, was resolved exploring many deep learning-based object detection algorithms, including ResNet-101 and Inception with Faster-RCNN, and it was found that Faster-RCNN based on ResNet-101 produces the best results [22].

Research on drone detection emphasizes using deep learning, especially CNNs, for accurate, real-time results. Models like YOLOv2 and YOLOv4 have proven effective, showcasing CNNs' potential in drone detection. For instance, YOLOv2 was modified and trained on various datasets, achieving real-time accuracy [23,24]. Similarly, studies using YOLOv4 and YOLOv5 modes achieved high accuracy and efficiency in drone detection tasks [25,26].

Despite these advancements, deploying CNN-based models on resource-constrained platforms is challenging due to high computational demands. To overcome this, lightweight models like Fast-YOLOv4 and techniques such as knowledge distillation have been introduced, enabling faster and more accurate detection on edge devices [27]. Knowledge distillation enables transferring knowledge from complex to simpler models, making it ideal for real-time drone detection in constrained environments. Bayesian optimization further enhances detection accuracy by optimizing CNN hyperparameters to find the best model configuration [28,29].

Knowledge distillation has been effectively employed to create lightweight models suitable for edge deployment. For instance, the KeepEdge framework utilizes knowledge distillation to develop a compact model for visual-assisted positioning in UAV delivery services, demonstrating the practical benefits of this approach in resource-constrained environments [30].

Recent studies have introduced innovative models tailored for drone detection. The TGC-YOLOv5 model integrates Transformer encoders and attention mechanisms to improve the detection of small drones in complex environments, showcasing significant performance enhancements over traditional YOLOv5 models [31]. Similarly, in a study presented by Zhao [32], the TPH-YOLOv5++ model incorporates a cross-layer asymmetric transformer to address challenges such as scale variation and high-density scenes in drone-captured images, achieving state-of-the-art results on datasets like VisDrone2021.

Efficient deployment of detection models on edge devices is critical for real-time applications. The MobileSAM-Track framework used by [33] in their work presents a lightweight solution for one-shot tracking and segmentation of small objects on edge devices, emphasizing the importance of model efficiency in resource-limited settings. Additionally, UAV-YOLOv5 proposes a lightweight object detection algorithm optimized for drone-captured scenarios, balancing precision and computational efficiency [34].

Most existing research on lightweight CNNs and knowledge distillation for drone detection does not systematically optimize hyperparameters, which results in incomplete comparative analyses and leaves a gap in effective parameter tuning. For instance, ref. [35] provides a comprehensive overview of machine learning-based drone detection techniques yet does not address the challenges of hyperparameter tuning in knowledge distillation. In contrast, ref. [28] demonstrates the use of Bayesian optimization for auto-tuning student

models via knowledge distillation, highlighting its potential to enhance model performance. Moreover, ref. [4] discusses the effectiveness of Bayesian optimization in deep learning contexts, reinforcing the need for a more rigorous optimization strategy. Advanced models such as TGC-YOLOv5 [11] and UAV-YOLOv5 [36] focus on balancing precision and computational efficiency, yet they do not incorporate systematic hyperparameter optimization within the knowledge distillation framework. The present work addresses this gap by integrating Bayesian optimization with knowledge distillation to develop LDDm-CNN, a lightweight, CNN-based drone detection model optimized for deployment on edge devices.

2.2. Research Gap

Existing research on drone detection often lacks a systematic approach to hyperparameter optimization, relying on manual tuning, which is both time-consuming and suboptimal. This limitation is addressed by integrating Bayesian optimization and knowledge distillation to develop LDDm-CNN, a lightweight yet highly effective drone detection model. This integration enables efficient learning while ensuring that the model remains computationally feasible for real-time deployment.

Unlike prior works that rely on trial-and-error methods for selecting hyperparameters, our model employs Bayesian optimization to systematically fine-tune critical hyperparameters, such as the distillation temperature (T) and balancing factor (α). By leveraging Bayesian optimization, the proposed approach eliminates the inefficiencies of manual tuning and ensures optimal model performance while minimizing the computational cost associated with hyperparameter selection.

Furthermore, knowledge distillation is utilized to transfer knowledge from a larger, high-performing teacher model to a compact student model. Unlike conventional distillation techniques that solely prioritize accuracy preservation, this approach enhances the distillation process by optimizing hyperparameter selection through Bayesian optimization. This ensures that the student model achieves competitive accuracy while significantly reducing computational complexity, making it more practical for edge deployment.

Many existing drone detection models are computationally intensive, limiting their suitability for edge devices such as surveillance cameras, drones, and IoT systems. By designing a lightweight model that preserves high detection accuracy while ensuring computational efficiency, this issue is directly addressed. LDDm-CNN achieves a balance between detection accuracy and computational efficiency, making it highly suitable for real-time drone detection in resource-constrained environments.

By systematically optimizing hyperparameters and enhancing knowledge transfer, the proposed approach provides a novel and practical solution for drone detection in edge environments.

3. Materials and Methods

This section introduces the methodology employed in our research, detailing the proposed model and its individual components. Additionally, we explain the concept of knowledge distillation, including the distinctions between soft and hard classes, and outline the techniques used to build lightweight models using this approach.

3.1. Proposed LDDm-CNN Model

Figure 1 illustrates the overall architecture of the proposed LDDm-CNN model by clearly outlining the interactions between its key components. In the figure, the teacher network first processes the input data to produce logits, which are then converted into a softened probability distribution using the softmax (or scaled softmax) function; these outputs serve as the soft labels, capturing detailed information about the input. These

soft labels are transferred to the student network during Phase 1 of training, where the student learns to mimic the teacher's behavior through a distillation loss computed against these soft predictions. Concurrently, in Phase 2, the student network undergoes standard supervised training using ground truth labels, with a conventional cross-entropy loss reinforcing correct classifications. The overall loss is a weighted sum of these two components, ensuring balanced learning. Additionally, Bayesian optimization is integrated to automatically tune the hyperparameters. Specifically, the temperature (T) controls the softness of the probability distribution produced by the softmax function and the weight alpha (α) balances the distillation loss and the cross-entropy loss.

