

Article

A Lightweight Drone Detection Method Integrated into a Linear Attention Mechanism Based on Improved YOLOv11

Sicheng Zhou ¹, Lei Yang ¹, Huiting Liu ¹, Chongqing Zhou ¹, Jiacheng Liu ¹, Shuai Zhao ² and Keyi Wang ^{1,*}

¹ Department of Precision Machinery and Precision Instrument, University of Science and Technology of China, Hefei 230026, China; sichengz@mail.ustc.edu.cn (S.Z.); lyang01@mail.ustc.edu.cn (L.Y.); lht66@mail.ustc.edu.cn (H.L.); zchongqing2002@mail.ustc.edu.cn (C.Z.); lj20020220@mail.ustc.edu.cn (J.L.)

² National Synchrotron Radiation Laboratory, University of Science and Technology of China, Hefei 230029, China; shuai.zhao@ustc.edu.cn

* Correspondence: kywang@ustc.edu.cn; Tel.: +86-1303-3059-750

Abstract: The timely and accurate detection of unidentified drones is vital for public safety. However, the unique characteristics of drones in complex environments and the varied postures they may adopt during approach present significant challenges. Additionally, deep learning algorithms often require large models and substantial computational resources, limiting their use on low-capacity platforms. To address these challenges, we propose LAMS-YOLO, a lightweight drone detection method based on linear attention mechanisms and adaptive downsampling. The model's lightweight design, inspired by CPU optimization, reduces parameters using depthwise separable convolutions and efficient activation functions. A novel linear attention mechanism, incorporating an LSTM-like gating system, enhances semantic extraction efficiency, improving detection performance in complex scenarios. Building on insights from dynamic convolution and multi-scale fusion, a new adaptive downsampling module is developed. This module efficiently compresses features while retaining critical information. Additionally, an improved bounding box loss function is introduced to enhance localization accuracy. Experimental results demonstrate that LAMS-YOLO outperforms YOLOv11n, achieving a 3.89% increase in mAP and a 9.35% reduction in parameters. The model also exhibits strong cross-dataset generalization, striking a balance between accuracy and efficiency. These advancements provide robust technical support for real-time drone monitoring.



Academic Editor: Richard Gloaguen

Received: 14 January 2025

Revised: 15 February 2025

Accepted: 17 February 2025

Published: 19 February 2025

Citation: Zhou, S.; Yang, L.; Liu, H.; Zhou, C.; Liu, J.; Zhao, S.; Wang, K. A Lightweight Drone Detection Method Integrated into a Linear Attention Mechanism Based on Improved YOLOv11. *Remote Sens.* **2025**, *17*, 705. <https://doi.org/10.3390/rs17040705>

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/>).

Keywords: drone detection; lightweight; attention mechanism

1. Introduction

In recent years, with the development and widespread adoption of drone technology, drones have been extensively utilized in fields such as environmental monitoring, infrastructure inspection, and agriculture [1–7]. However, with the widespread application of drones, safety concerns have also increased. These incidents have resulted in injuries and property damage, drawing significant global attention. Nick Tepylo et al. [8] investigated public attitudes toward drones and found that most respondents were concerned about their safety risks. Consequently, developing a technology for the accurate and efficient detection of unidentified drones is both urgent and essential.

Drone detection methods can be broadly categorized into traditional techniques and computer vision-based approaches [9]. Traditional techniques include radar detection and other similar methods [10–16]. Wang et al. [14] proposed a drone detection method that integrates range-Doppler maps with satellite images. They also developed a labeling

technique for generating a frame-by-frame labeled echo dataset. Hu et al. [15] examined the effects of drone geometric acceleration and path loss on radar detection performance. However, the resolution of radar systems is inherently limited by factors such as frequency and wavelength. For small drones, this resolution is often inadequate to effectively identify and distinguish targets with low radar cross-sections. Furthermore, radar systems are susceptible to external influences, including terrain, weather conditions, and electromagnetic interference. In contrast, computer vision-based detection, with its capability to extract multi-level features and its robustness in complex environments, has emerged as a widely adopted approach for drone detection.

Computer vision research methods are generally divided into traditional machine learning and deep learning approaches. Traditional machine learning involves manually extracting features such as edges, textures, colors, and shapes from input images using predefined techniques. These features are then classified and identified using algorithms like support vector machine (SVM), decision trees, or random forests. For instance, Lee et al. [17] detected pipeline leaks using various machine learning models, achieving an accuracy of up to 99.79%. Similarly, Sayer et al. [18] applied machine learning methods based on mechanical control to classify four types of drones. By utilizing different algorithms on a drone control dataset, they achieved an accuracy exceeding 90%. Anwar et al. [19] developed a novel machine learning framework to identify amateur drone (ADr) sounds in noisy environments. This framework, utilizing an SVM cubic kernel algorithm, achieved an ADr detection accuracy of approximately 96.7%. Wei et al. [20] introduced a GPS spoofing detection method for drones, termed PERDET, which is based on perception data. Experimental data were gathered during actual flights, and various machine learning methods were applied to the dataset for performance evaluation and comparison. The findings showed that PERDET is highly effective, achieving a detection rate of 99.69%. However, traditional machine learning methods, relying on manually designed features, often fail to effectively process complex, high-dimensional data or detect small targets. This limitation can lead to reduced accuracy and increased false positive rates. In contrast, deep learning has gained widespread adoption in target detection due to its ability to automatically extract features from data and model complex patterns using multi-layer networks.

Deep learning methods are divided into two-stage and one-stage algorithms. Two-stage algorithms first employ a region proposal network (RPN) to generate candidate regions. These regions are then refined through classification and regression. Prominent examples of two-stage algorithms include Faster-RCNN, Mask-RCNN, and Cascade-RCNN. Feng et al. [21] introduced MAVFE for multi-scale voxel representation, MSRGF for RoI pooling, and CAM for incremental bicycle and pedestrian detection on the KITTI dataset. Li et al. [22] proposed a simple and effective two-stage fusion framework for traffic sign detection, achieving experimental results of 89.7% mAP and 65 FPS on the TT100K dataset. Li et al. [23] introduced an anchor-free quality-oriented proposal network (QOPN) leveraging dynamic label assignment and attention-based decomposition. They also developed a novel adaptive recognition loss (ARL), achieving state-of-the-art results on various FGOD datasets. However, the method requires two inference stages, resulting in high computational costs. Additionally, the generation of candidate regions introduces delays, limiting its suitability for real-time applications. In contrast, one-stage algorithms streamline the detection process into an end-to-end framework, delivering faster results with lower computational demands, making them better suited for time-sensitive tasks like drone detection.

One-stage deep learning methods include the YOLO series [24–29], the SSD series [30], and RetinaNet [31]. Peng et al. [32] improved YOLOv5 with CA and BiFPN, creating a lightweight model for remote sensing detection. Xue et al. [33] introduced EL-YOLO,

enhancing YOLOv5 with SCAFPN and CSL-MHSA, achieving 12.4% and 1.3% mAP50 improvements over the baseline. Wang et al. [34] improved YOLOv8n with CIAM and TAM, achieving 93.9% average precision and 95.7% amount precision for cow estrus detection. Huang et al. [35] introduced a lightweight, real-time, and accurate anti-drone detection model, EDGS-YOLOv8. On the DUT anti-UAV dataset, EDGS-YOLOv8 achieved an AP value of 0.971, surpassing the mAP of YOLOv8n by 3.1%, while maintaining a compact model size of only 4.23 MB. Wang et al. [36] developed a lightweight drone swarm detection method based on YOLOX. This approach utilizes depthwise separable convolutions to streamline and optimize the network, reducing the total number of parameters. Experimental results demonstrate that the proposed method attains an mAP of 82.32%, approximately 2% higher than the baseline model, with a model size of just 3.85 MB. Bo et al. [37] proposed the YOLOv7-GS model, which improves the detection of small drones in complex backgrounds. By adjusting the size of prior bounding boxes, incorporating the InceptionNeXt module at the neck section's end and integrating the SPPFCSPC-SR and Get-and-Send modules, the final model delivered excellent results on both the DUT anti-UAV and amateur unmanned air vehicle detection datasets.

However, YOLOX introduces an anchor-free object detection framework, which enhances adaptability to objects of various scales. However, its feature representation capability remains relatively weak when handling dense objects. YOLOv5 exhibits lower accuracy in small object detection and struggles with complex backgrounds, showing limited adaptability to extreme scenarios. YOLOv7 improves real-time performance through speed optimization, but its accuracy in detecting multi-scale objects in complex environments is still suboptimal. YOLOv8 incorporates multi-scale feature fusion and enhanced generalization, though it still faces challenges with small object detection and special scenarios. YOLOv10 further optimizes the network architecture, resulting in improved detection accuracy, but issues with stability and speed in high-density objects and prolonged operation persist. In contrast, YOLOv11 significantly enhances detection accuracy by optimizing both the feature extraction module and training strategies. It efficiently processes complex scenarios while maintaining real-time performance, offering improved stability and performance compared to its predecessors.

In conclusion, although current object detection algorithms have achieved significant advancements across various applications, challenges remain in handling high-resolution tasks and complex backgrounds. The high computational complexity of these algorithms limits their ability to capture fine details in high-resolution tasks. Additionally, the considerable computational cost of many methods restricts their applicability in real-time detection tasks. While some faster inference algorithms yield good results, their robustness in complex scenarios requires further enhancement. To address these challenges, this paper presents LAMS-YOLO, a lightweight object recognition model based on the YOLOv11 architecture. Compared to previous YOLO versions, LAMS-YOLO significantly reduces model parameters, achieving lightweight optimization. Moreover, the linear attention mechanism and adaptive downsampling module are incorporated into the neck layer, improving the model's detection capabilities in complex backgrounds. Finally, an enhanced loss function boosts the model's shape-matching ability for complex drones. The main contributions of this paper are as follows:

1. A lightweight feature extraction backbone network is introduced, significantly reducing model size and parameter count through depthwise separable convolutions and efficient activation functions. In resource-constrained environments, residual blocks are combined with depthwise separable convolutions to reduce computational costs, thereby enhancing real-time performance for drone detection applications.

