drones

MDPI

*Article*

# EDGS-YOLOv8: An Improved YOLOv8 Lightweight UAV Detection Model

**Min Huang [1,2], Wenkai Mi [2] and Yuming Wang [1,*]**

[1] Shijiazhuang Campus, Army Engineering University, Shijiazhuang 050003, China; huangmin@hebust.edu.cn
[2] Hebei University of Science and Technology, Shijiazhuang 050018, China; mwk@stu.hebust.edu.cn
* Correspondence: wangyuming@aeu.edu.cn

**Abstract:** In the rapidly developing drone industry, drone use has led to a series of safety hazards in both civil and military settings, making drone detection an increasingly important research field. It is difficult to overcome this challenge with traditional object detection solutions. Based on YOLOv8, we present a lightweight, real-time, and accurate anti-drone detection model (EDGS-YOLOv8). This is performed by improving the model structure, introducing ghost convolution in the neck to reduce the model size, adding efficient multi-scale attention (EMA), and improving the detection head using DCNv2 (deformable convolutional net v2). The proposed method is evaluated using two UAV image datasets, DUT Anti-UAV and Det-Fly, with a comparison to the YOLOv8 baseline model. The results demonstrate that on the DUT Anti-UAV dataset, EDGS-YOLOv8 achieves an AP value of 0.971, which is 3.1% higher than YOLOv8n's mAP, while maintaining a model size of only 4.23 MB. The research findings and methods outlined here are crucial for improving target detection accuracy and developing lightweight UAV models.

**Keywords:** drone detection; small target drones; YOLOv8; EMA; DCNv2; Ghostnet



**Citation:** Huang, M.; Mi, W.; Wang, Y. EDGS-YOLOv8: An Improved YOLOv8 Lightweight UAV Detection Model. *Drones* **2024**, *8*, 337. https://doi.org/10.3390/drones8070337

Academic Editors: Liuguo Yin and Shu Fu

Received: 24 June 2024
Revised: 17 July 2024
Accepted: 17 July 2024
Published: 20 July 2024



## 1. Introduction

Unmanned aerial vehicles (UAVs) are extensively utilized in multiple domains, including power inspections [1], emergency rescues [2], and data collection [3]. However, the broad adoption of UAVs has introduced certain issues [4], such as the potential for unauthorized UAVs to cause public safety incidents [5] and pose threats to aviation [6]. Given the potential dangers of UAVs to air traffic, public safety, personal privacy, etc., developing an efficient UAV identification model is very important. This helps prevent incidents such as drone disturbances related to the security of public incidents and military security. Advances in identifying drones are expected to promote the development of intelligent surveillance systems and exert greater application potential in the market, driving technological breakthroughs and standardization improvements in the entire drone industry chain. At present, the miniaturization, concealment, and diversification of UAV flight modes have brought a lot of challenges to their detection and recognition.

Currently, the primary methods for UAV detection comprise radar, photoelectric, radio frequency, and acoustic detection [7,8]. As "low, slow, and small" targets, UAVs often have weak electromagnetic reflections. Traditional radar detection systems exhibit low detection capabilities in intricate and congested environments, and improving detection performance is costly [9]. Radiofrequency (RF) detection is vulnerable to clutter interference, and acoustic detection is unsuitable for UAV detection due to its short detection distance and high cost. Compared with other methods, visual detection methods have received more and more attention for their low price, high accuracy, and high reliability.

Traditional computer vision methods struggle to the extract features of small UAV targets, making them difficult to detect and easily disrupted by environmental factors. This results in low detection accuracy and poor overall performance. Recently, the swift progress

of deep learning has resolved numerous challenges in computer vision, but challenges remain. At long distances, the image size of the UAV target is small, and there needs to be more semantic information, limiting the effectiveness of target recognition, especially in complex backgrounds. Enhancing UAV detection in complex scenarios while dealing with resource limitations such as computing power and storage space presents an enhanced EDGS-YOLOv8 network derived from YOLOv8. The network increases its focus on the image's region of interest (ROI) by improving the neck structure and incorporating an attention mechanism. It reduces the information loss in target detection. At the same time, the Detect module has been improved using deformable convolution (DCNv2) to enhance the model's capability for capturing local details. In addition, we implement ghost convolution and C3Ghost modules to compress the neck network and create a model that achieves high precision while requiring fewer computing resources.

This paper's key contributions can be summarized as follows:

1. By incorporating a dedicated small-target detection head, the algorithm enhances the recognition accuracy of UAVs by minimizing the loss of small-target feature information and emphasizing tiny targets. Additionally, eliminating the detection head in the network that is less sensitive to small-target UAVs contributes to a lighter model without compromising accuracy;
2. The C3 module replaces the C2f module in the network's neck, offering higher-level feature extraction capability and better multi-scale adaptability to capture richer semantic and contextual information. Additionally, the bottleneck module in the C3 module is replaced by GhostBottleneck, which reduces model parameters and mitigates information loss during long-distance feature transmission. Using ghost convolution instead of traditional convolution further lightens the UAV detection network;
3. To significantly preserve pixel-level attributes and spatial information on the feature map, we integrate the efficient multi-scale attention mechanism (EMA) into the model's neck. This mechanism, capable of parallel processing, facilitates cross-channel feature interactions. This enhancement enables EDGS-YOLOv8 to focus more accurately on crucial regions within the image, thus extracting more comprehensive, stable, and discriminative features for detection;
4. Introducing deformable convolution net v2 to the detection head improves the model's robustness in accurately recognizing and detecting UAV features through its local detail and geometric deformation-capturing capabilities. This initiative not only reduces the computational complexity but also results in improved detection accuracy.

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 presents and details the improved EDGS-YOLOv8 UAV detection model. Section 4 describes the experimental environment and parameter settings and performs ablation and comparison experiments on the open-source dataset DUT Anti-UAV. In addition, to evaluate the performance of EDGS-YOLOv8 on other datasets, comparative experiments are also conducted on the publicly available UAV dataset Det-Fly and the small target detection dataset VisDrone2019, and the experimental results are interpreted visually to verify the superiority of the improved model. Section 5 summarizes the results and looks at future research directions.

## 2. Related Work

The wide application of UAVs poses a great threat to air defense security, and their effective detection is necessary for the correct implementation of monitoring and warning. The detection of UAV targets is faced with complex detection scenarios, and its low flight altitude, small size, and strong maneuverability have always been hot issues in computer vision research.

Machine vision-based methods are increasingly being applied across various fields with the advancement of deep learning technology due to their low cost and simple configuration. Deep learning-based detection frameworks can be categorized into two

types: two-stage detectors and single-stage detectors. Two-stage detectors, like Faster R-CNN [10] and Mask R-CNN [11], use separate neural networks for region proposal generation and classification. While this approach achieves high accuracy, it results in large and time-consuming models suitable for portable deployment and real-time applications. On the other hand, single-stage detectors, such as the YOLO algorithm, have fewer model parameters, balance accuracy and speed well, capture target features effectively, and are widely used in UAV target detection.