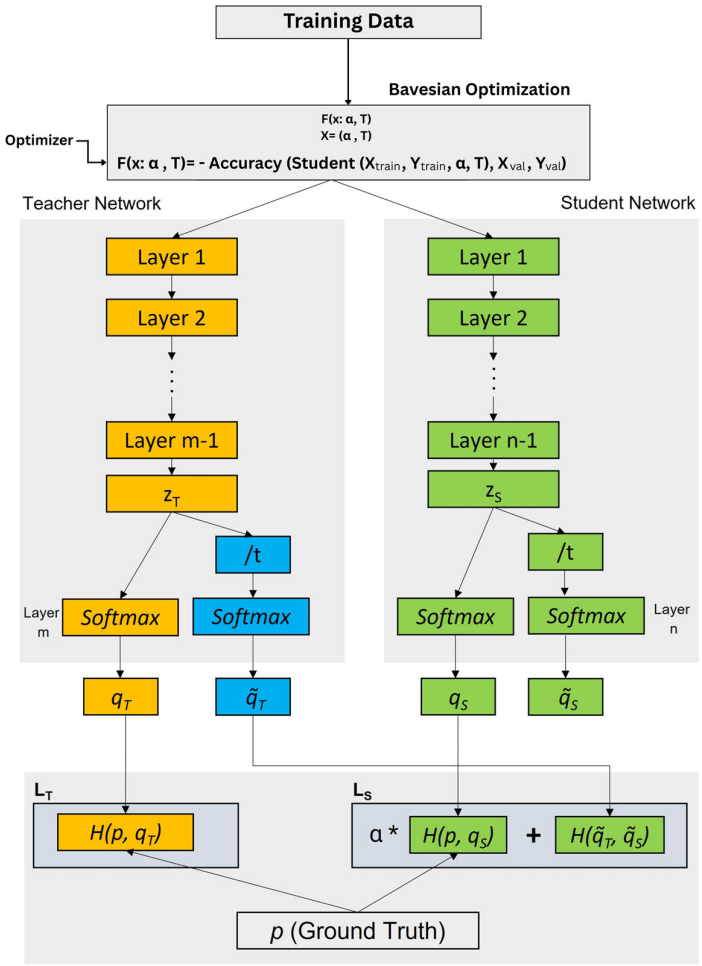


Figure 1. Proposed LDDm-CNN model architecture.

The optimization follows a probabilistic model-based approach, where a surrogate function, typically a Gaussian process, is used to model the objective function. An acquisition function then guides the search by balancing exploration and exploitation. The search process iteratively selects promising hyperparameter values, evaluates the model performance, and updates the surrogate function until convergence is achieved. To further illustrate this process, we have included a flowchart outlining the key steps involved in Bayesian optimization for hyperparameter tuning.

As illustrated in Figure 2, the training pipeline for the proposed LDDm-CNN model starts with Bayesian optimization, which identifies the optimal hyperparameters α and T, followed by teacher model's training to produce soft labels, which serve the student model as distilled knowledge. Concurrently, the student model also learns from the hard labels

via cross-entropy loss. By combining both losses and iteratively minimizing the general loss, the final LDDm-CNN model is obtained.