2. To mitigate detail loss during feature extraction, an adaptive downsampling module is incorporated into the neck layer. This module dynamically adjusts the feature extraction process using dynamic convolutions and multi-scale fusion, improving the model's adaptability to diverse target regions.

3. To address feature processing challenges in complex backgrounds, a linear attention mechanism is introduced in the neck layer. This mechanism decomposes global dependencies into local operations via linear combinations based on feature partitions, simplifying the complexity of feature interactions. Additionally, rotational position encoding is employed in place of traditional absolute position encoding, boosting the model's ability to capture spatial position information.

4. To enhance the detection of complex-shaped objects, an improved bounding box regression loss function is introduced. This function incorporates a shape similarity metric for target boxes, considering aspect ratio and orientation. As a result, the predicted boxes not only exhibit a high overlap with the ground truth but also maintain shape consistency as much as possible.

Section 1 reviews the current status of traditional handcrafted feature extraction algorithms and deep learning detection methods, followed by an introduction to the proposed algorithm. Section 2 provides a detailed description of the LAMS-YOLO network. Section 3 analyzes the experimental setup and results, comparing them with existing literature. Finally, Section 4 discusses the experimental findings and draws conclusions.

2. Proposed Methods

YOLOv11 is a highly efficient real-time object detection algorithm. It uses CSPDarknet as its backbone network, offering greater efficiency and richer feature extraction compared to Darknet53 in YOLOv7 and YOLOv8. PANet is introduced for feature fusion, which outperforms the FPN used in YOLOv7 and YOLOv8 in terms of efficiency. Additionally, the non-maximum suppression (NMS) process is optimized to reduce redundant detection boxes, thereby improving detection accuracy. Given the real-time requirements of drone detection, this study adopts and further optimizes the lightweight YOLOv11n model as the baseline. The YOLOv11n architecture comprises four key components: the input layer, backbone layer, neck layer, and output layer. The structural details are shown in the Figure 1 below.

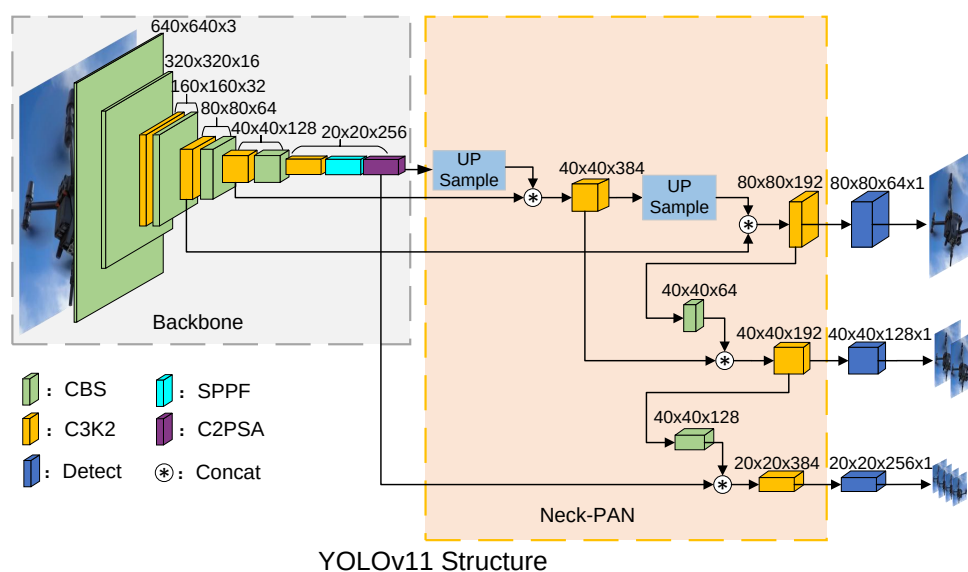


Figure 1. The structure of YOLOv11.

The input layer receives raw image data and preprocesses it. Its primary role is to convert the data into a format compatible with the network, enabling smooth progression through subsequent layers for feature extraction and object detection.

The backbone layer extracts image features by capturing deep semantic information, ranging from low-level features like edges and textures to high-level features like shapes. It comprises multiple convolutional layers, pooling layers, and activation functions.

The neck layer, situated between the backbone and output layers, facilitates feature fusion and enhancement. It enables accurate predictions of object locations and categories across different scales. PANet, incorporated in the neck layer, improves feature transmission and multi-scale learning by strengthening connections between features at various levels, enhancing detection performance across different scales.

The output layer translates the processed feature information from previous layers into final detection results. These results include object bounding box locations, class labels, and confidence scores.

2.1. Lightweight Feature Extraction Network Module

In drone detection, high latency can prevent the system from promptly identifying unidentified drones. Therefore, achieving high real-time detection for drones is essential. The backbone layer of YOLOv11 employs conventional convolution modules and the C3K2 module to achieve high-quality feature extraction and downsampling. The C3K2 module primarily facilitates feature fusion and cross-layer information transmission. However, conventional convolution operations are computationally intensive, particularly for high-resolution images, leading to slower inference speeds. This limitation makes them less suitable for real-time drone detection tasks. To address this, a new LCbackbone [38] has been proposed and designed in this study. The core concept of this module involves enhancing the activation function in BaseNet, incorporating an SE module at the end of the depthwise separable convolution, and increasing the convolution kernel size. Furthermore, a 1×1 convolution is added after the global average pooling layer. This design creates a lightweight backbone model while improving the network's fitting capability, achieving a better balance between speed and accuracy.

As shown in Figure 2, the LCBackbone module utilizes depthwise separable convolutions as its core building blocks, eliminating additional operations such as concat or elementwise-add to preserve inference speed. As shown in Equation (2), the H-Swish activation function in BaseNet is optimized by applying a linear transformation to the input and hardening the sigmoid output, which reduces computational complexity by minimizing exponential operations. An SE module is incorporated at the network's final layer to improve attention to critical features through adaptive channel calibration. Additionally, a 5×5 convolution replaces the 3×3 convolution at the network's tail, enabling high-precision detection without compromising inference speed. The structure diagram of LCbackbone is provided below.

$$\text{ReLU6}(x) = \min(\max(0, x), 6) \quad (1)$$

$$H\text{-swish}(x) = x \frac{\text{ReLU6}(x + 3)}{6} \quad (2)$$

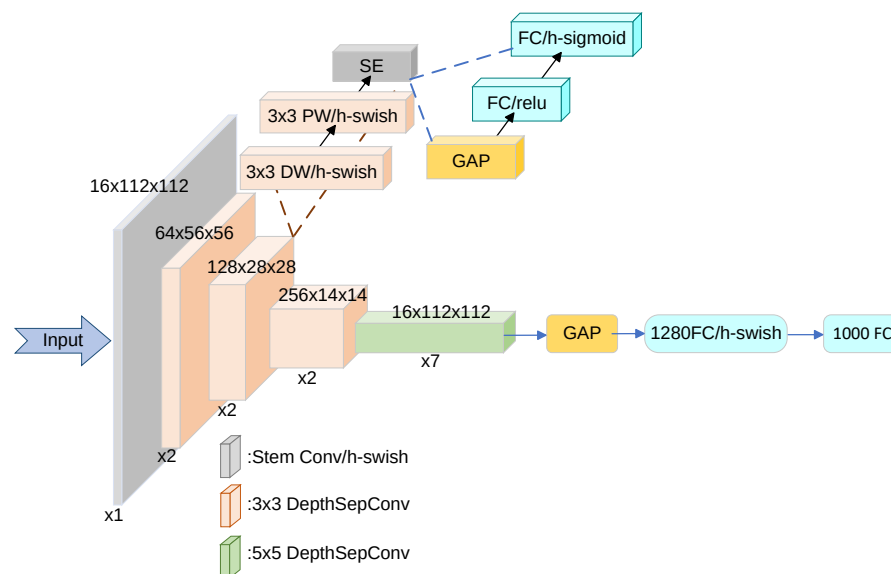


Figure 2. Improvement scheme at the backbone combined with the PP-LCNet. The dotted box represents optional modules. The Stem denotes the initial feature extraction layer built using a standard 3×3 convolution. DepthSepConv means depth-wise separable convolutions, DW means depth-wise convolution, PW means point-wise convolution, GAP means global average pooling.

2.2. Adaptive Downsampling Module in the Neck Layer

Accurately detecting small drones at long distances is highly challenging due to the presence of small targets and complex backgrounds. Improving detection accuracy and feature representation is, therefore, essential. In YOLOv11, the neck layer employs conventional convolution modules for feature map downsampling and multi-scale feature fusion. However, these convolutions often cause the loss of fine-grained information during downsampling and reduce feature map resolution, adversely affecting small object detection performance. To overcome this limitation, this paper introduces an efficient downsampling Adown module from YOLOv9 into the neck of YOLOv11 [27]. The core idea of Adown is to combine convolution and pooling for efficient downsampling, reducing information loss through effective feature fusion. This approach enables the efficient extraction of both global and local features while preserving the detailed characteristics of the target, thereby enhancing the capability for small object detection.