Given the rapid progress in computer vision, extensive research has been conducted on UAV detection algorithms. Hu et al. [12] were the first to introduce the YOLOv3-based algorithm into UAV target detection. They used the last four scale feature maps to predict object bounding boxes and calculated the UAV size on these feature maps based on input data, adjusting the number of anchor frames accordingly. This method achieves higher detection accuracy and obtains a more accurate UAV bounding box. Zhai et al. [13] enhanced the detection of small targets by adding multi-scale prediction. Dadrass Javan, F et al. [14] improved the YOLOv4 network by altering the number of convolution layers, extracting more accurate and detailed semantic features, and achieving better performance on the same drone dataset compared to the baseline model. For the detection of small UAVs, Delleji et al. [15] proposed an instance enhancement strategy, added a shallow layer, performed hyperparameter tuning operations on YOLOv5, and established the corresponding UAV data set to verify the improved algorithm's effectiveness. Zhu et al. [16] proposed an enhanced YOLOv5 UAV aerial image detection model by adding a detection head, incorporating a transformer detection head (TPH) instead of the original, and integrating a convolutional block attention model (CBAM) to identify attention areas in dense scenes. Zhao et al. [17] enhanced the YOLOv5 model for rapid and precise detection of small targets in intricate settings by integrating a transformer encoder module, global attention mechanism, and coordinate attention mechanism into the C3 module, and robustness and generalization performance were validated on the custom SUAV-DATA dataset. Ma et al. [18] present an efficient LA-YOLO network based on YOLOv5 for UAV detection in various environments and conditions, especially low-altitude backgrounds. The network integrates the SimAM attention mechanism and introduces the normalized Wasserstein distance fusion block, enhancing the model's checking accuracy.

To deploy the model on edge computing devices and improve its accuracy, Zhang et al. [19] utilized the K-means algorithm to optimize YOLOv3's anchor frame and then compressed the model using pruning techniques. This approach enhanced UAV detection accuracy and enabled real-time detection. Andrew et al. [20] investigated utilizing the lighter MobileNet as the backbone network to create lightweight models, balancing speed and accuracy. Sun et al. [21] presented TIBNet, a network for UAV detection employing a compact iterative backbone to reduce computational overhead and compress the network, albeit without achieving real-time performance. Dai et al. [22] introduced CrossConv in the C3 module to address feature similarity loss during fusion, enhancing feature representation and improving mAP by 7.5%. Wang et al. [23] implemented deformable convolution, resulting in a 2.3% increase in detection accuracy over the original model. Wang et al. [24] developed a YOLOX-based lightweight UAV swarm detection method that simplifies the network structure through deeply divisible convolution. SE and CBAM modules are introduced to enhance the ability to extract and focus on important features, using DIoU instead of IoU to calculate regression losses, and combining data enhancement technology to expand the dataset, making UAV swarm detection suitable for edge computing devices. Bai et al. [25] proposed a lightweight and effective T-YOLO model based on the YOLOv5 baseline for detecting vegetative buds in tea gardens. The model incorporates lightweight modules and a high-performance feature extraction network and introduces a dynamic detection head to mitigate feature loss due to its lightweight, decreasing model parameters and complexity while enhancing detection capability. Zhou et al. [26] proposed the VDTNet UAV detection network. It integrates an SPP module to enhance detection performance, compresses and reduces the model size to increase speed, introduces SPPS and ResNeck

modules, and adds a backbone attention module to compensate for precision loss. VDTNet achieves low cost and fast real-time detection. When improving the synchronous detection model of grape picking points based on YOLOv8, Chen et al. [27] decreased the model parameters by adopting innovations like multi-scale attention mechanisms, BiFPN structures, and PConv. Li et al. [28] developed an enhanced YOLOv8s UAV aerial image detection model by incorporating the Bi-PAN-FPN concept, optimizing feature fusion and model parameters, and employing the GhostblockV2 structure to minimize model parameters and WiseIoU loss to enhance detection performance.

In current research, many researchers prioritize either high accuracy or small model size, neglecting the importance of maintaining a high frame rate (FPS) while balancing both model size and accuracy. Hence, this study aims to develop a model that balances high accuracy, compact size, and FPS performance.

## 3. Materials and Methods

### *3.1. YOLOv8*

YOLOv8 is structured into three components: the backbone, neck, and YOLO head, as illustrated in Figure 1. YOLOv8 [29] enhances the YOLOv5 [30] architecture by incorporating CSPDarkNet-53 as the backbone for feature extraction. It generates three efficient feature layers from the input image by substituting the C3 module with the C2f module (CSPLayer_2Conv). This architecture enhances feature extraction by merging the three effective feature layers of the backbone network using a PAN-FPN mechanism similar to YOLOv5. The YOLO decoupled head is a classifier and regressor, determining object associations with the feature points.

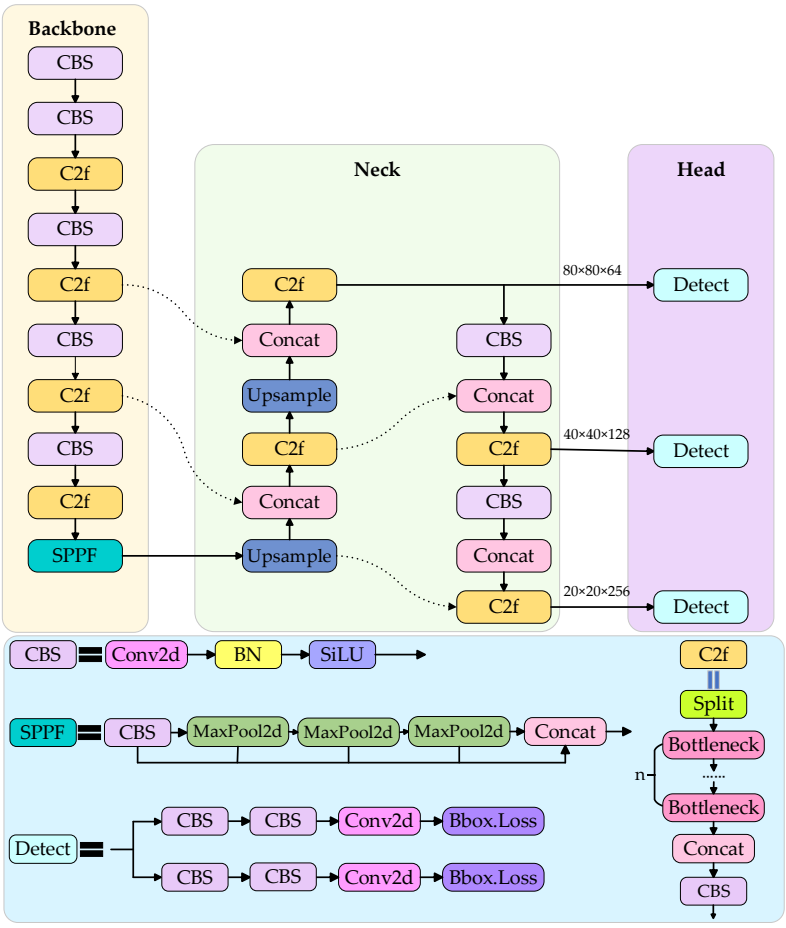



**Figure 1.** YOLOv8 block diagram, with backbone, neck, and head units.

The YOLOv8 series comprises five variants: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, tailored for various application needs. Although they share the same architecture, their differences in width and depth yield varying performance and speed. Compact models like YOLOv8n are resource-efficient, run swiftly, and are suitable for mobile deployments and high FPS scenarios. Conversely, larger models like YOLOv8x offer enhanced performance at the cost of complexity and increased computational demands. To address high FPS requirements and ease deployment, we enhanced the smallest model, YOLOv8n, to minimize resource usage while maintaining real-time detection capabilities.

In this paper, we make several improvements to YOLOv8 to improve target detection accuracy and reduce the model size. By improving the neck structure, the model's size is reduced while improving its accuracy. In the network neck, the C3Ghost module replaces the C2f module, and ghost convolution replaces traditional convolution, which reduces the computational overhead of the model and makes the model more compact and efficient. The integrated EMA mechanism is integrated at the neck of the network to enhance the stability, generalization ability, and robustness of the model. A new DDetect detection head is designed by introducing DCNv2 into the Detect module, improving detection accuracy and robustness.