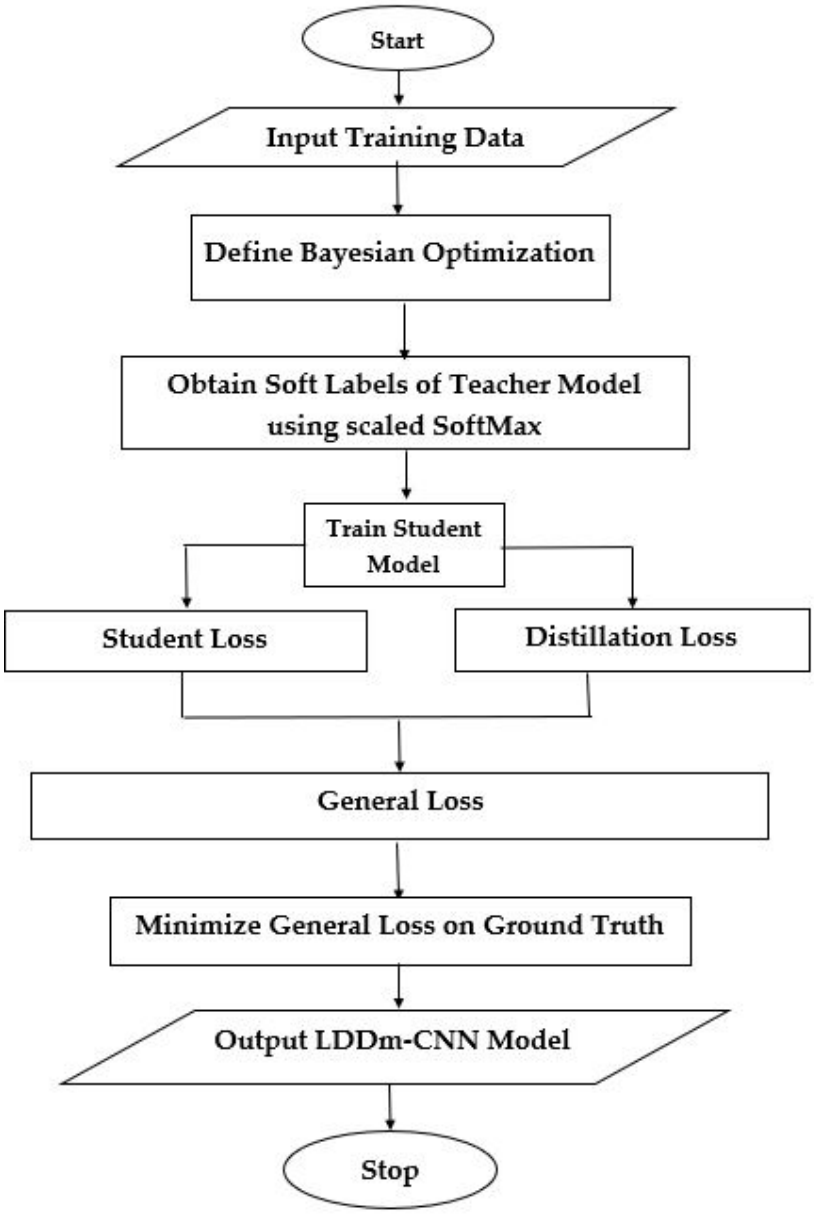


Figure 2. Proposed LDDm-CNN model flowchart.

3.2. Dataset Collection

The dataset utilized in this study is an open-source dataset named "Drone vs. Birds" obtained from Kaggle, a popular platform for data science and machine learning enthusiasts. The dataset is designed purposely for image classification to identify whether it is a bird or drone image in the sky. Originally, the dataset contained two directories with a total of 828 images of birds and drones combined. The dataset has 400 images of birds and 428 drones, respectively [33]. Additional images of 194 airplanes were collected through a digital camera and mobile phone imaging in various locations across the sky. This was done using various preprocessing techniques, such as data augmentation, flipping, and randomly sourcing over the internet, and added to the dataset to create a third airplane class in the dataset. This is necessary to ensure class balance among the classes of objects under investigation.

Figure 3 displays sample images from the dataset used in this study. These images illustrate the variety of conditions and perspectives captured during data collection, demonstrating the diverse visual characteristics that the proposed model must handle for effective drone detection.

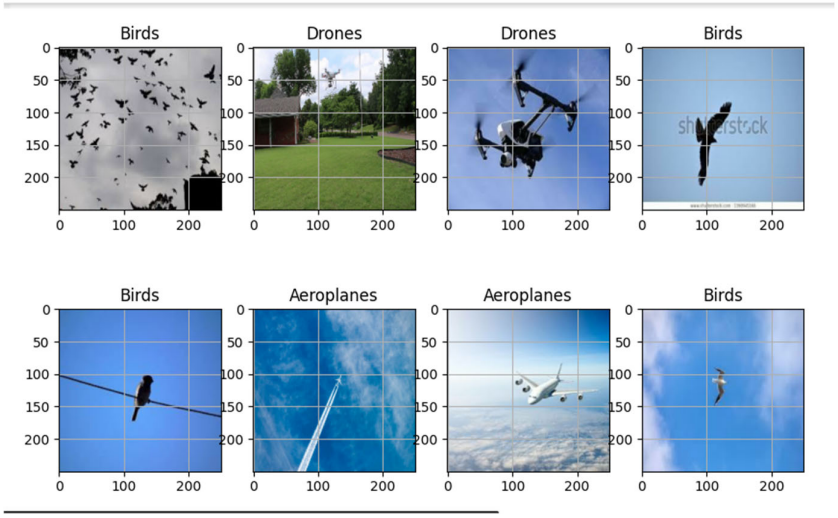


Figure 3. Sample images from the dataset.

3.3. Model Formulation

Equation (1) defines the process of obtaining soft labels, which serve as distilled knowledge for the student model; the logits produced by the teacher model are passed through the softmax function, transforming them into a probability distribution. These soft labels provide additional information beyond hard class labels, facilitating effective knowledge transfer during the distillation process.

$$\text{Softmax}\ (\mathrm{Z_i}) = \frac{e^{Z_i}}{\sum_j e^{Z_j}}, \tag{1}$$

where:

$\mathrm{Z_i}$ represents the logit for the ith class from the teacher model.

e^{z_i}: This term applies the exponential function to the logit Z_i.

$\sum_{je} z_j$: The sum in the denominator is the normalization factor, ensuring that all output values sum to one.

However, the traditional softmax function produces hard labels that are difficult for the student model to mimic; therefore, knowledge distillation introduces the concept of scaled softmax. Equation (2) presents the details of the scaled softmax function which is used to soften the probabilities using a hyper-parameter T.

$$\text{Softmax}\ (\mathrm{Z_i}) = \frac{e^{Z_i/T}}{\sum_j e^{Z_i/T}} \tag{2}$$

where:

T is the temperature hyper-parameter that controls the softness of the output probability distribution.

Z_i is the logit (unscaled prediction) for class i produced by the teacher model.

$e^{Z_i/T}$ scales the logit $\mathrm{Z_i}$ by the temperature before exponentiation.

$\sum_j e^{Z_i/T}$ sums the exponentials over all classes to normalize the probabilities so that they add up to one.

Scaled softmax is applied to the teacher model's logits, yielding distilled knowledge that is transferred to the student model during Phase 1, where the distillation loss defined in Equation (3) is computed by comparing the student's soft predictions to the distilled knowledge; in Phase 2, the student model is further refined by calculating the cross-entropy loss between its predictions and the ground truth labels. For a clearer understanding of these sequential operations, a signal flow chart has been added (see Figure 2), which visually outlines the process from knowledge distillation to loss computation.

$$\text{General loss} = \alpha\ (\text{student loss}) + (1 - \alpha)\ \text{distillation loss} \tag{3}$$

where:

α stands for Alpha, which is a knowledge distillation hyper-parameter.