As illustrated in Figure 3, the Adown module begins by downsampling the input feature map using average pooling, reducing its size by half. The resulting feature map is divided into $\times 1$ and $\times 2$ parts along the channel dimension. A 3×3 convolution is applied to $\times 1$ for feature extraction and dimensionality reduction. Meanwhile, $\times 2$ undergoes max pooling followed by a 1×1 pointwise convolution to enhance nonlinear feature representation and further reduce dimensionality. The two processed feature maps are then combined to form the output of the Adown module. Unlike the conventional convolution-based downsampling in YOLOv11, the Adown module integrates both max pooling and average pooling, enabling more comprehensive feature extraction. Additionally, the Adown module employs a multi-branch structure, which enhances the network's flexibility and improves the capture of features across multiple scales. This refined downsampling approach boosts the model's capacity to extract and represent features more effectively.

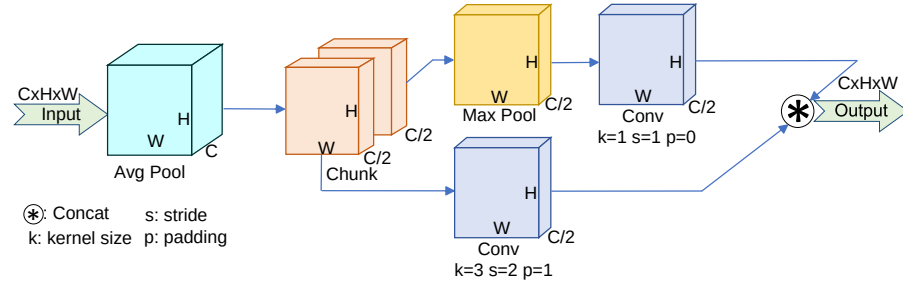


Figure 3. The structure of the adaptive downsampling module.

2.3. Mamba-Inspired Linear Attention Mechanism in Neck Layer

Drone detection tasks are often performed in dynamic environments, such as urban areas or natural settings, where high-resolution images captured by sensors frequently include complex backgrounds. These conditions place increased demands on the model's computational efficiency and ability to model global information. The neck layer of YOLOv11 integrates FPN and PAN structures to enhance object detection through multi-scale feature fusion. However, it primarily relies on a local receptive field, limiting its ability to capture global contextual information and reducing its effectiveness in dynamic background scenarios. Furthermore, with high-resolution inputs, the convolutional computations in the neck layer grow quadratically with input resolution. This increase leads to slower inference speeds, making it difficult to meet the requirements for real-time detection.

To address these challenges, this study introduces the MILA module [39], a novel linear attention mechanism. The module explicitly captures long-range dependencies in feature maps using attention operations with linear complexity and positional encoding. This design preserves spatial positional information, enabling efficient global information modeling while reducing computational costs. The MILA module incorporates a linear attention mechanism as its core component, simplifying the design by eliminating traditional multi-head attention. This approach reduces the computational overhead associated with multi-head calculations, significantly enhancing inference speed.

As shown in Figure 4, after the data are standardized, they enter the MLLA Block module. This module first undergoes a linear transformation through the Linear layer. The combination of linear layers, convolutional layers, and normalization operations within the MLLA Block enables the model to perform complex feature extraction and data processing. The data are then passed through the linear attention mechanism in the linear Attention module for global information aggregation. Finally, after passing through the Norm layer, the data enter the MLP module. The MLP, as a multilayer perceptron, contains multiple fully connected layers, which can perform nonlinear transformations on the data, further enhancing the model's expressive power and extracting richer features.

Specifically, the MILA module replaces conventional softmax attention with global linear attention, defined by the following formula:

$$y_i = \frac{Q_i \left(\sum_{j=1}^i K_j^T V_j \right)}{Q_i \left(\sum_{j=1}^i K_j^T \right)} \triangleq \frac{Q_i S_i}{Q_i Z_i}, S_i = \sum_{j=1}^i K_j^T V_j, Z_i = \sum_{j=1}^i K_j^T \quad (3)$$

Here, y_i represents the final attention output value, which is the output vector calculated by the model at the i -th time step or position. S_i denotes the weighted sum, which computes the weighted sum of all key–value pairs from 1 to i . Z_i is the normalization factor used to normalize the similarity between the query and the key. $Q_i = \phi(xW_Q)$ represents the query vector, derived from the linear mapping of input features using the query matrix W_Q . The query vector is primarily used to compute similarities with key vectors, thereby

determining the weight of the corresponding value vectors. Similarly, $K_j = \phi(xW_K)$ denotes the key vector, obtained via the key matrix W_K . Together with the query vector, the key vector is involved in the similarity computation, helping to identify which value vectors should be included in the output. $V_j = xW_V$ is the value vector, produced through the value matrix W_V . The weight of the value vector in the final output is determined by combining it with the similarity between the query and the key. The kernel function $\phi(\cdot)$ is used to enhance nonlinear features. The MILA module reduces the computational complexity of traditional Softmax attention from $O(N^2)$ to $O(N)$, significantly increasing efficiency in high-resolution image processing. It demonstrates excellent performance in handling high-resolution images. Furthermore, the MILA module introduces an input-dependent input gate that dynamically filters input features using the softplus function, optimizing selective feature representation. The dynamic input gate controls the contribution of each input to the hidden state at the current time step. The mathematical formula is as follows:

$$\Delta_i = \text{Softplus}(x_i W_1 W_2) \quad (4)$$

In this equation, Δ_i represents the importance weight of the input features, while W_1 and W_2 are weight matrices. This mechanism dynamically adjusts input feature weights, improving the network's focus on critical features. Furthermore, the MILA module replaces the forget gate with efficient positional encoding techniques, such as RoPE. This substitution captures local biases and positional information while removing the bottleneck associated with recursive computations. RoPE incorporates positional information into the token embeddings of the input sequence using a rotational transformation. This approach enables the model to capture relative positional information more effectively, eliminating the need for traditional recursive computations. The corresponding formula is as follows:

$$A_i = \exp(\Delta_i A) \quad (5)$$

Here, A_i represents the forget gate weight enhanced with positional encoding, while Δ_i controls the bias strength. This design improves the model's global perception capabilities and accelerates inference speed. Additionally, the MILA module incorporates shortcut connections in each submodule to directly map input features to output features, as described by the following formula. This approach enhances the network's training stability and facilitates gradient flow.

$$y_i = C_i h_i + D \odot x_i \quad (6)$$

where C_i represents the weight matrix mapping the hidden state to the output, D denotes the shortcut connection weight, and $\odot$ indicates element-wise multiplication. MILA is designed with a focus on lightweight efficiency. By incorporating pointwise optimizations such as linearized attention and dynamic input gating, it facilitates efficient feature interaction, enhancing detection accuracy while preserving inference efficiency.

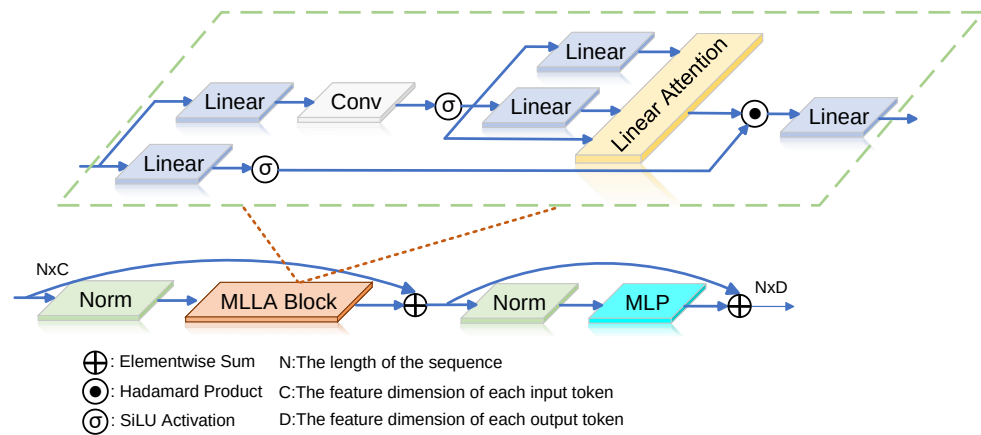


Figure 4. The structure of the MILA linear attention mechanism.

2.4. Improved Bounding Box Regression

In drone detection, targets in drone-captured scenes are often small and exhibit diverse shapes. The shape and scale of these small objects significantly affect bounding box regression accuracy. However, traditional loss functions (e.g., IoU, GIoU, SIoU) primarily emphasize the geometric relationship between predicted and ground truth boxes, neglecting the influence of shape and scale on regression precision. Despite being an advanced object detection algorithm, YOLOv11's bounding box regression module still relies on traditional loss functions. Consequently, it struggles to accurately measure alignment when there are substantial differences in the aspect ratios of the ground truth and predicted boxes. Furthermore, it fails to effectively leverage the shape and scale information of bounding boxes to guide the model toward faster and more accurate convergence. This study introduces Shape-IoU as a new regression loss function in YOLOv11 [40] to enhance bounding box regression performance. Shape-IoU improves alignment accuracy between predicted and ground truth boxes by incorporating shape and scale weights. As shown in Figure 5, B_{gt} represents the ground truth box, with (x_c^{gt}, y_c^{gt}) indicating the center coordinates of the ground truth box. B represents the predicted box, with (x_c, y_c) denoting the center coordinates of the predicted box. W_{gt} and h_{gt} represent the width and height of the ground truth box, respectively, while W and h represent the width and height of the predicted box. Shape-IoU extends the IoU formula by incorporating shape and scale weights of the bounding boxes, offering a more comprehensive measure of alignment. This is illustrated in Formula (7):

$$IoU = \frac{|B \cap B_{gt}|}{|B \cup B_{gt}|} \quad (7)$$

Building on this foundation, Shape-IoU models the effects of shape and scale by calculating horizontal and vertical weights based on the bounding box's aspect ratio:

$$ww = \frac{2 \cdot w_{gt}^{scale}}{w_{gt}^{scale} + h_{gt}^{scale}}, hh = \frac{2 \cdot h_{gt}^{scale}}{w_{gt}^{scale} + h_{gt}^{scale}} \quad (8)$$