In summary, these improvements enable our improved YOLOv8 model to perform well in small-target UAV detection missions with higher accuracy and efficiency while maintaining good stability and robustness.

### *3.2. Improvement of the Neck*

In the YOLOv8 training process, feature inputs are categorized into five types of scale features by the backbone (B1–B5), FPN (P3–P5) [31], and PAN [32] (N3–N5). Traditional FPNs typically use a single top-down approach to convey semantic features deeply. By fusing B3–P3 and B4–P4, PAN–FPN can enhance the semantics of the feature pyramid, but it may lead to a certain degree of localization information loss. PAN–FPN complements the bottom-up part of the FPN structure and enhances the learning of localization features by fusing P4–N4 and P5–N5, playing a complementary role. However, given that UAVs typically appear to be small in size in images, and some even have a width and height of only a few pixels, we find room for improvement when applying existing structures to small target detection. Large-scale feature maps receiving increased attention might lead to overlooking useful features in the detection model, potentially diminishing detection quality. Despite efforts to fuse and complement B, P, and N features, there may be a higher reuse rate of features. Moreover, extended up-and-down sampling paths could result in information loss in the original features. In addition, after a long up-and-down sampling path, the original features may lose some information. The improved neck structure is illustrated in Figure 2.

To improve the detection against the drone dataset, we made the following adjustments:

First, to enhance the YOLOv8 network's applicability for object detection, we skip over the feature output from the first convolutional layer due to their high resolution, which is often unnecessary for general object detection. For better small target detection, where high-resolution feature maps are vital for spatial information, we introduce an up-sampling process in the FPN. This process connects the high-resolution feature maps from the backbone B2 layer to the P2 layer, increasing the total sampling layers from three to four. This adjustment aims to minimize spatial information loss. In the original YOLOv8 model, detecting targets of various sizes typically involves using feature maps of different scales. Larger targets utilized feature maps obtained by 32-fold downsampling, while smaller targets used smaller downsampling. We added a 4-fold downsampling process to the feature extraction pyramid to focus more on small UAV targets, generating an N2 layer. This modification, along with the feature fusion network, reduces the sensory field of the feature map and enriches target information, leading to improved learning of target features through multi-scale fusion.

Our second adjustment was to optimize the model further to adapt to UAV detection targets with a few large objects. We removed the 32-fold down-sampling layer N5. This change, based on our experiments, directed the model's focus toward detecting smaller UAV targets

and significantly reduced its overall size. This adjustment not only improved the model's performance but also made it more efficient and practical for real-world applications.

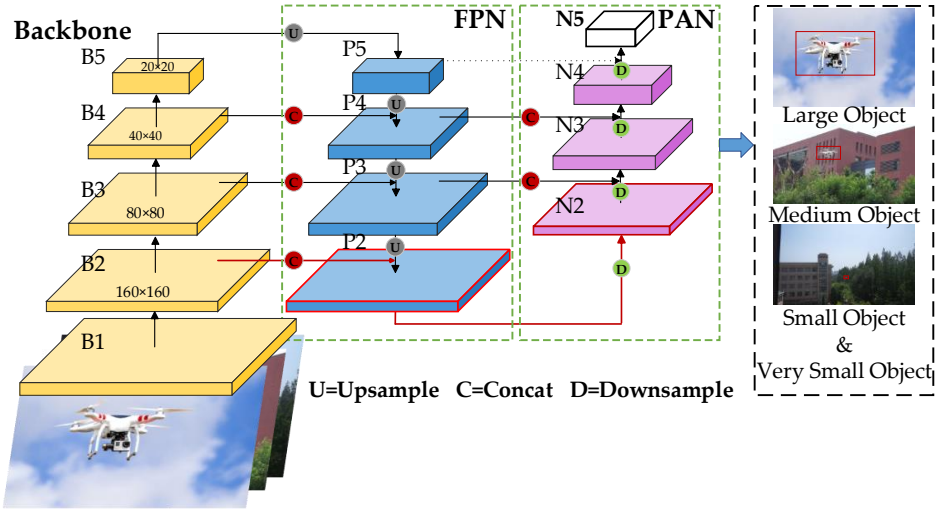



**Figure 2.** Improvement scheme at the neck.

### *3.3. C3Ghost*

Compared with the C2f module, the C3 module has a higher feature extraction ability and better multi-scale adaptability. It can capture richer semantic and contextual information and enhance target detection precision and accuracy. To compress model size and reduce model deployment costs. We replaced the C2f module with C3, while replacing the bottleneck in the C3 module with GhostBottleneck and the conventional convolution in the neck network with ghost convolution [33]. The structures of C3Ghost and GhostBottleneck are shown in Figure 3A,B.

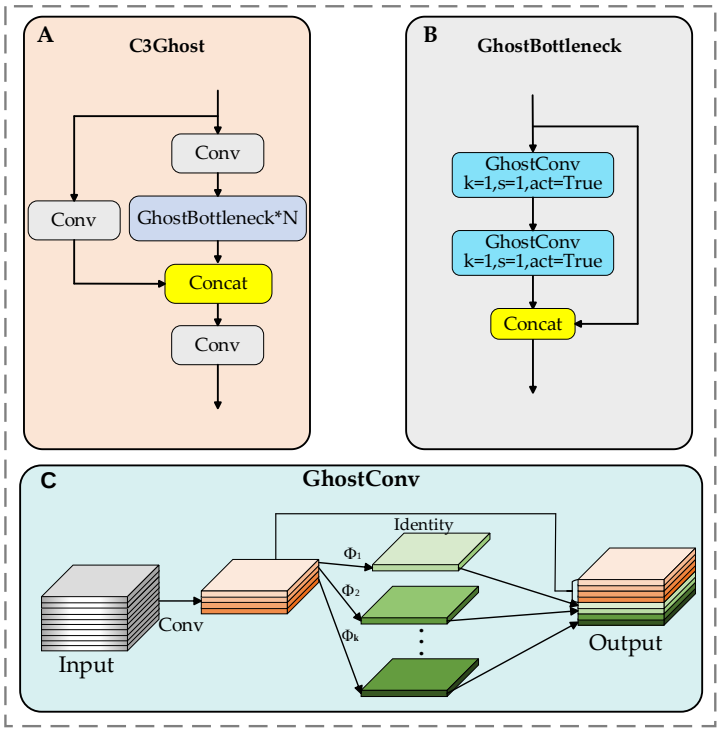



**Figure 3.** Diagram of C3Ghost in EDGS-YOLOv8. (**A**) C3Ghost module. (**B**) GhostBottleneck module. (**C**) GhostConv module.

The GhostBottleneck architecture, based on ghost convolution, is an efficient network design. As depicted in Figure 3C, ghost convolution achieves a feature map similar to standard convolution but with reduced parameters, simultaneously optimizing computational efficiency and model accuracy. It involves an initial convolution with minimal computation and an inexpensive linear transformation to generate the remaining feature maps. These generated maps are combined with the channel size to produce the final output, enhancing model efficiency without compromising feature representation capabilities.

### *3.4. Efficient Multi-Scale Attention Module*

Ensuring UAV identification accuracy also requires detecting precise spatial information, which is crucial for the UAV detection model's accurate operation of equipment. As the distance from the target drone increases, its size gradually decreases. This phenomenon can cause the contrast of the target drone in the image to decrease, making it more difficult to mark and identify the drone. We incorporate the efficient multi-scale attention (EMA) [34] mechanism to enhance the model's capability to capture small target features. EMA, the coordinate attention (CA) variant, utilizes a concurrent framework for data handling. This approach speeds up the model's training on extensive datasets and processes features at various scales to boost accuracy, as illustrated in Figure 4.