The objective of knowledge distillation is to minimize the overall loss function, thereby achieving an accuracy level comparable to that of the teacher model. This requires the careful selection of knowledge distillation hyperparameters, α and T. However, due to the vast search space of these hyperparameters, identifying the optimal values for a specific task is highly complex. There is no straightforward method for selecting the best hyperparameter set, necessitating extensive experimentation with various parameter combinations. Even with numerous trials, certain potential configurations may remain unexplored.

Therefore, a novel approach to fused Bayesian optimization with knowledge distillation is proposed. Bayesian optimization is a powerful technique for optimizing hyperparameters of machine learning models [29]. Bayesian optimization is used to optimize a knowledge distillation hyperparameter search to find the optimal value of α and T based on the performance of the student model on the validation set. In particular, the hyperparameters are represented as x = (α,T) in Equation (5), and the optimization objective is defined as the negative accuracy of the student model presented in Equation (6); Equation (4) introduces this objective function. This integrated technique efficiently explores the hyperparameter space, ultimately identifying the values of α and T that yield the best performance in the knowledge distillation task.

The objective function for the optimization is given as:

$$\text{F(x: } \alpha, \text{ T)} \tag{4}$$

$$\text{x} = (\alpha, \text{ T}) \tag{5}$$

where:

α, T are the hyper-parameters to be optimized.

x = (α, T): hyper-parameters to optimize.

The optimization problem can be defined as:

$$\text{F(x: } \alpha, \text{ T)} = -\text{Accuracy (Student } (X_{\text{train}}, Y_{\text{train}}, \alpha, \text{ T}), X_{\text{val}}, Y_{\text{val}}) \tag{6}$$

Given:

X_{train}, X_{val}: Input features of training and validation datasets.

Y_{train}, Y_{val}: True labels of training and validation datasets.

The optimizer will therefore search the entire dense hyper-parameter search space until stopping conditions are met, in this case until the best set of results are found.

The optimizer returns this set of values for the teacher model to use and produce soft labels, which can then be used by the student model. The proposed model is given in equations below.

$$\text{Softmax}_{\text{scaled}}(Z_i) = \frac{e^{Z_i / Optimized_T}}{\sum_j e^{Z_i / Optimized_T}} \tag{7}$$

where:

Z_i is the logit (unscaled prediction) for class i produced by the teacher model.

$Optimized_T$ is the optimal value of T, which is determined through Bayesian optimization, ensuring that the student model achieves the best balance between learning from soft labels and generalizing well on real data.

We used Equation (7) to obtain distilled knowledge, which the student model will be used to train on. Subsequently, the student model training is given as follows:

$$\text{Loss} = \text{optimized_}\alpha \cdot (T(x)||S(x)) + (1-\alpha) \cdot CE(S(x), y) \quad (8)$$

where:

optimized_α represents the best weighting factor that balances the contribution of distillation loss and cross-entropy loss in the training process of the student model; it is determined using Bayesian optimization.

$T(x)$ represents the logits (pre-softmax outputs) of the teacher model for input x. It contains raw scores before applying the softmax function, which are later converted into probability distributions.

$S(x)$ represents the logits of the student model for the same input x. Like T(x), these are the un-normalized predictions made by the student model before applying softmax.

y represents the ground truth labels (i.e., the actual class labels of the input data).

CE is the cross-entropy loss.

The computed loss in Equation (8) is the loss function for enhanced knowledge distillation with parameter optimization.

$$\text{Loss} = \alpha \cdot SoftmaxLoss(Z_{student}, soft_labels) + (1-\alpha) . \cdot CE(Z_{student}, hard_labels) \quad (9)$$

where:

$SoftmaxLoss$ is the distillation loss, which measures the difference between the student model's predicted probabilities and the softened output of the teacher model.

$Z_{student}$ is the logits (raw output values) produced by the student model before applying the softmax function.

$soft_labels$ is the probability distributions generated by the teacher model after applying the softmax function with a temperature scaling factor, providing richer information than hard labels.

$hard_labels$ is the true class labels used in traditional supervised learning.

CE: is the cross-entropy loss

The total loss function in the knowledge distillation process, as described in Equation (9), combines distillation loss and cross-entropy loss to optimize the student model's learning. The distillation loss allows the student model to learn from the softened predictions of the teacher model, enhancing knowledge transfer. Meanwhile, the cross-entropy loss ensures that the student model correctly classifies data based on the actual labels. The weighting factor determines the balance between these two losses, where a higher value prioritizes knowledge transfer from the teacher, and a lower value focuses more on supervised learning. This approach allows the student model to maintain high accuracy while remaining efficient for deployment on resource-constrained edge devices.

3.4. Experimental Setup

The experiments were implemented using the Python 3.6 programming language in conjunction with the TensorFlow 2.7 machine learning framework on Google Colaboratory (Colab), which provided GPU T4 support, 107 GB of hard disk space, and 12 GB of RAM.

3.5. Evaluation Metrics

3.5.1. Accuracy

Accuracy is utilized in this work, being one of the most popular evaluation metrics used for evaluating classifier performance of a balance class. It measures how well the model correctly distinguishes between drones, birds, and airplanes. It is calculated as the proportion of correctly identified drones, birds, and airplanes out of all the images analyzed. A high accuracy means the model effectively recognizes drones while avoiding misclassifications with birds and airplanes. However, as Equation (10) illustrates, accuracy alone may not always provide a comprehensive measure of performance, particularly when the occurrence of drones is significantly lower than that of other objects. In such cases, additional metrics are required to ensure a more balanced evaluation.

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{P} + \text{N}), \tag{10}$$

where:

P and N are total positive and negative classes.

Accuracy tells what proportion of negative and positive classes are correctly classified.