Here, w_{gt}^{scale} and h_{gt}^{scale} denote the scale factors for the width and height of the ground truth box. Shape-IoU also incorporates the Euclidean distance between the center points of the predicted and ground truth boxes, along with the influence of their shapes:

$$dis \tan ce_{shape} = hh \cdot \frac{(x_c - x_c^{gt})^2}{c^2} + ww \cdot \frac{(y_c - y_c^{gt})^2}{c^2} \quad (9)$$

Here, (x_c, y_c) and (x_c^{gt}, y_c^{gt}) denote the center points of the predicted box and the ground truth box, respectively, while c represents the maximum diagonal distance of the boxes. A shape deviation term is then introduced to quantify the shape differences between the predicted and ground truth boxes:

$$\Omega_{shape} = \sum_{t=w,h} (1 - e^{-w_t}) \theta, \theta = 4 \quad (10)$$

$$w_w = ww \cdot \frac{|w - w_{gt}|}{\max(w, w_{gt})}, w_h = hh \cdot \frac{|h - h_{gt}|}{\max(h, h_{gt})} \quad (11)$$

The complete Shape-IoU loss function is expressed as follows:

$$L_{Shape-IoU} = 1 - IoU + \text{distance}_{shape} + 0.5 \cdot \Omega_{shape} \quad (12)$$

In summary, Shape-IoU models both the shape and scale of target bounding boxes, enabling more accurate regression that better reflects the geometric characteristics of the ground truth boxes and thereby improves detection accuracy. By incorporating shape and scale weights, it also enhances the robustness of the bounding box regression. Furthermore, utilizing shape and scale information accelerates convergence and increases training efficiency.

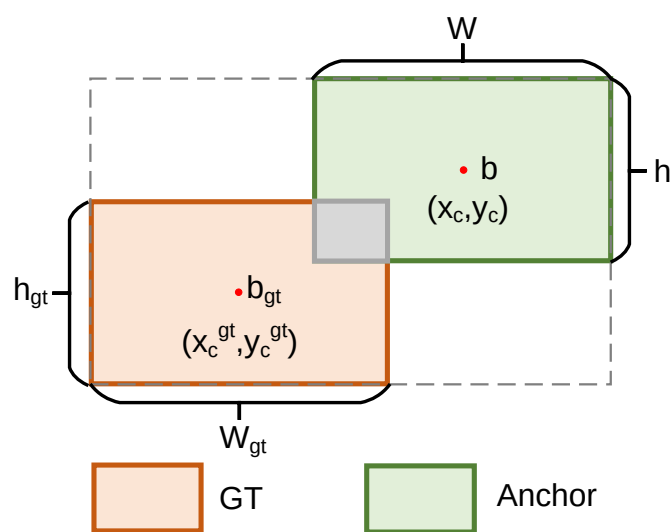


Figure 5. The scheme of Shape-IoU loss function.

As shown in Figure 6, this study introduces an optimized and enhanced YOLOv11 network, designed to be lightweight and efficient: 1. A lightweight CPU network utilizing the MKLDNN acceleration strategy was developed as the backbone, reducing model parameters and computational complexity. 2. The Adown module was added to the network's neck layer, enabling efficient extraction of both local and global features while minimizing computational costs. 3. An improved attention mechanism is employed, integrating positional encoding at the neck layer to enhance the capability for global information modeling. 4. Shape-IoU was adopted as the loss function to accelerate model convergence and improve training efficiency.

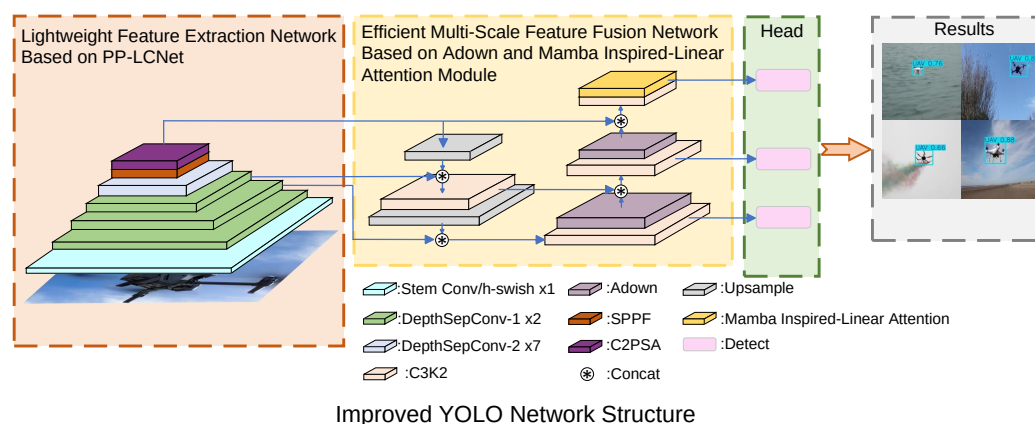


Figure 6. Optimization techniques in LAMS-YOLO include utilizing PP-LCNet as the backbone. Furthermore, the Adown incorporates the Mamba-inspired linear attention mechanism to create an enhanced feature fusion network.

3. Experimental Setup and Analyses

With the growing use of civilian drones, their applications in daily life are rapidly expanding. Publicly available datasets offer diverse drone images captured under various environmental conditions. However, many datasets are constrained by simple backgrounds and lack sufficient information on different drone postures in complex scenarios, which can hinder model accuracy in drone detection. To bridge this gap, this study enhances existing datasets by manually screening drones under various contexts from publicly available online datasets. To improve the model's ability to accurately recognize drones and prevent misclassification of common birds as drones, the dataset used in this study is divided into two categories: birds and drones. In the detection airspace, drones and birds exhibit distinct visual features. As shown in Figure 7, drones generally have clear geometric shapes, such as rectangular, square, or circular bodies, along with quadcopter structures. These features result in outlines that display strong symmetry and regularity in images. In contrast, birds have more complex and irregular shapes, with varied wing and head contours, a broad wingspan, and dynamic morphological changes during flight.

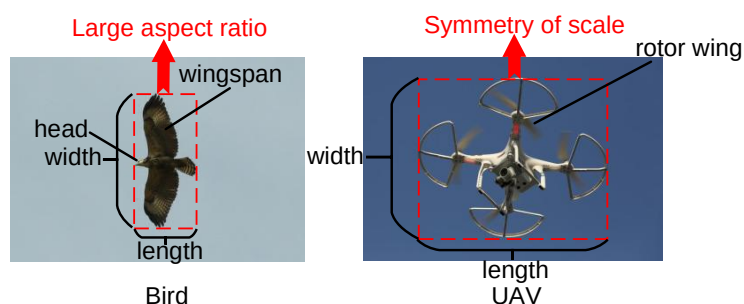


Figure 7. Comparison of features between birds and drones.

Bird images are sourced from public datasets, including 100 bird species from Kaggle, birds from the CoCo dataset, and images from the cub-200-2011 bird dataset. Drone images are drawn from the drone vs. bird dataset and DUT Anti-UAV dataset [41]. The dataset comprises 5288 images for training and 1700 images for validation, with the training set containing 1438 bird images and 3850 drone images. The validation set includes 348 bird images and 1352 drone images. The drone's flight altitude, measured from the ground, ranges from 10 to 100 m. The dataset is labeled using a text file with five columns: category label, center coordinates (x and y) of the bounding box, width, and height.

Figure 8 shows the data distribution and bounding box specifications for the drone category in the training set. To maintain consistency in controlled experiments, identical experimental environments were applied across all models, with an input image size of 640×640 .

The experimental environment in this paper is based on the Ubuntu 20.04 operating system, with an Tensor Core A100 GPU, 40 GB of memory. The programming language used is Python 3.9.11, and the deep learning model is built using PyTorch 1.10.0, cudnn 8.2.0, and torchvision 0.12.0. The computational library used is numpy 1.23.3, and parallel computing is supported by the NVIDIA CUDA Toolkit 10.1.0. The code is available at <https://github.com/Surprise-Zhou/LAMS-YOLO> (accessed on 5 February 2025).

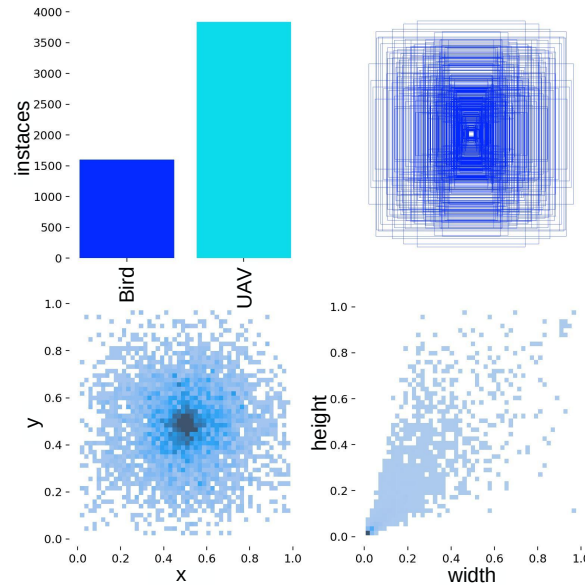


Figure 8. Label volume and distribution for drone detection.