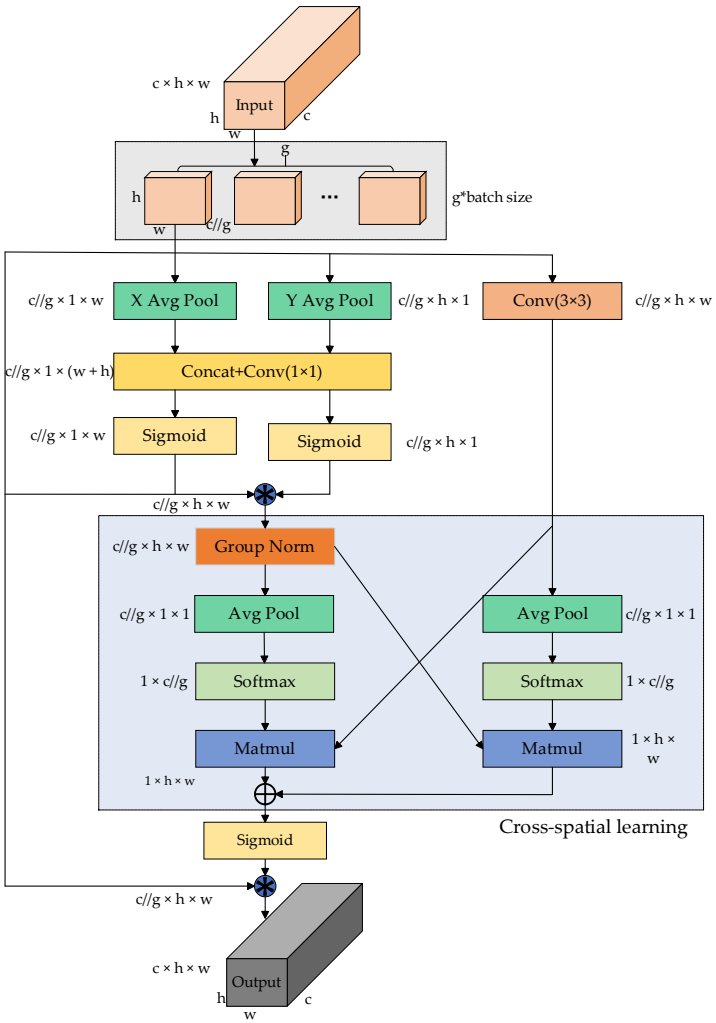



**Figure 4.** Structure of EMA. Here, "g" denotes grouping, "X Avg Pool" represents 1D horizontal global pooling, and "Y Avg Pool" represents 1D vertical global pooling. Different colors in the figure indicate various functional modules.

EMA initially partitions the input $X \in R^{C\times H\times W}$ into G groups across various dimensions to capture diverse semantic information, where C is the number of channels, H is the height, and W is the width of the input tensor. This grouping is defined as $X = [X_0, X_1, \ldots, X_{G-1}]$, $X_i \in R^{C//G\times H\times W}$. EMA then uses three parallel paths to capture multi-scale spatial details: in the $1 \times 1$ branch, data are encoded in both spatial directions and then undergo 1D global average pooling across the image's horizontal and vertical directions. The output of the two $1 \times 1$ branch pools is adjusted to a 2D bifurcated distribution using two sigmoid functions after concat and conv. To encourage varied cross-channel interaction, within each group, two-channel attention maps are multiplied together. In the $3 \times 3$ branch, EMA utilizes a $3 \times 3$ convolution kernel to expand multiscale features. Consequently, EMA processes inter-channel information across multiple channels, maintaining precise spatial structural details and adjusting the importance of different channels.

Additionally, the EMA mechanism utilizes a cross-spatial information aggregation strategy to coordinate the output features between different channels. To preserve the global spatial information of the $1 \times 1$ branch, 2D global pooling is applied to encode and then transform to the corresponding dimension. Meanwhile, the adjusted $3 \times 3$ branch features are incorporated into the joint activation mechanism. The results of these parallel processing stages are combined through matrix dot products to create a spatial attention map. This method allows spatial information at different scales to be effectively collected at the same processing stage. Similarly, an additional spatial attention map is produced in the $3 \times 3$ branch, preserving accurate spatial location information. Subsequently, the output feature mapping combines the two spatial attention weights via a sigmoid function to produce an output feature map for each group.

EMA is a parallel processing mechanism with low model complexity, relatively small computational effort, and higher computational efficiency. Cross-space learning facilitates the model's capture of feature dependencies at different scales and improves its overall performance. However, it is important to acknowledge that using EMA involves exponentially weighted averaging of historical attention weights, which can increase computational complexity, particularly with lengthy sequences and large models. Additionally, the attenuation factor must be continuously adjusted during application, complicating the model tuning process.

### *3.5. DDetect*

Traditional neural networks typically use fixed-size convolution kernels (e.g., $3 \times 3, 5 \times 5$), making adjusting to varying target shapes challenging [35]. DAI et al. [36] introduced a deformable convolutional network to address this limitation. Deformable convolution networks offer adaptability to geometric changes, enhancing recognition capability [37]. Increasing trainable offset via deformable convolution networks to accommodate object shape variations has improved object detection robustness [38].

DCNv1 (deformable convolutional networks v1) incorporates deformable convolution to address limitations in traditional image enhancement and affine transformations [39]. In DCNv1, an offset variable is assigned to each sampling point in a convolution kernel (e.g., a $3 \times 3$ kernel with nine sampling points). This offset enables random sampling near the current position, allowing the convolution kernel's size and position to adjust dynamically according to the target object. This flexibility enhances the network model's performance in detecting irregularly shaped and sized objects.

The output at sampling point $t_0$ in a conventional 2D convolution is defined as follows:

$$y(t_0) = \sum_{t_n \in R} x(t_0 + t_n) \cdot w(t_n) \tag{1}$$

In Equation (1), $R = \{(-1,1), (-1,0), (-1,-1), (0,-1), (0,0), (1,-1), (1,0), (1,1), (0,1)\}$ represents the receptive fields. $t_n$ and $t_0 + t_n$ are the elements of R, representing the sampling position in the receptive field. At the corresponding location, $w(t_n)$ represents the weight of the convolution kernel, and $x(t_0 + t_n)$ is the input x.

By incorporating the offset $\{\Delta t_n | n = 1, 2, \ldots, N\}, N = |R|$ into R, we derive the expression for DCNv1 as follows:

$$y(t_0) = \sum_{t_n \in R} x(t_0 + t_n + \Delta t_n) \cdot w(t_n) \quad (2)$$

Sampling is now performed at the irregular offset position $t_n + \Delta t_n$. Therefore, the pixel $x(t_0 + t_n + \Delta t_n)$ after the offset is introduced is usually realized by the bilinear interpolation method. The formula for bilinear interpolation is as follows:

$$x(t) = \sum_{v} x(v) \cdot \Phi(v, t) \quad (3)$$

where $v$ represents every integer spatial location within the input feature map, specifically the four integer points surrounding any position in the region, and $t = t_0 + t_n + \Delta t_n$ indicates the value at a fixed position in the feature map. $\Phi(v, t)$ represents the bilinear interpolation kernel function, a 2D kernel separable into two 1D kernels, as shown in Formula (5):

$$\Phi(v, t) = \varphi(v_x, t_x) \cdot \varphi(v_y, t_y) \quad (4)$$

And, $\Phi(v, t) = \max\{0, 1 - |v - t|\}$.