TP and TN represent True Positive and True Negative.

3.5.2. Mean Average Precision (mAP)

Mean Average Precision, as defined in Equation (11), is a crucial evaluation metric used to assess the effectiveness of the model in distinguishing drones from birds and airplanes. It calculates the mean of the average precision values across all object categories, ensuring a more detailed performance analysis beyond overall accuracy. By incorporating precision and recall across different detection thresholds, mAP provides a comprehensive measure of how well the model correctly identifies drones while minimizing false detections of birds and airplanes. A higher mAP score indicates that the model consistently detects drones with minimal confusion, making it particularly valuable for evaluating real-world drone detection scenarios.

$$\text{mAP} = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \tag{11}$$

where:

mAP stands for mean average precision.

n is the number of classes.

APk represents the Average Precision for the k-th class.

$\sum(k = 1\ to\ n)$ donates the summation from k = 1 to n.

3.5.3. F1-Score

In the context of drone detection, the F1 score as expressed mathematically in Equation (12) provides a balanced measure of a model's performance by considering both precision and recall. This metric is particularly valuable in our drone detection context, as it addresses potential class imbalances, ensuring a more reliable evaluation of the model's ability to accurately identify drones while minimizing false positives and negatives [34].

$$\text{F1} - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

where:

Precision is the ratio of true positive predictions to the total number of positive predictions made by the model (i.e., the sum of true positives and false positives).

Recall is the ratio of true positive predictions to the total number of actual positive instances (i.e., the sum of true positives and false negatives).

4. Results and Discussion

In this research, Bayesian optimization was employed to identify the optimal α and T values for a knowledge distillation process. ResNetV50, a pre-trained model widely recognized for classification tasks, served as the teacher model. The proposed model achieved promising performance compared to larger pre-trained models, significantly reducing complexity in terms of size, number of parameters, and training time, making it ideal for deployment on edge devices.

Three key experiments were conducted. The first involved constructing the Bayesian optimizer to optimize the hyperparameters α and T of the knowledge distillation technique. This optimization step was crucial for reducing the training time of the knowledge distillation model. Without optimal hyperparameters, the distillation process might not converge or deliver the best performance. The drone detection dataset was pre-processed by removing corrupted images, augmenting data to prevent overfitting, resizing images to a uniform size, and normalizing pixel values. The dataset was split into 70% for training and 30% for testing, yielding 714 training samples and 306 testing samples.

Using logistic regression as a mini-model, the search space for α (ranging from 0 to 1) and T (ranging from 1 to 10) was defined. The Bayesian optimization ran for 10 iterations, evaluating candidate values based on accuracy. The optimized process yielded the best α value at 0.183 and T value at 8; α is selected with three decimal places to capture its fine-tuning variations during optimization, while T is rounded to an integer because its impact on the softness of the probability distribution is best represented in discrete increments. The experimental results are illustrated in Figure 4.

```
Current minimum: -0.5400
Iteration No: 7 started. Evaluating function at random point.
Iteration No: 7 ended. Evaluation done at random point.
Time taken: 846.7951
Function value obtained: -0.5400
Current minimum: -0.5400
Iteration No: 8 started. Evaluating function at random point.
Iteration No: 8 ended. Evaluation done at random point.
Time taken: 848.3261
Function value obtained: -0.5400
Current minimum: -0.5400
Iteration No: 9 started. Evaluating function at random point.
Iteration No: 9 ended. Evaluation done at random point.
Time taken: 839.1597
Function value obtained: -0.5400
Current minimum: -0.5400
Iteration No: 10 started. Evaluating function at random point.
Iteration No: 10 ended. Evaluation done at random point.
Time taken: 839.9491
Function value obtained: -0.5400
Current minimum: -0.5400
Best T: 7.985775569916307
Best alpha: 0.18343478986616382
Best validation accuracy: 0.54
```

Figure 4. Bayesian optimization of best alpha and T.

Figure 4 illustrates the optimization process used to determine the optimal values of α and T. In this diagram, Bayesian optimization iteratively explores the hyperparameter space to identify the configuration that minimizes the overall loss. The resulting optimal values are then employed during model training to balance the distillation and cross-entropy loss components effectively.

In the second experiment, the teacher model was constructed using ResNet50v2, a pre-trained model with 73,755,403 parameters and a size of 281.35 MB. A dense layer was added as a prediction head on top of the base model, and the model was trained for 60 epochs with early stopping. The teacher model demonstrated strong performance on the drone detection dataset, as evidenced by the accuracy and validation curves in Figure 5 and the loss curve in Figure 6.

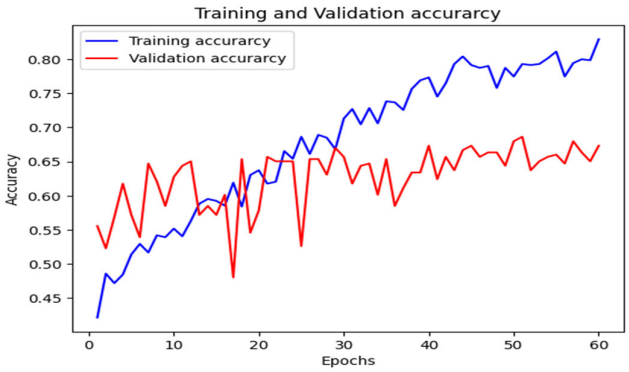


Figure 5. Teacher model training and validation accuracy.

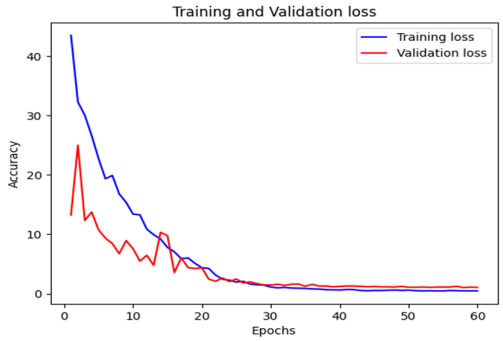


Figure 6. Teacher model training and validation loss.