3.1. Experimental Setup and Accuracy Evaluation

To comprehensively evaluate the drone fault detection model, four metrics were used: precision (P), recall (R), $F1$ score, and mAP . The formulas for calculating these metrics are as follows:

$$P = \frac{TP}{TP + FP} \quad (13)$$

$$R = \frac{TP}{TP + FN} \quad (14)$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \quad (15)$$

$$mAP = \frac{\sum_{q=1}^Q AP(q)}{Q} \quad (16)$$

TP represents true positives, where drones are correctly detected. FP indicates false positives, or incorrect detections of non-existent drones. FN refers to false negatives, where drones are missed. Precision (P) measures correct detections among positives, while recall (R) assesses correctly identified drones among actual targets. The $F1$ score balances precision and recall, and mean average precision (mAP) averages AP across categories for overall performance. Model parameters, such as weights and biases, are also analyzed to evaluate complexity and computational needs. GFLOPs quantifies the number of floating-point operations performed per second, serving as an indicator of the computational complexity and efficiency of deep learning models. Model size represents the storage space required

for the model, while parameters refer to the total number of trainable parameters, including weights and biases. Enhancements in feature extraction improve the model's precision and recall, leading to higher mAP and F1 scores. Optimizing the model architecture increases its expressiveness and flexibility, thereby improving precision, recall, and mAP. Additionally, an efficient network architecture reduces both GFLOPs and parameters. The loss function design plays a key role in balancing precision and recall, which ultimately enhances the F1 score.

3.2. Evaluation of Model Lightweighting and Detection Accuracy

To validate the effectiveness of LAMS-YOLO, the improved YOLO model was compared with the baseline model using the same training dataset. As shown in Table 1, LAMS-YOLO outperformed the baseline model across all performance metrics, including mean average precision (mAP50 and mAP95), recall, and F1 score, with improvements of 3.89%, 4.04%, 6.18%, and 3.11%, respectively. The observed improvements can be primarily attributed to the newly designed linear attention mechanism module. This module enhances feature extraction by incorporating a forget gate to introduce local bias and positional information, coupled with an optimized block design. Additionally, the adaptive downsampling mechanism in the neck layer improves downsampling efficiency through effective feature fusion, minimizing the loss of critical information.

Table 1. Comparison of the YOLOv11n and LAMS-YOLO on the test data of the self-built dataset.

Methods	Precision (%)	Recall (%)	F1 (%)	mAP50 (%)	mAP95 (%)	Parameters (M)	Model Size (MB)	GFLOPs
YOLOv11n	96.3	84.1	89.79	89.9	71.8	2.583	5.358	6.3
LAMS-YOLO	96.1	89.3	92.58	93.4	74.7	2.332	4.857	5.2

LAMS-YOLO achieves a compact model size of 4.857 MB and a parameter count of 2.33 M. Compared to the original YOLOv11n baseline, this represents a parameter reduction of 9.35% and 9.69%, respectively. This improvement is primarily due to the introduction of the LCBackbone module, which utilizes depthwise separable convolutions as its core building blocks, significantly reducing the model's parameter count.

Experimental results confirm the effectiveness of the proposed lightweight design, making LAMS-YOLO particularly suitable for micro-embedded systems. To further evaluate model performance, PR curves were generated during testing at an IoU threshold of 0.5, comparing results before and after the improvements, as shown in Figure 9.

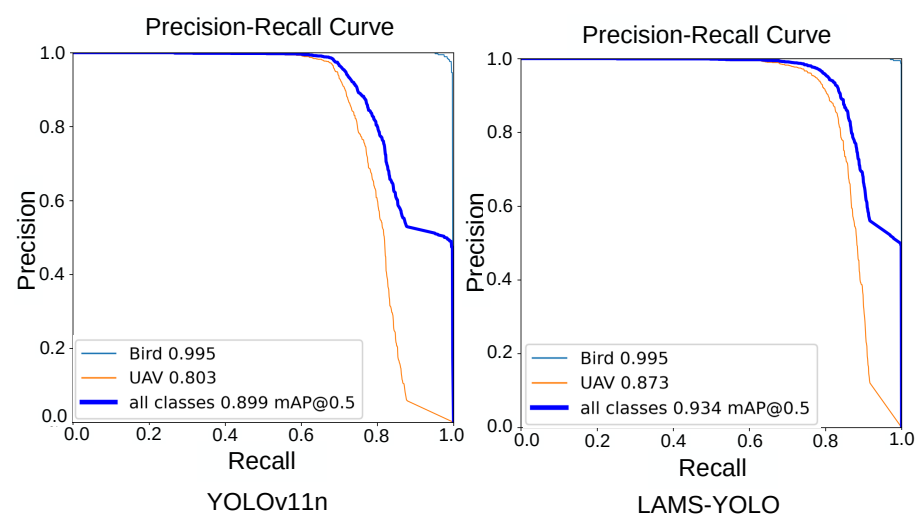


Figure 9. The PR diagram between YOLOv11n and LAMS-YOLO.

The precision–recall area under the curve (AUC-PR) is a widely used metric for evaluating model performance. A higher AUC-PR indicates superior performance across various precision–recall combinations. The improved model achieves a significantly higher AUC-PR.

3.3. Ablation Experiments

To evaluate the detection performance of our innovative designs, “MILA”, “Adown”, “PP-LCNet”, and “Shape-IoU”, this paper conducted ablation experiments. These experiments assessed the impact of each algorithmic improvement, including model simplification, attention mechanisms, and multi-level feature integration. The evaluation metrics included mean average precision (mAP) and model size. Table 2 and Figure 10 present the results of LAMS-YOLO on the custom dataset under different optimization strategies.

Table 2. Results of ablation experiments on the self-built dataset.

Methods	MILA	Adown	Shape-IoU	PP-LCNet	mAP50 (%)	mAP95 (%)	Parameters (M)	GFLOPs	Model Size (MB)
YOLOv11n					89.9	71.8	2.583	6.3	5.358
Methods (1)	✓				91.2	73.0	2.717	6.4	5.595
Methods (2)		✓			91.6	73.3	2.449	6.1	5.091
Methods (3)		✓ *			90.1	71.9	2.490	6.3	5.178
Methods (4)				✓	90.7	72.3	2.331	5.2	4.875
Methods (5)	✓	✓			91.6	73.5	2.584	6.3	5.337
Methods (6)	✓			✓	93.1	74.3	2.465	5.3	5.113
Methods (7)	✓	✓	✓		91.7	73.3	2.584	6.3	5.339
Methods (8)	✓	✓	✓	✓	93.4	74.5	2.332	5.2	4.857

* Indicates that the original Adown module’s 3×3 and 1×1 convolutions are replaced with two identical 3×3 convolutions.

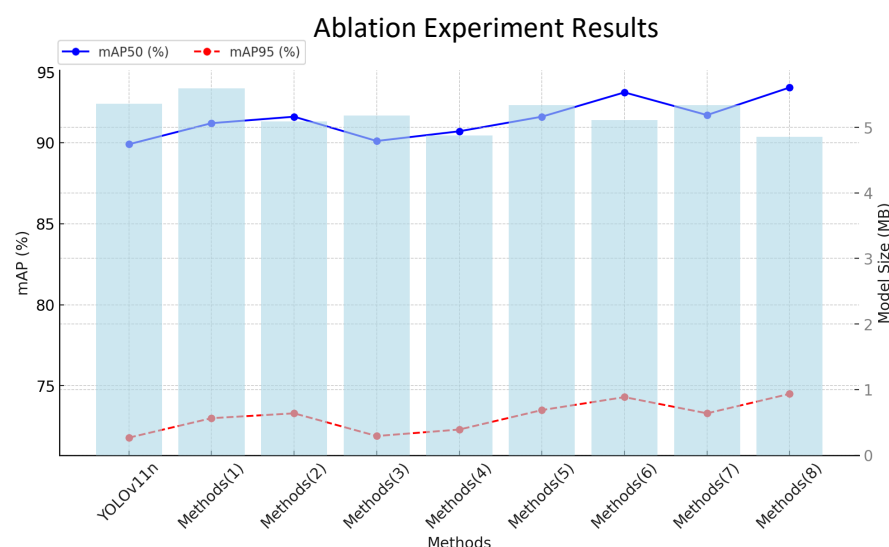


Figure 10. Comparison of mAP and model size with different methods.

As presented in Table 2 and Figure 10, method (1) increased mAP50 by 1.45% compared to the baseline model, with a slight model size increase of 0.237 MB. This improvement is attributed to the introduction of a linear attention mechanism that utilizes rotational positional encoding (RoPE) in place of absolute positional encoding, allowing the model to better retain positional information while effectively capturing global dependencies.

Method (2) reduces the model size by 0.267 Mb and increases mAP by 1.89% compared to the baseline model. This improvement is attributed to the Adown module, which divides the input feature map into two parts: one part is processed through convolution for feature extraction, while the other is downsampled using pooling. The two parts are then fused.

This design reduces the spatial dimensions of the feature map and mitigates potential information loss typically associated with traditional convolution-based downsampling. As a result, the model's parameters are reduced, and detection accuracy is enhanced.

Method (3) builds on Method (2) by replacing the two convolution layers in the Adown module with uniform 3×3 convolutions, replacing the original hybrid design that combined both 3×3 and 1×1 convolutions. This modification converts the multi-scale convolution into a single-scale convolution module. Method (2) outperforms Method (3) by 1.66% in mAP, indicating the effectiveness of the multi-scale feature extraction and downsampling operation in Method (2).