As shown in Figure 5A, a convolution kernel identical in parameters to the existing convolution layer is applied to the current input feature map to determine the offset (e.g., the kernel in Figure 5A is also $3 \times 3$ with an expansion degree of 1). During forward propagation, the convolution operation and the generation of offsets for the output features are performed simultaneously. The output displacement field retains the same spatial resolution as the input feature map, with 2N channels corresponding to N 2D offsets, where N represents the number of convolution kernel pixels. The gradient is backpropagated through bilinear operations as described in Formulas (3) and (4) to learn the offset.

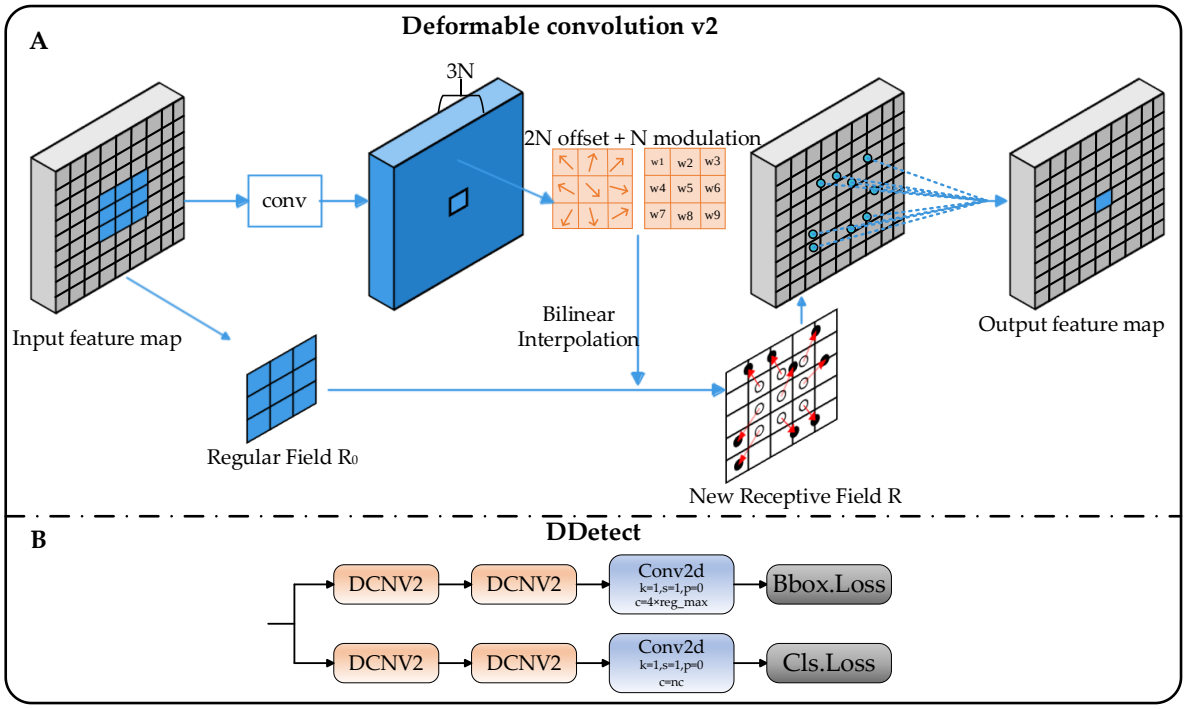



**Figure 5.** Diagram of DDetect in EDGS-YOLOv8. (**A**) DDetect module. (**B**) Schematic diagram of deformable convolution v2.

Visualization of the results of the experiments using DCNv1 showed that the corresponding positions of the receptive range extend beyond the object range, leading to

features not being influenced by the image's essence. Ideally, all corresponding locations should be within the target range. To further enhance the deformable convolution network's spatial manipulation ability, a modulation mechanism was introduced [40], resulting in DCNv2 (deformable convolution networks v2) [41]. This version extends the deformable convolution and integrates it more comprehensively within the network. The area covered by the deformable convolution is larger than the target area, enhancing the modeling capability and increasing the receptive field. The convolution incorporating modulation can be expressed as follows:

$$y(t_0) = \sum_{t_n \in R} x(t_0 + t_n + \Delta t_n) \cdot w(t_n) \cdot \Delta m_n \quad (5)$$

The above equation incorporates the weight coefficient $\Delta m_n (0 \leq \Delta m_n \leq 1)$. The modulation mechanism learns the weights at each sampling point to adjust the amplitude of input features at various locations, suppressing irrelevant background information and minimizing the interference of external factors. Figure 5A depicts the structure of the DCNv2 module.

Applying the DCNv2 module to the detection head involves replacing a portion of the CBS module to form the DDetect detection head. This modification provides significant advantages for UAV target detection, improving the model's flexibility in adapting to UAV positions in the air, including attitudes, angles, and distances, as demonstrated in Figure 5B. At the same time, the local detail capture capability of the DCNv2 module helps to identify the features of the UAV more finely, such as wings, propellers, etc., further improving the detection accuracy. This optimization can also reduce the computational complexity, which enhances the real-time efficiency of target detection and is suitable for UAV monitoring, tracking, and application fields.

### *3.6. EDGS-YOLOv8*

To enhance UAV target detection accuracy, achieve a lightweight model, and reduce deployment costs, we introduce an efficient detection model, EDGS-YOLOv8, built upon YOLOv8n. Figure 6 depicts the network's overall structure.

Compared with YOLOv8, we have performed the following optimizations: (1) As described in Section 3.2, the neck structure was improved. The first C2F layer's high-resolution feature map was connected to the neck with a down-sampled 4x output layer as a new head to focus on small objects in the model. The 32-fold downsampled output layer in the neck was removed to optimize the model further to accommodate drone detection targets and lightweight requirements. (2) In Section 3.3, the C2f module in the neck was substituted with the C3Ghost module. The C2f module in the neck network was replaced with the C3 module for efficient feature extraction and comprehensive spatial context processing. The Ghost Bottleneck structure was adopted to optimize the bottleneck and replace traditional convolution with ghost convolution. This enhances model performance, lowers computational overhead, and makes the model more compact and efficient. (3) As outlined in Section 3.4, the EMA mechanism is introduced at the neck of the network. The EMA mechanism is integrated into the neck section to enhance the system's stability, generalization, and robustness, thereby increasing the accuracy and consistency of UAV target identification. (4) As detailed in Section 3.5, a new DDetect detection header was developed. We employ the DCNv2 module to design the DDetect detection head, enabling the detection model to more effectively adjust to the variations in UAVs' positions, angles, and distances in the air, thereby enhancing the precision and robustness of detection. The characteristics of drones are identified to more accurately improve detection accuracy. It decreases computational demands and enhances the model's verification accuracy.