The third experiment involved developing and training the proposed LDDm-CNN model, a shallow CNN with only one convolutional layer, one dense layer, a dropout layer, and a max-pooling layer, totaling eight layers with 1,477,123 parameters and a size of 5.63 MB, as illustrated in Table 1.

Table 1. Proposed model architecture.

Layer (Type)	Output Shape	Parameter
Conv2d (conv2D)	(None, 248, 248, 32)	896
activation_12 (Activation)	(None, 248, 248, 32)	0
max_poling2d_14 (Maxpooling2D)	(None, 124, 124, 32)	0
dropout_23 (Dropout)	(None, 124, 124, 32)	0
batch_normalization_10 (BatchNormalization)	(None, 124, 124, 32)	128
dropout_24 (Dropout)	(None, 124, 124, 32)	0
flatten_10 (Flatten)	(None, 492,032)	0
dense_24 (Dense)	(None, 3)	1,476,099

Total params: 1,477,123 (5.63 MB). Trainable params: 1,477,059 (5.3 MB).

The model was trained using knowledge distillation with optimal values of α and T obtained through Bayesian optimization, significantly improving performance and reducing training time. The accuracy and loss curves for the proposed model, shown in Figures 7 and 8, indicate that it achieved 95% accuracy with minimal loss, outperforming the teacher model.



Figure 7. Accuracy of the proposed LDDm-CNN model.

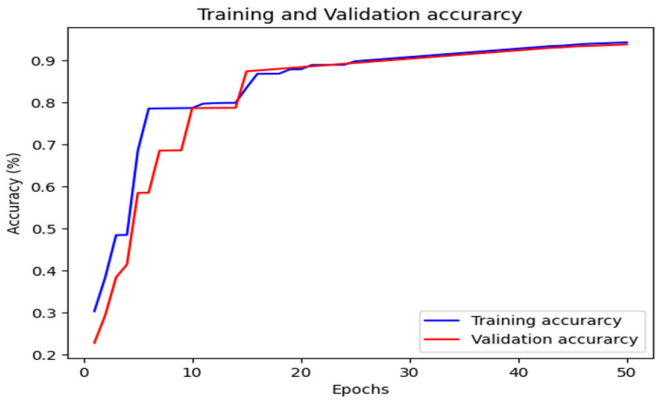


Figure 8. Proposed LDDm-CNN model loss.

Performance metrics such as precision, recall, and F1-score, presented in Table 2, further validated the model's superior performance compared to the teacher model. The proposed model reduced training time from 14 min to 10 min and required only 5 MB of disk space, making it highly suitable for deployment on edge devices. In contrast, the teacher model's 288 MB size limits its usability in resource-constrained environments.

Table 2. Performance of the proposed LDDm-CNN model.

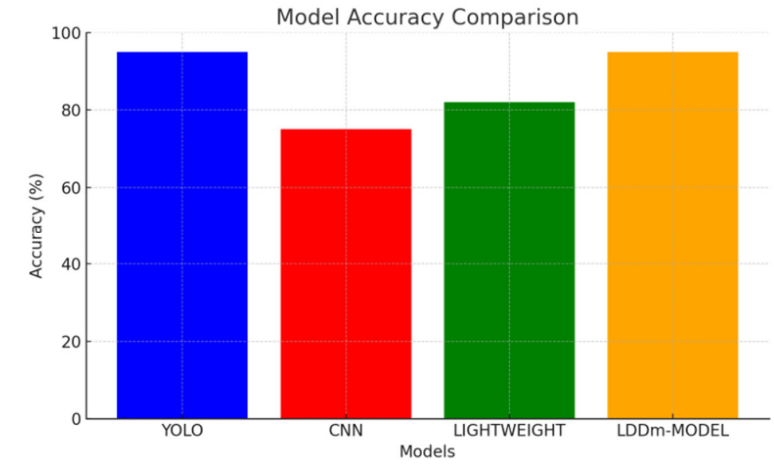
Models	Precision (%)	Recall (%)	F1 Score (%)	Accuracy	Model Size	Training Time	No. Params
Proposed model	0.89	0.90	0.89	0.95	5.63 MB	10 min	1,477,123
Teacher model	0.70	0.74	0.73	0.74	281.35 MB	14 min	73,755,403

When benchmarked against existing drone detection models, as detailed in Table 3, the proposed model excels in terms of size, training time, and real-time inference capabilities, despite being smaller and less complex. This demonstrates the effectiveness of knowledge distillation and hyperparameter optimization in building efficient, lightweight models. Therefore, the proposed LDDm-CNN model is a superior choice for edge device applications, offering versatility across various tasks beyond drone detection.