Method (4) reduces the model parameters by 9.76% while improving mAP by 0.89% compared to the baseline model. This demonstrates that the enhanced Backbone module achieves both model reduction and improved detection accuracy.

Method (5), building on method (1), improved mAP50 by an additional 0.43% while reducing the model size by 0.258 MB. This was achieved through the inclusion of an adaptive downsampling module, which employs dynamic convolution and multi-scale fusion to compress features efficiently while preserving essential information, thereby minimizing redundant computations.

Method (6) achieved a 2.08% increase in mAP50 over method (1) while reducing model parameters by 8.61%. This highlights the effectiveness of the improved Backbone module, which incorporates depthwise separable convolutions and the H-Swish activation function to lower computational complexity. The addition of the SE module further optimized performance by enhancing inter-channel feature interactions.

Method (7), based on Method (5), delivered an additional 0.11% increase in mAP50 with only a 2 KB increase in model size. This improvement is attributed to an enhanced bounding box regression loss function, which introduces a shape similarity constraint, enabling the model to produce predicted boxes that more closely align with ground truth boxes.

Method (8), building upon Method (7), improved mAP50 by 1.85% while reducing model size by 0.482 MB. This demonstrates the effectiveness of combining "MILA", "Adown", "LCNet", and "Shape-IoU" in achieving a balance between linear attention mechanisms, adaptive downsampling, bounding box regression, and model lightweighting. Consequently, the LAMS-YOLO model adopts the structure of Method (5) to ensure optimal performance.

On the custom dataset, LAMS-YOLO achieved a remarkable mAP50 of 93.4% with a compact model size of 4.857 MB, representing a 3.89% improvement in mAP50 and a 0.50 MB reduction in model size compared to the YOLOv11n baseline.

This paper conducted an interpretability analysis of the model improvement strategy using Grad-CAM [42] implemented in PyTorch. Figure 11 shows the Grad-CAM-generated heat maps for YOLOv11n and LAMS-YOLO on the drone dataset. LAMS-YOLO demonstrates a superior focus on target regions. It also reduces attention to irrelevant background areas and non-target regions compared to YOLOv11n. Additionally, for targets of varying sizes, the LAMS-YOLO algorithm more effectively concentrates on positive sample regions and reduces attention to irrelevant environmental details. The results demonstrate two key improvements. First, the linear attention mechanism enhances the network's ability to represent information. Second, the adaptive downsampling structure significantly improves the model's detection performance for drone targets across different scales.

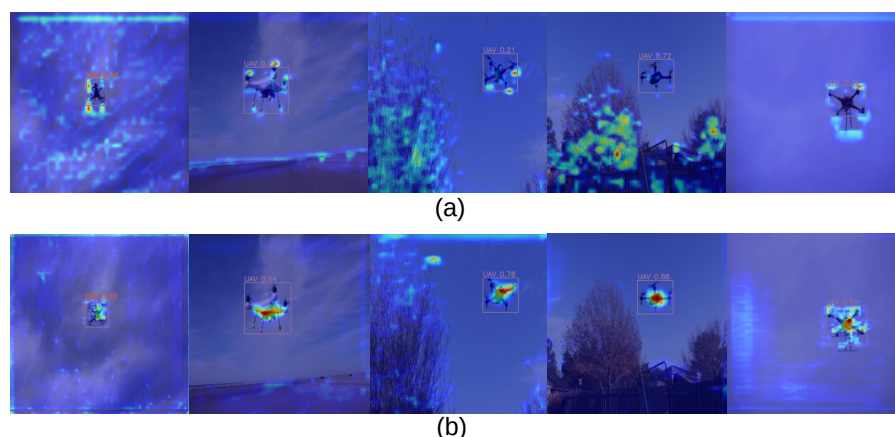


Figure 11. Heat maps of YOLOv11n and LAMS-YOLO for drones obtained by Grad-CAM. (a) Heat maps generated by YOLOv11n for drones on the self-built dataset. (b) Heat maps generated by LAMS-YOLO for drones on the self-built dataset.

3.4. Comparison with State-of-the-Art Methods

This study evaluates the proposed LAMS-YOLO method against the baseline YOLOv10n model and several established object detection algorithms, including Faster-RCNN, SSD, EfficientDet, YOLOv5, YOLOv6, YOLOv9-Tiny, and YOLOv10. The comparison uses performance metrics such as mean average precision (mAP) per category, recall, F1 score, and precision. Precision and mAP values were computed following the PASCAL VOC 2007 benchmark, with the intersection over union (IoU) threshold set at 0.5. Table 3 and Figure 12 summarize the results obtained on the custom dataset. The results show that LAMS-YOLO achieved an mAP of 93.4%, indicating a notable improvement in detection accuracy compared to the YOLOv11 baseline network. Furthermore, the mAP of LAMS-YOLO surpasses that of the YOLOv10 algorithm by 2.9% and the lightweight YOLOv9-Tiny algorithm by 4.9%.

Compared to SSD, EfficientDet, YOLOv5, and YOLOv6, LAMS-YOLO show significant overall performance improvements of 29.1%, 67.5%, 5.7%, and 4.7%, respectively. Notably, Faster-RCNN and RT-DETR did not converge on the drone dataset, indicating that not all end-to-end detection algorithms are suitable for drone detection.

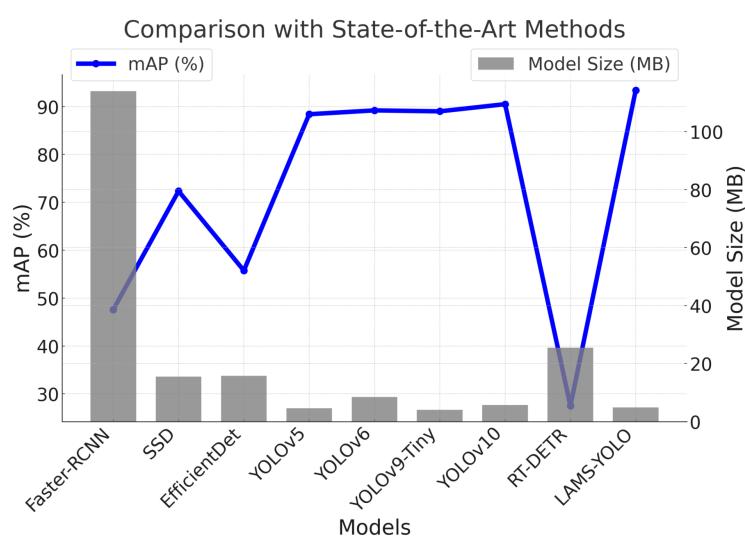


Figure 12. Comparative experiments with different models on the self-built dataset.

Table 3. Comparative experiments with different models on the self-built dataset.

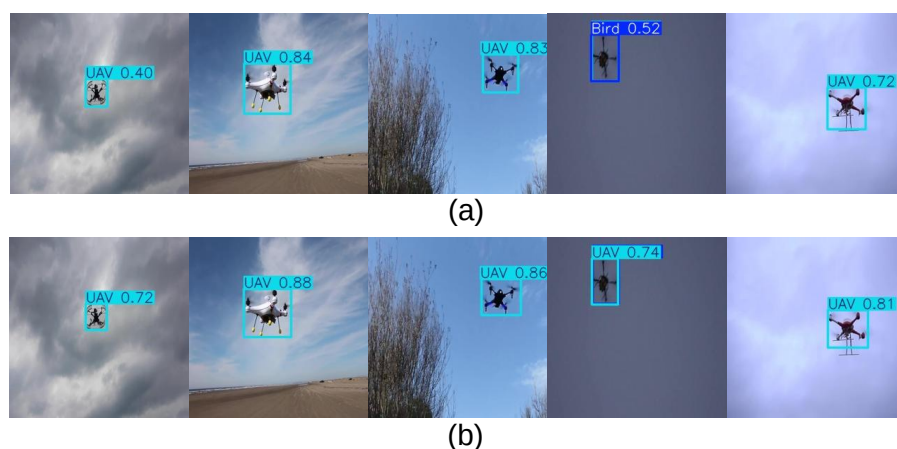
Models	Precision (%)	Recall (%)	F1 (%)	mAP50 (%)	FPS	Model Size (MB)
Faster-RCNN	87.40	46.3	60.53	47.6	4.98	114
SSD	87.60	54.00	66.81	72.35	36.65	15.5
EfficientDet	98.50	53.07	68.98	55.75	28.40	15.8
YOLOv5	94.04	83.02	88.19	88.4	109.25	4.6
YOLOv6	93.94	83.93	88.65	89.2	66.83	8.5
YOLOv9-Tiny	93.41	83.93	88.42	89.0	107.19	4.1
YOLOv10	90.55	84.20	87.26	90.5	99.67	5.7
RT-DETR	32.10	38.80	35.13	27.5	22.28	25.5
LAMS-YOLO	96.10	89.30	92.58	93.4	115.94	4.9

This study evaluates the proposed LAMS-YOLO method by comparing it with other lightweight algorithms that replace the YOLOv11 backbone, aiming to demonstrate the superiority of LAMS-YOLO in terms of lightweight design. As shown in Table 4, LAMS-YOLO outperforms lightweight networks such as C3Ghost, DwConv, and GhostConv, improving mAP by 4.44%, 4.73%, and 4.50%, respectively, while reducing the number of parameters by 0.85%, 2.92%, and 6.80%. These results clearly indicate that the lightweight object detection algorithm presented in this study surpasses other lightweight model architectures.

Table 4. Comparative experiments with different lightweight models on the self-built dataset.