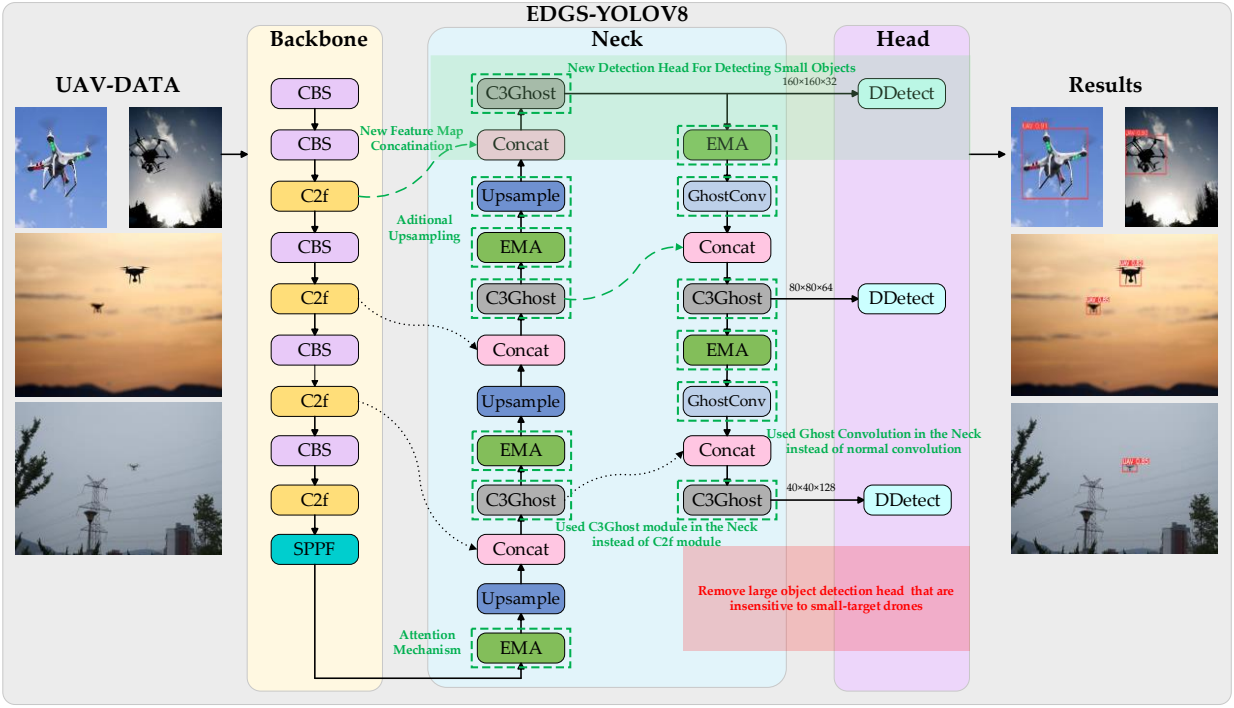



**Figure 6.** Frame diagram of EDGS-YOLOv8. A new high-resolution feature map and associated layers are incorporated into the neck, with their output serving as a new head for detecting small objects (layers added and removed are shown in green and red, respectively).

## 4. Results

### *4.1. Dataset and Its Preprocessing*

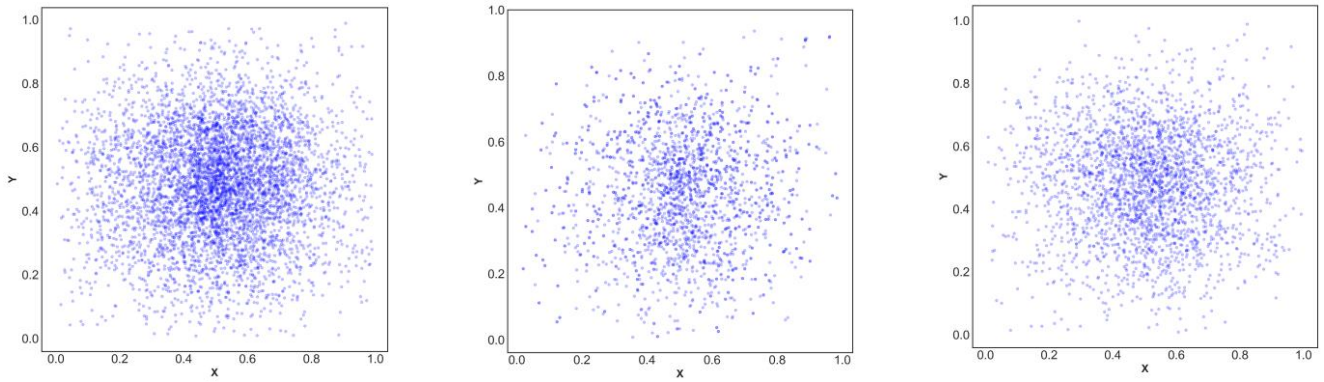
This paper selects the DUT Anti-UAV [42] anti-UAV dataset as the object of experimental verification. In computer vision tasks, selecting datasets is crucial to obtaining a robust model. Therefore, datasets for UAV detection and tracking are constantly being proposed. The following are descriptions of several relatively complete existing drone datasets:

(1) MAV-VID [43]: The dataset exhibits concentrated drone locations, mainly with horizontal differences. The detected objects are very small, averaging only 0.66% of the entire image size.

(2) Purdue [44]: This dataset contains 50 videos captured by cameras fixed to aerial drones at high flight speeds on three target drones. The unmanned aerial vehicle (UAV) and environment in this data set are single, which is not suitable for UAV detection tasks but better suited for small target UAV tracking issues.

(3) Anti-UAV [45]: This dataset includes 318 fully labeled videos with visible and infrared dual-mode information. The anti-UAV motion range is wide, although most instances are concentrated in the central area. The dataset aims to improve vision-based detector performance during nighttime operations.

Compared with the previously mentioned datasets, the datasets selected in this study present a more dispersed distribution of drones, showing relatively uniform characteristics both horizontally and vertically, as shown in Figure 7. This figure describes the position distribution of the objects relative to the center location using scatter plots: (a) the position distribution of the training set, (b) the position distribution of the test set, and (c) the position distribution of the validation set. The diversity of drone types, scenes, lighting conditions, and weather changes in the selected dataset is not just a random assortment but a deliberate choice that greatly enhances the robustness of the model, underscoring the importance of dataset diversity in our research.




(**a**) (**b**) (**c**)

**Figure 7.** Position distribution of the DUT Anti-UAV dataset. (**a**) Plot of training set location distribution. (**b**) Test set location distribution graph. (**c**) Validation set location distribution graph.

The dataset contains 10,000 images, including 5200 training sets, 2200 test sets, and 2600 verification sets. There are 10,109 detection objects, of which there are 5243, 2245, and 2621 objects in the training, test, and verification sets, respectively, and the proportion of small objects is relatively large. This dataset contains more than 35 types of drones, reflecting their extremely high diversity. In the selected dataset, most drones fly outdoors and operate in variable weather and lighting conditions. This is clearly shown in Figure 8.

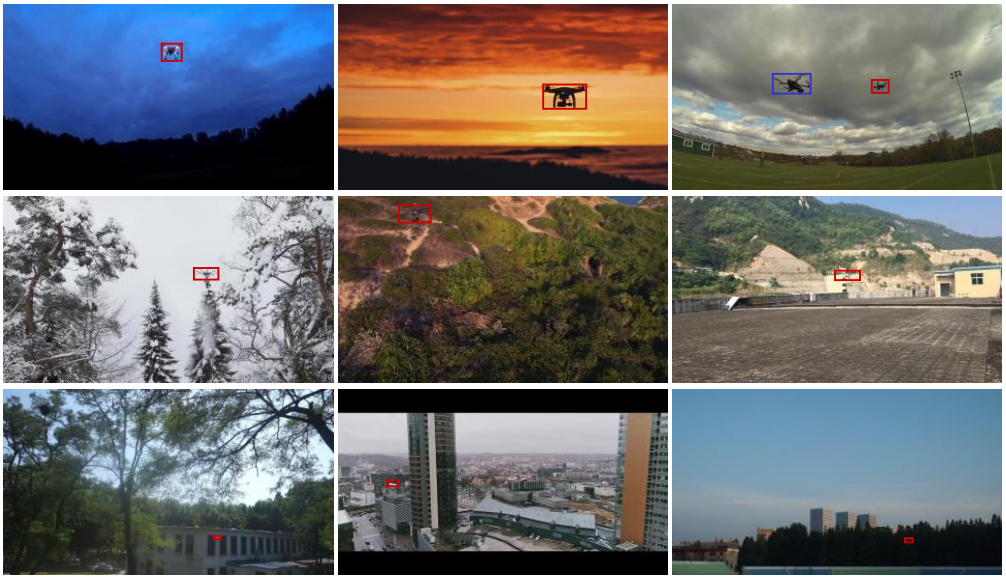


**Figure 8.** Examples of the detection images and annotations in our dataset. The objects enclosed by the box represent the drone detection targets in our dataset annotations.