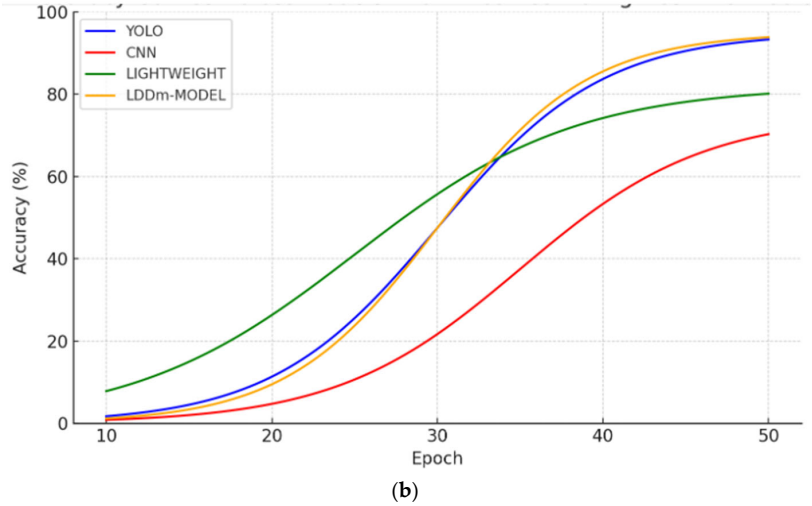
As depicted in Figure 9a, the proposed LDDm-MODEL attains an accuracy of 95%, matching that of the much larger YOLO architecture and significantly exceeding the standard CNN (75%) and the lightweight baseline (82%). This comparison confirms that LDDm-CNN successfully preserves high detection performance, on par with state-of-the-art models, while maintaining a compact size and reduced parameter count, thereby proving its suitability for resource-constrained edge deployment.

Table 3. Comparison of the proposed LDDm-CNN model with bigger models.

Models	Accuracy (%)	Recall (%)	Precision (%)	Model Size	Training Time	No. Params
[24]	-	0.68	0.95	244 MB	6 h	64 million
[35]	0.996	0.998	0.997	1.5 MB	68 min	300,000
[22].	0.975	0.980	0.980	12 MB	8.9 h	2,444,928
Proposed model	0.95	0.90	0.89	5.63 MB	10 min	1,477,123



(**a**)



(**b**)

Figure 9. (**a**): Accuracy comparison of drone detection models, (**b**): Accuracy curves comparison with existing models.

The accuracy curves for YOLO, CNN, LIGHTWEIGHT, and LDDm-MODEL, as seen in Figure 9b, show how each model's accuracy improves over training. By the 30th epoch, LDDm-MODEL reaches about 80% accuracy and climbs to 95% by the 50th epoch, matching YOLO and outperforming both the standard CNN and the lightweight model. This fast convergence and high final accuracy support our claim that the LDDm-CNN architecture delivers strong performance while remaining efficient enough for edge device deployment.

5. Conclusions

In conclusion, the study introduces the LDDm-CNN, a lightweight drone detection model specifically designed for resource-constrained environments. This model leverages a shallow Convolutional Neural Network (CNN) architecture, optimized for efficiency, making it suitable for real-time detection on edge devices with limited computational

resources. The experimental results demonstrate the effectiveness of the proposed model, achieving a remarkable accuracy of 95% with reduced training time and a model size of only 5 MB. The LDDm-CNN outperforms the teacher model in several metrics, including accuracy, precision, recall, and F1-score, making it a superior choice for real-time UAV detection on edge devices. Comparing the LDDm-CNN with existing models highlights the model's advantages in terms of size, speed, and efficiency, confirming that the research objectives have been successfully achieved.

Author Contributions: Conceptualization, M.L.S. and F.L.G.; methodology, M.L.S.; software, A.M.; validation, M.L.S., F.L.G. and A.A.A.; formal analysis, A.M.; investigation, M.L.S.; resources, F.L.G.; data curation, A.A.A.; writing—original draft preparation, M.L.S.; writing—review and editing, F.L.G. and A.M.; visualization, A.A.A.; supervision, F.L.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by TETFUND National Research Fund (TETF/DR&D-CE/NRF2021/SETI/ICT/00133)-2021 intervention.

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 privacy but can be provided for research purposes only.

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

References

1. Kardasz, P.; Doskocz, J. Drones and Possibilities of Their Using. *J. Civ. Environ. Eng.* **2016**, *6*, 233. [CrossRef]
2. Hossain, R. A Short Review of the Drone Technology. *Int. J. Mechatron. Manuf. Technol.* **2022**, *7*, 53–67.
3. Al-Sa'd, M.F.; Al-Ali, A.; Mohamed, A.; Khattab, T.; Erbad, A. RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database. *Future Gener. Comput. Syst.* **2019**, *100*, 86–97. [CrossRef]
4. Bera, B.; Das, A.K.; Sutrala, A.K. Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in Internet of Drones environment. *Comput. Commun.* **2021**, *166*, 91–109. [CrossRef]
5. Musa, A.; Hamada, M.; Hassan, M. A Theoretical Framework Towards Building a Lightweight Model for Pothole Detection using Knowledge Distillation Approach. *SHS Web Conf.* **2022**, *139*, 03002. [CrossRef]
6. Taha, B.; Member, S.; Shoufan, A. Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research. *IEEE Access* **2019**, *7*, 138669–138682. [CrossRef]
7. Musa, A.; Hassan, M.; Hamada, M.; Aliyu, F. Low-Power Deep Learning Model for Plant Disease Detection for Smart-Hydroponics Using Knowledge Distillation Techniques. *J. Low Power Electron. Appl.* **2022**, *12*, 24. [CrossRef]
8. Loka, N.R.B.S.; Couckuyt, I.; Garbuglia, F.; Spina, D.; Van Nieuwenhuyse, I.; Dhaene, T. Bi-objective Bayesian optimization of engineering problems with cheap and expensive cost functions. *Eng. Comput.* **2022**, *39*, 1923–1933. [CrossRef]
9. Dong, Y.; Ma, Y.; Li, Y.; Li, Z. High-precision real-time UAV target recognition based on improved YOLOv4. *Comput. Commun.* **2023**, *206*, 124–132. [CrossRef]
10. Medaiyese, O.O.; Ezuma, M.; Lauf, A.P.; Guvenc, I. Wavelet transform analytics for RF-based UAV detection and identification system using machine learning. *Pervasive Mob. Comput.* **2022**, *82*, 101569. [CrossRef]
11. Balachandran, V.; Sarath, S. A Novel Approach to Detect Unmanned Aerial Vehicle using Pix2Pix Generative Adversarial Network. In Proceedings of the 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 23–25 February 2022. [CrossRef]
12. Unlu, E.; Zenou, E.; Riviere, N.; Dupouy, P.-E. Deep learning-based strategies for the detection and tracking of drones using several cameras. *IPSJ Trans. Comput. Vis. Appl.* **2019**, *11*, 7. [CrossRef]
13. 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. Available online: https://ieeexplore.ieee.org/abstract/document/8078558 (accessed on 12 December 2024).
14. Mahdavi, F.; Rajabi, R. Drone Detection Using Convolutional Neural Networks. In Proceedings of the 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), Mashhad, Iran, 23–24 December 2020.