Model	Precision (%)	Recall (%)	mAP50 (%)	Parameters (%)	GFLOPs
C3Ghost-YOLOv11	93.40	83.53	89.43	2.35	6.1
DwConv-YOLOv11	92.43	84.21	89.18	2.40	6.2
GhostConv-YOLOv11	94.15	82.93	89.38	2.50	6.3
LAMS-YOLO	96.1	89.3	93.40	2.33	5.2

Figure 13 illustrates a comparison of detection results between YOLOv11n and LAMS-YOLO on the custom dataset. YOLOv11n shows inconsistent confidence levels for targets in complex backgrounds. In contrast, LAMS-YOLO demonstrates significantly higher confidence levels and fewer instances of drones being misclassified as birds, leading to improved detection accuracy. This enhancement is primarily due to the integration of the linear attention mechanism and adaptive downsampling structure, which effectively strengthen spatial information representation.

**Figure 13.** Visual comparison of detection results between YOLOv11n and LAMS-YOLO on the self-built dataset. (a) YOLOv11n. (b) LAMS-YOLO.

4. Discussion

In drone detection, existing object detection models encounter challenges such as limited recognition capabilities in complex backgrounds and inadequate feature extraction in high-resolution tasks. While YOLOv11 performs well in one-stage object detection, its reliance on numerous convolutional operations, many of which are inefficient, imposes a significant computational burden. This limitation hinders its real-time detection capability for drones. To address these challenges, this study presents LAMS-YOLO, a lightweight object detection model specifically designed for multi-scale, high-precision, and low-latency drone detection in complex environments.

Specifically, this study enhances the conventional convolutions in the YOLOv11n backbone using LCbackbone. This architecture, centered on depthwise separable convolutions and the H-Swish activation function, optimizes computational efficiency and feature extraction capabilities, enabling the model to operate efficiently on resource-constrained devices. Additionally, a novel linear attention mechanism is proposed, incorporating a forget gate to dynamically filter critical information, thereby improving the model's ability to focus on targets in complex backgrounds. Rotational positional encoding is used instead of absolute positional encoding, allowing the model to more accurately preserve positional information while capturing global dependencies. Furthermore, an adaptive downsampling module is introduced in the neck layer. This module uses dynamic convolutions and feature fusion to achieve efficient feature compression while retaining essential information. Finally, an improved loss function incorporates a shape similarity measure for target bounding boxes, enhancing localization accuracy.

Deep learning algorithms surpass traditional machine learning methods in object detection by offering superior feature representation and the ability to process large-scale data. These algorithms are broadly divided into one-stage and two-stage approaches. One-stage detection algorithms typically provide faster inference speeds and require fewer computational resources than two-stage methods. Among these, YOLOv11 strikes an effective balance between detection accuracy and inference speed, making it suitable for diverse detection tasks. This study presents the one-stage LAMS-YOLO model and evaluates its performance against traditional two-stage detection methods, such as Faster-RCNN and SSD, as well as lightweight one-stage approaches, including YOLOv5, YOLOv6, YOLOv9-Tiny, and YOLOv10. As shown in Table 3, LAMS-YOLO surpasses all compared algorithms across evaluation metrics.

Faster-RCNN extracts features from entire images before generating region proposals. However, its complex structure leads to low detection accuracy for drones and fails to meet real-time requirements, as reflected in its F1 score and high parameter count. Similarly, the SSD model shows limited detection accuracy due to its reliance on predefined anchor boxes, which restrict its ability to capture drone variations. Additionally, RT-DETR fails to converge on the dataset, further highlighting its limitations.

In contrast, YOLOv5 improves multi-scale detection performance by leveraging CSP modules and FPN feature fusion techniques. However, its high computational complexity in high-resolution scenarios restricts its real-time applicability. YOLOv6 enhances detection accuracy through a deeper backbone network and large-scale pre-training, but its high parameter count limits use in resource-constrained settings. YOLOv9-Tiny focuses on lightweight design, achieving faster inference but lacking adaptability to complex backgrounds and multi-object detection, resulting in frequent missed and false detections. YOLOv10 improves detection accuracy with an enhanced feature fusion module but suffers from high computational costs, limiting its suitability for real-time drone detection. In comparison, LAMS-YOLO offers a lightweight design with improved detection efficiency

and enhanced performance in identifying drone targets within complex backgrounds, presenting an innovative solution for drone detection.

However, this study has several limitations. First, the dataset used does not include complex weather conditions such as strong winds, heavy fog, and snowfall, leaving the model's performance in adverse environments untested. Second, although the adaptive downsampling module enhances feature retention during downsampling through multi-scale fusion, it still has limitations in extracting detailed information from small objects. Its receptive field may be insufficient for capturing features of extremely small targets. Figure 14 indicates that LAMS-YOLO exhibits lower detection accuracy for long-range small object detection, with limitations such as false positives and missed detections. Finally, while the loss function optimizes bounding box regression for complex objects, in scenarios with high-density small targets, the loss signal for small targets may be weakened during training, potentially affecting detection performance for such objects.

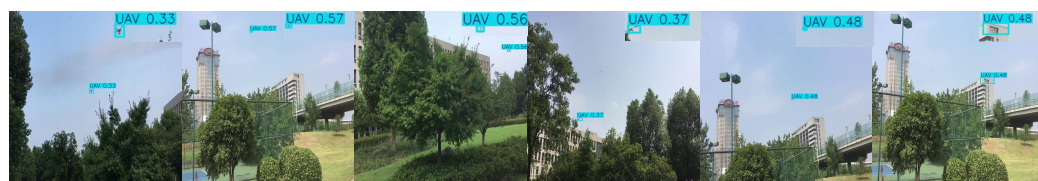


Figure 14. Visual detection results based on LAMS-YOLO in small drone object detection.

In future research, the dataset should be expanded to include complex extreme weather conditions and diverse scenarios to enhance the model's generalization capability across different environments. Additionally, the receptive field design of the downsampling module should be further optimized, incorporating a dynamic receptive field expansion mechanism to improve feature extraction for small and distant targets. Moreover, integrating super-resolution techniques with object detection tasks can generate higher-quality target features during training, thereby improving the detection accuracy of small objects. Lastly, introducing enhanced mechanisms that combine channel attention and self-attention into the network structure can improve the model's ability to capture critical features. Simultaneously, optimizing the loss function during training by reweighting the loss signals of small targets could further enhance the model's detection performance for small objects. These improvements would provide a more robust solution for efficient object detection in drone-related tasks.

5. Conclusions

To overcome the challenges of limited accuracy in multi-scale drone detection, inadequate real-time performance, and high computational complexity, this study presents LAMS-YOLO. This lightweight model incorporates a linear attention mechanism and an adaptive downsampling module, making it highly suitable for deployment in embedded systems within demanding environments. The algorithm integrates lightweight network components, a linear attention mechanism, and adaptive downsampling to balance streamlined design with high accuracy. Depthwise separable convolutions and efficient activation functions are incorporated into the backbone layer, significantly reducing model parameters and optimizing for lightweight performance. Additionally, a novel linear attention mechanism and an adaptive downsampling module are introduced in the neck layer to enhance detection capabilities. An improved loss function further refines shape matching, particularly optimizing bounding boxes for drones with complex geometries. Results on the custom dataset demonstrate that this method delivers robust detection performance for drone targets in challenging environments. However, the model's generalization remains limited under adverse conditions such as heavy snow, storms, and intense light, especially

for detecting small targets. Future work will focus on expanding drone detection datasets to include extreme weather scenarios. Moreover, integrating super-resolution techniques to improve small-target detection will be a key area of exploration.

Author Contributions: Conceptualization, S.Z. (Sicheng Zhou); methodology, S.Z. (Sicheng Zhou); software, S.Z. (Sicheng Zhou) and L.Y.; validation, H.L., C.Z. and J.L.; formal analysis, S.Z. (Sicheng Zhou); resources, S.Z. (Sicheng Zhou); data curation, K.W.; writing—original draft preparation, S.Z. (Sicheng Zhou); writing—review and editing, S.Z. (Shuai Zhao); visualization, S.Z. (Sicheng Zhou); supervision, K.W.; project administration, S.Z. (Shuai Zhao); funding acquisition, K.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (61935008), the China Postdoctoral Science Foundation (2024M753120), the joint funding from the National Synchrotron Radiation Laboratory (NO. KY2090000080).

Data Availability Statement: The data pertinent to this research are available from the corresponding authors upon request. These data are not publicly accessible as they are derived from lab results.

Acknowledgments: The authors appreciate the editors and anonymous reviewers for their valuable recommendations.