In this paper, the method of pre-partitioning datasets is used. Given the prevalence of small targets in our sample images, we normalize the sample size to 640 × 640 to balance real-time processing and accuracy requirements. This size allows the model to be deployed effectively on edge devices while retaining essential image information. Our hardware setup features an Intel Core i9-12900H CPU with 14 cores and 20 threads, running at 3.19 GHz, 32 GB of RAM, and a GeForce RTX 3070Ti laptop GPU with 8 GB of VRAM. We use PyTorch 1.9.1 and Torchvision 0.10.1 for the deep learning framework, along with Ultralytics YOLOv8 version 8.0.25, trained using official pretraining weights. Table 1 details the main parameter settings used in the training process, incorporating data augmentation techniques like image translation, scaling, left-right flipping, and mosaic, along with their respective probabilities of occurrence.

**Table 1.** Training parameter setting table.

| Parameters | Setup |
|---|---|
| Epochs | 800 |
| entry 2 Batch Size | 4 |
| Optimizer | SGD |
| NMS IoU | 0.7 |
| Initial Learning Rate | $1 \times 10^{-2}$ |
| Final Learning Rate | $1 \times 10^{-4}$ |
| Momentum | 0.937 |
| Weight-Decay | $5 \times 10^{-4}$ |
| Image Translation | 0.1 |
| Image Scale | 0.5 |
| Image Flip Left-Right | 0.5 |
| Mosaic | 1.0 |
| Close Mosaic | Last 10 epochs |

### *4.2. Ablation Experiment*

To validate the effectiveness of our proposed improvement, we presented the experimental process in this section using ablation experiments. According to different evaluation criteria (precision, recall, mAP50, mAP95, GFLOPS, model size, and FPS), the evaluation results are shown in Table 2 and Figure 9. The impact of different approaches is outlined in the table: benchmark model (A), designed neck network (P), introduction of the C3Ghost lightweight module and ghost convolution (G), introduction of the EMA attention mechanism (E), replacement of the original detect detection head (D) with the DDetect module. The table sequentially defines the baseline model A, the improved models A+P, A+P+G, A+P+G+D, and A+P+G+E+D, and quantitatively discusses the changes in five evaluation metrics across these models.

**Table 2.** Experimental results of various indices of ablation experiments.

| Model | Precision | Recall | mAP50 | mAP95 | GFLOPS | Model Size | FPS (Tasks/s) |
|---|---|---|---|---|---|---|---|
| A | 0.959 | 0.896 | 0.94 | 0.668 | 8.1 | 5.96 MB | 96.2 |
| A+P | 0.977 | 0.923 | 0.962 | 0.696 | 11.9 | 4.29 MB | 91.7 |
| A+P+G | 0.972 | 0.923 | 0.962 | 0.697 | 10.1 | 3.83 MB | 90.1 |
| A+P+G+D | 0.976 | 0.93 | 0.966 | 0.699 | 7.7 | 4.22 MB | 69.9 |
| A+P+G+E+D | 0.97 | 0.935 | 0.971 | 0.702 | 7.9 | 4.23 MB | 56.2 |

The baseline model has the lowest accuracy index but a high FPS of 96.2, indicating that the refined model, despite reducing parameters, adds some inference time. The refined model achieves an FPS of 56.2, ensuring real-time deployment requirements are met.

According to the experimental results, modifying the model structure is the most crucial method to enhance UAV detection accuracy (see Section 3.2). These modifications resulted in a 2.2% improvement in mAP50 and a 2.8% improvement in mAP95, emphasizing the significance of a more detailed feature image output layer and a compact target detection layer. Removing the large target detection header also reduced the model's size by 28%, decreasing its weight.

By replacing the original C2f module and conventional convolution with the C3Ghost module and ghost convolution (refer to Section 3.3) in the neck network, we achieved a 15.12% reduction in GFLOPS to 10.1 while preserving accuracy and FPS. The model size was reduced to 3.83 MB compared to the A+P model. The experimental findings indicate that decreasing the number of parameters without compromising accuracy can lead to a smaller model size and lower computational complexity, thereby enhancing the model.

By substituting the Detect module of the baseline model with the custom-designed DDetect module (refer to Section 3.5), compared with the A+P+G model, mAP50 has increased by 0.4%, and the GFLOPS has reached 7.7. This suggests that the designed DDetect module boosts accuracy and significantly reduces the model complexity.

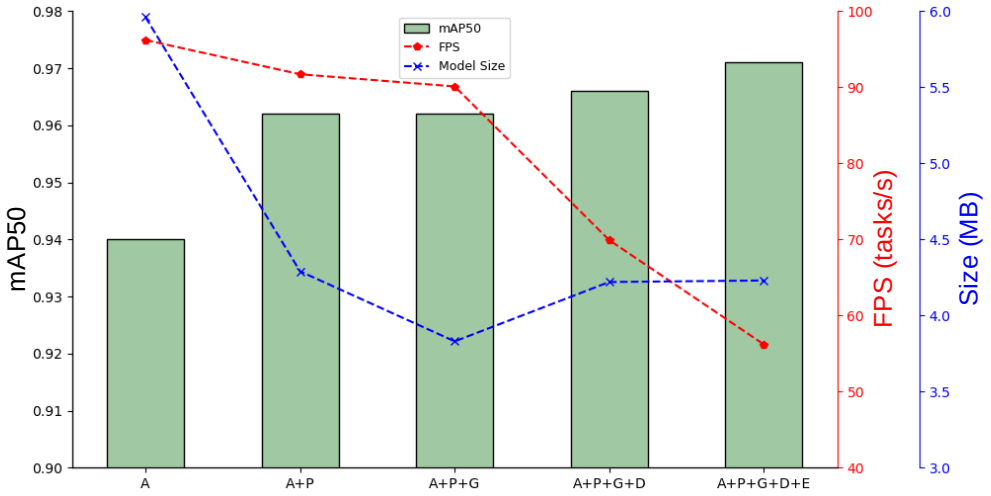



**Figure 9.** Comparison of mAP50, FPS, and model size before and after the improved structure.

We improved the model's accuracy by introducing the EMA based on the A+P+G+D model (see Section 3.4). This allows mAP50 and mAP95 to achieve high precision standards of 97.1% and 70.2%, respectively. This shows that adding an efficient multi-scale attention mechanism (EMA) to the network effectively utilizes multi-scale information, allowing the model to handle multi-scale input data better and positively impact bounding box regression. It is worth mentioning that these improvements have further improved the model's accuracy while ensuring the FPS of the model.

### *4.3. Performance Comparison Experiment with the Deep Learning Model*

In target detection, deep learning techniques are categorized into one-stage and two-stage methods based on their anchor generation mechanisms. Using experimental data from the original dataset, they implemented Faster-RCNN, Cascade-RCNN [46], and ATSS [47] for the two-stage approach and YOLOX [48] and SSD [49] for the one-stage approach. Given that real-time processing of UAV images better aligns with practical requirements, we also chose one-stage target detection methods with low hardware dependence and high precision, such as YOLOv5n and YOLOv8s. To illustrate the superiority of the model presented in this paper, we included networks published by other researchers in this dataset, such as LA-YOLO and VDTNet.