15. 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]
16. Tong, J.; Xie, W.; Hu, Y.-H.; Bao, M.; Li, X.; He, W. Estimation of low-altitude moving target trajectory using single acoustic array. *J. Acoust. Soc. Am.* **2016**, *139*, 1848–1858. [CrossRef] [PubMed]
17. Busset, J.; Perrodin, F.; Wellig, P.; Ott, B.; Heutschi, K.; Rühl, T.; Nussbaumer, T. Detection and tracking of drones using advanced acoustic cameras. In Proceedings of the Unmanned/Unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications, Toulouse, France, 13 October 2015; Volume 9647. [CrossRef]
18. 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. Available online: https://ieeexplore.ieee.org/abstract/document/9266538 (accessed on 14 November 2022).
19. 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]
20. Akbal, E.; Akbal, A.; Dogan, S.; Tuncer, T. An automated accurate sound-based amateur drone detection method based on skinny pattern. *Digit. Signal Process.* **2023**, *136*, 104012. [CrossRef]
21. Akter, R.; Doan, V.-S.; Lee, J.-M.; Kim, D.-S. CNN-SSDI: Convolution neural network inspired surveillance system for UAVs detection and identification. *Comput. Netw.* **2021**, *201*, 108519. [CrossRef]
22. 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. [CrossRef]
23. Wu, M.; Xie, W.; Shi, X.; Shao, P.; Shi, Z. Real-Time Drone Detection Using Deep Learning Approach. In *Machine Learning and Intelligent Communications*; Springer: Cham, Swizterland, 2018; pp. 22–32. [CrossRef]
24. Singha, S.; Aydin, B. Automated Drone Detection Using YOLOv4. *Drones* **2021**, *5*, 95. [CrossRef]
25. Cheng, Q.; Wang, H.; Zhu, B.; Shi, Y.; Xie, B. A Real-Time UAV Target Detection Algorithm Based on Edge Computing. *Drones* **2023**, *7*, 95. [CrossRef]
26. Kishore, J.; Mukherjee, S. Autotuning Student Models via Bayesian Optimization with Knowledge Distilled from Self-Supervised Teacher Models. January 2023. Available online: https://ssrn.com/abstract=4579155 (accessed on 12 December 2024).
27. Agnihotri, A.; Batra, N. Exploring Bayesian Optimization. *Distill* **2020**, *5*, e26. [CrossRef]
28. Luo, H.; Chen, T.; Li, X.; Li, S.; Zhang, C.; Zhao, G.; Liu, X. 2023. KeepEdge: A Knowledge Distillation Empowered Edge Intelligence Framework for Visual Assisted Positioning in UAV Delivery. *IEEE Trans. Mob. Comput.* **2023**, *22*, 4729–4741. [CrossRef]
29. Zhao, Y.; Ju, Z.; Sun, T.; Dong, F.; Li, J.; Yang, R.; Fu, Q.; Lian, C.; Shan, P. TGC-YOLOv5: An Enhanced YOLOv5 Drone Detection Model Based on Transformer, GAM & CA Attention Mechanism. *Drones* **2023**, *7*, 446. [CrossRef]
30. Musa, A.; Adam, F.M.; Ibrahim, U.; Zandam, A.Y. Learning from small datasets: An efficient deep learning model for COVID-19 detection from chest X-Ray using dataset distillation technique. In Proceedings of the 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), Lagos, Nigeria, 5–7 April 2022; pp. 1–6. [CrossRef]
31. Zhao, Q.; Liu, B.; Lyu, S.; Wang, C.; Zhang, H. TPH-YOLOv5++: Boosting Object Detection on Drone-Captured Scenarios with Cross-Layer Asymmetric Transformer. *Remote Sens.* **2023**, *15*, 1687. [CrossRef]
32. Liu, Y.; Zhao, Y.; Zhang, X.; Wang, X.; Lian, C.; Li, J.; Shan, P.; Fu, C.; Lyu, X.; Li, L.; et al. MobileSAM-Track: Lightweight One-Shot Tracking and Segmentation of Small Objects on Edge Devices. *Remote Sens.* **2023**, *15*, 5665. [CrossRef]
33. Li, G.; Liu, E.; Wang, Y.; Yang, Y.; Liu, H. UAV-YOLOv5: A Lightweight Object Detection Algorithm on Drone-Captured Scenarios. *Sci. J. Intell. Syst. Res.* **2024**, *6*, 24–33. [CrossRef]
34. Available online: https://www.kaggle.com/datasets/harshwalia/birds-vs-drone-dataset (accessed on 12 December 2024).
35. Sokolova, M.; Japkowicz, N.; Szpakowicz, S. Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In *AI 2006: Advances in Artificial Intelligence*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4304, pp. 1015–1021. [CrossRef]
36. Ajakwe, S.; Arkter, R.; Kim, D.; Kim, D.; Lee, J.M. Lightweight cnn model for detection of unauthorized uav in military reconnaissance operations. *Korea Inst. Commun. Sci.* **2021**, *11*, 113–115.

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.