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

Abbreviations

LSTM	Long Short-Term Memory
UAV	Unmanned Aerial Vehicle
SVM	Support Vector Machine
ADr	Amateur Drone
PERDET	Perception-data-based UAV GPS Spoofing Detection
RPN	Region Proposal Network
MAVFE	Multiattentive Voxel Feature Encoding Module
RoI	Region of Interest
MSRGP	Multiscale Region of Interest (RoI)-grid Pooling
mAP	Mean Average Precision
FPS	Frames Per Second
QOPN	Quality-Oriented Proposal Network
ARL	Adaptive Recognition Loss
SSD	Single Shot MultiBox Detector
CA	Coordinate Attention
BiFPN	Bidirectional Feature Pyramid Network
SCAFPN	Sparsely Connected Asymptotic Feature Pyramid Network
CSL-MHSA	Cross-Space Learning Multi-Head Self-Attention mechanism
CIAM	Context Information Augmentation Module
TAM	Triplet Attention Module
NMS	Non-Maximum Suppression
SE	Squeeze-and-Excitation
FPN	Feature Pyramid Network
PAN	Path Aggregation Network
MILA	Mamba-Inspired Linear Attention
RoPE	Rotary Position Embedding
LePE	Locally-enhanced Positional Encoding
GFLOPs	Giga Floating Point Operations per Second
IoU	Intersection over Union

PR	Precision–Recall
AUC-PR	Precision–Recall–Area Under the Curve
Grad-CAM	Gradient-weighted–Class Activation Mapping

References

1. Zhang, J.; Campbell, J.F.; Sweeney, D.C., II. A Continuous Approximation Approach to Integrated Truck and Drone Delivery Systems. *Omega-Int. J. Manag. Sci.* **2024**, *126*, 103067. [\[CrossRef\]](#)
2. Estevez, J.; Nunez, E.; Lopez-Guede, J.M.; Garate, G. A Low-Cost Vision System for Online Reciprocal Collision Avoidance with UAVs. *Aerosp. Sci. Technol.* **2024**, *150*, 109190. [\[CrossRef\]](#)
3. Saadaoui, F.Z.; Cheggaga, N.; Djabri, N.E.H. Multi-Sensory System for UAVs Detection Using Bayesian Inference. *Appl. Intell.* **2023**, *53*, 29818–29844. [\[CrossRef\]](#)
4. Nwaogu, J.M.; Yang, Y.; Chan, A.P.C.; Chi, H.I. Application of Drones in the Architecture, Engineering, and Construction (AEC) Industry. *Autom. Constr.* **2023**, *150*, 104827. [\[CrossRef\]](#)
5. Meng, Z.; Zhou, Y.; Li, E.Y.; Peng, X.; Qiu, R. Environmental and Economic Impacts of Drone-Assisted Truck Delivery under the Carbon Market Price. *J. Clean. Prod.* **2023**, *401*, 136758. [\[CrossRef\]](#)
6. Lee, S.; Hong, D.; Kim, J.; Baek, D.; Chang, N. Congestion-Aware Multi-Drone Delivery Routing Framework. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9384–9396. [\[CrossRef\]](#)
7. Fotouhi, A.; Ming, D.; Hassan, M. DroneCells: Improving spectral efficiency using drone-mounted flying base stations. *J. Netw. Comput. Appl.* **2021**, *174*, 102895. [\[CrossRef\]](#)
8. Tepylo, N.; Debelle, L.; Laliberte, J. Public Perception of Remotely Piloted Aircraft Systems in Canada. *Technol. Soc.* **2023**, *73*, 102242. [\[CrossRef\]](#)
9. Al-Iqubaydhi, N.; Alenezi, A.; Alanazi, T.; Senyor, A.; Alanezi, N.; Alotaibi, B.; Alotaibi, M.; Razaque, A.; Hariri, S. Deep Learning for Unmanned Aerial Vehicles Detection: A Review. *Comput. Sci. Rev.* **2024**, *51*, 100614. [\[CrossRef\]](#)
10. Oh, B.S.; Lin, Z. Extraction of Global and Local Micro-Doppler Signature Features from FMCW Radar Returns for UAV Detection. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 1351–1360. [\[CrossRef\]](#)
11. Rudys, S.; Ragulis, P.; Laucys, A.; Brucas, D.; Pomarnacki, R.; Plonis, D. Investigation of UAV Detection by Different Solid-State Marine Radars. *Electronics* **2022**, *11*, 2502. [\[CrossRef\]](#)
12. Alvarez Lopez, Y.; Garcia Fernandez, M.; Las-Heras Andres, F. Comment on the Article “a Lightweight and Low-Power UAV-borne Ground Penetrating Radar Design for Landmine Detection”. *Sensors* **2020**, *20*, 3002. [\[CrossRef\]](#)
13. Zheng, J.; Chen, R.; Yang, T.; Liu, X.; Liu, H.; Su, T.; Wan, L. An Efficient Strategy for Accurate Detection and Localization of UAV Swarms. *IEEE Internet Things J.* **2021**, *8*, 15372–15381. [\[CrossRef\]](#)
14. Wang, Q.; Xu, H.; Lin, S.; Zhang, J.; Zhang, W.; Xiang, S.; Gao, M. A Low-Slow-Small UAV Detection Method Based on Fusion of Range-Doppler Map and Satellite Map. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 4767–4783. [\[CrossRef\]](#)
15. Hu, N.; Li, Y.; Pan, W.; Shao, S.; Tang, Y.; Li, X. Geometric Distribution of UAV Detection Performance by Bistatic Radar. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 2445–2452. [\[CrossRef\]](#)
16. Sipos, D.; Gleich, D. A Lightweight and Low-Power UAV-borne Ground Penetrating Radar Design for Landmine Detection. *Sensors* **2020**, *20*, 2234. [\[CrossRef\]](#)
17. Lee, S.; Kim, B. Machine Learning Model for Leak Detection Using Water Pipeline Vibration Sensor. *Sensors* **2023**, *23*, 8935. [\[CrossRef\]](#)
18. Sayed, A.N.; Ramahi, O.M.; Shaker, G. Machine Learning for UAV Classification Employing Mechanical Control Information. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 68–81. [\[CrossRef\]](#)
19. Anwar, M.Z.; Kaleem, Z.; Jamalipour, A. Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2526–2534. [\[CrossRef\]](#)
20. Wei, X.; Wang, Y.; Sun, C. PerDet: Machine-Learning-Based UAV GPS Spoofing Detection Using Perception Data. *Remote Sens.* **2022**, *14*, 4925. [\[CrossRef\]](#)
21. Feng, C.; Xiang, C.; Xie, X.; Zhang, Y.; Yang, M.; Li, X. HPV-RCNN: Hybrid Point-Voxel Two-Stage Network for LiDAR Based 3-D Object Detection. *IEEE Trans. Comput. Soc. Syst.* **2023**, *10*, 3066–3076. [\[CrossRef\]](#)
22. Li, Z.; Chen, H.; Biggio, B.; He, Y.; Cai, H.; Roli, F.; Xie, L. Toward Effective Traffic Sign Detection via Two-Stage Fusion Neural Networks. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 8283–8294. [\[CrossRef\]](#)
23. Li, W.; Zhao, D.; Yuan, B.; Gao, Y.; Shi, Z. PETDet: Proposal Enhancement for Two-Stage Fine-Grained Object Detection. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5602214. [\[CrossRef\]](#)
24. 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. [\[CrossRef\]](#)

25. YOLO9000: Better, Faster, Stronger [arXiv]-All Databases. Available online: <https://www.webofscience.com/wos/allldb/full-record/INSPEC:16848896> (accessed on 3 January 2025).
26. YOLOv4: Optimal Speed and Accuracy of Object Detection [arXiv]-All Databases. Available online: <https://www.webofscience.com/wos/allldb/full-record/INSPEC:19672657> (accessed on 3 January 2025).
27. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information [arXiv]-All Databases. Available online: <https://www.webofscience.com/wos/allldb/full-record/INSPEC:24758387> (accessed on 3 January 2025).
28. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475. [\[CrossRef\]](#)
29. 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.
30. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2016**, arXiv:1512.02325. [\[CrossRef\]](#)
31. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [\[CrossRef\]](#)
32. Peng, G.; Yang, Z.; Wang, S.; Zhou, Y. AMFLW-YOLO: A Lightweight Network for Remote Sensing Image Detection Based on Attention Mechanism and Multiscale Feature Fusion. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–16. [\[CrossRef\]](#)
33. Xue, C.; Xia, Y.; Wu, M.; Chen, Z.; Cheng, F.; Yun, L. EL-YOLO: An Efficient and Lightweight Low-Altitude Aerial Objects Detector for Onboard Applications. *Expert Syst. Appl.* **2024**, *256*, 124848. [\[CrossRef\]](#)
34. Wang, Z.; Hua, Z.; Wen, Y.; Zhang, S.; Xu, X.; Song, H. E-YOLO: Recognition of Estrus Cow Based on Improved YOLOv8n Model. *Expert Syst. Appl.* **2024**, *238*, 122212. [\[CrossRef\]](#)
35. Huang, M.; Mi, W.; Wang, Y. EDGS-YOLOv8: An Improved YOLOv8 Lightweight UAV Detection Model. *Drones* **2024**, *8*, 337. [\[CrossRef\]](#)
36. Wang, C.; Meng, L.; Gao, Q.; Wang, J.; Wang, T.; Liu, X.; Du, F.; Wang, L.; Wang, E. A Lightweight Uav Swarm Detection Method Integrated Attention Mechanism. *Drones* **2023**, *7*, 13. [\[CrossRef\]](#)
37. Bo, C.; Wei, Y.; Wang, X.; Shi, Z.; Xiao, Y. Vision-Based Anti-UAV Detection Based on YOLOv7-GS in Complex Backgrounds. *Drones* **2024**, *8*, 331. [\[CrossRef\]](#)
38. Cui, C.; Gao, T.; Wei, S.; Du, Y.; Guo, R.; Dong, S.; Lu, B.; Zhou, Y.; Lv, X.; Liu, Q.; et al. PP-LCNet: A Lightweight CPU Convolutional Neural Network. *arXiv* **2021**, arXiv:2109.15099. [\[CrossRef\]](#)
39. Demystify Mamba in Vision: A Linear Attention Perspective-All Databases. Available online: <https://webofscience.clarivate.cn/wos/allldb/full-record/INSPEC:25143401> (accessed on 3 January 2025).
40. Zhang, H.; Zhang, S. Shape-IoU: More Accurate Metric considering Bounding Box Shape and Scale. *arXiv* **2023**, arXiv:2312.17663.
41. Zhao, J.; Zhang, J.; Li, D.; Wang, D. Vision-Based Anti-UAV Detection and Tracking. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25323–25334. [\[CrossRef\]](#)
42. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [\[CrossRef\]](#)

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.