Table 3 presents the results of comparing the entire test dataset with the leading UAV target detectors currently available. The first three lines represent the backbone network that replaced these detectors with ResNet18, ResNet50, and VGG16 as the classic backbone network, and a total of nine different versions of the detection method were obtained. Rows four to seven represent the one-stage target detection approach, while rows eight to nine show results from other researchers on the DUT Anti-UAV dataset, including LA-YOLO and VDTNet. The last line is our proposed method, which is EDGS-YOLOv8.

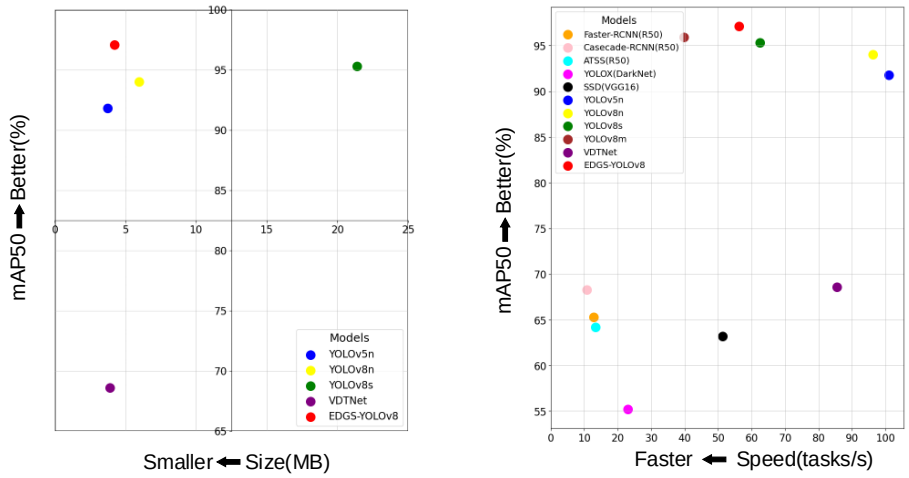
In this study, we also selected YOLOv8s and YOLOv8m, which have an expanded model architecture and additional parameters to enhance detection effectiveness relative to YOLOv8n (baseline model). As shown in Table 3, the accuracy of YOLOv8m reached 95.9%, surpassing the baseline model YOLOv8n. However, the FPS is relatively low due to the high number of model parameters. In contrast, EDGS-YOLOv8, our proposed method, has higher accuracy, a smaller model size, and faster detection speed, demonstrating its superiority in UAV target detection. The test equipment we use is the RTX 3070Ti laptop, which performs similarly to the RTX 2080 Super in the original text and is used to test the delay. On the DUT Anti-UAV dataset, EDGS-YOLOv8 still achieves higher accuracy even compared with Cascade (ResNet50), the highest accuracy model tested in the original text. In addition, EDGS-YOLOv8 is 2.5 tasks faster than the fastest YOLOX in the original text. On the other hand, VDTNet performed well on FPS, reaching 85.5/f.S-1, but its mAP was low, indicating frequent misdetections and omissions. YOLOv5n has the traits of high FPS and a minimal model parameter count, but the mAP does not meet the high precision standard.

**Table 3.** Comparison of experimental results.

| Model | Backbone | mAP50 | Model Size | FPS(Tasks/s) |
|---|---|---|---|---|
| Faster-RCNN [27] | ResNet50 | 0.653 | - | 12.8 |
| | ResNet18 | 0.605 | - | 19.4 |
| | VGG16 | 0.633 | - | 9.3 |
| Casecade-RCNN [46] | ResNet50 | 0.683 | - | 10.7 |
| | ResNet18 | 0.652 | - | 14.7 |
| | VGG16 | 0.667 | - | 8.0 |
| ATSS [47] | ResNet50 | 0.642 | - | 13.3 |
| | ResNet18 | 0.61 | - | 20.5 |
| | VGG16 | 0.641 | - | 9.5 |
| YOLOX [48] | ResNet50 | 0.427 | - | 21.7 |
| | ResNet18 | 0.400 | - | 53.7 |
| | DarkNet | 0.552 | - | 23.0 |
| SSD [49] | VGG16 | 0.632 | - | 51.3 |
| YOLOv5n [30] | | 0.918 | **3.73 MB** | **101** |
| YOLOv8n [29] | | 0.94 | 5.96 MB | 96.2 |
| YOLOv8s [29] | | 0.953 | 21.4 MB | 62.5 |
| YOLOv8m [29] | | 0.959 | 197 MB | 39.8 |
| LA-YOLO [18] | | 0.929 | - | - |
| VDTNet [26] | | 0.686 | 3.9 M | 85.5 |
| EDGS-YOLOv8 (Ours) | | **0.971** | 4.23 MB | 56.2 |

[1] In the original paper [42], the latency marked cyan was tested using the RTX 2080 Super GPU, and the latency of EDGS-YOLOv8 was tested using the RTX 3070Ti laptop GPU. Its performance is less different from that of the RTX2080 Super GPU. [2] "-" indicates that there was no official indicator report in the original paper; the original paper [26] did not report model size values, and the original paper [18] did not report model size values and FPS sizes. [3] The top results in each evaluation metric column are highlighted in bold.

As shown in Figure 10a, the EDGS-YOLOv8 results maintain the highest accuracy while having a small model size. Figure 10b demonstrates that the EDGS-YOLOv8 results have a significant advantage in balancing FPS and model accuracy. These findings have practical implications, confirming the benefits of EDGS-YOLOv8 in identifying small UAV targets in intricate scenes with high precision and real-time FPS performance, thereby enhancing the effectiveness of UAV detection in real-world scenarios.




(**a**) (**b**)

**Figure 10.** Contrast of the experimental results. (**a**) Model size versus accuracy on DUT Anti-UAV. (**b**) Inference speed versus accuracy on DUT Anti-UAV.

### *4.4. Experimental Results from the Public Dataset Det-Fly*

To assess the performance of EDGS-YOLOv8 on additional publicly accessible drone datasets, we selected the Det-Fly dataset [50]. This dataset includes over 13,000 images of flying target drones captured by another drone. Although the Det-Fly dataset contains only one type of drone, it overcomes the limitations of the drone dataset from a single perspective. The dataset directly captures a variety of attitudes of aerial target UAVs, including elevation, pitch, and below the horizon. This presents challenges for target detection regarding UAV attitude angle changes, etc., so we use this dataset to evaluate the model's adaptability under such scene changes. During this study, we assessed the model's performance following data partitioning rules similar to those in [50]. The detailed experimental results can be found in Table 4.

**Table 4.** Comparative experiments with the Det-Fly dataset.

| Model | Precision | Recall | F1 | mAP50 | mAP95 |
|---|---|---|---|---|---|
| YOLOv8n | 0.921 | 0.872 | 0.90 | 0.911 | 0.6 |
| Ours | 0.914 | 0.919 | 0.92 | 0.934 | 0.621 |

Based on the comparison experiment results, the recall value of EDGS-YOLOv8 is 4.7% higher than YOLOv8n, with the F1 value up by 2%, mAP50 up by 2.3%, and mAP95 up by 2.1%. This demonstrates the improvements of our EDGS-YOLOv8 across all metrics on the Det-Fly dataset compared to the original model.

### *4.5. Comparison of Small Object Detection Algorithms*

We evaluated the small target detection performance of EDGS-YOLOv8 across different domains using the VisDrone2019 dataset [51], known for its many small object instances spanning various scenarios and categorized into ten classes like people, cars, vans, buses, etc. This dataset is widely recognized and authoritative in small target detection. Table 5 compares precision, recall, F1, mAP50, and mAP95 metrics with the baseline YOLOv8n model. Our model demonstrated improvements of over 2% in both mAP50 and mAP95, along with enhancements in precision and recall. These results indicate that EDGS-YOLOv8 enhances detection accuracy and reduces missed detection rates compared to the baseline.

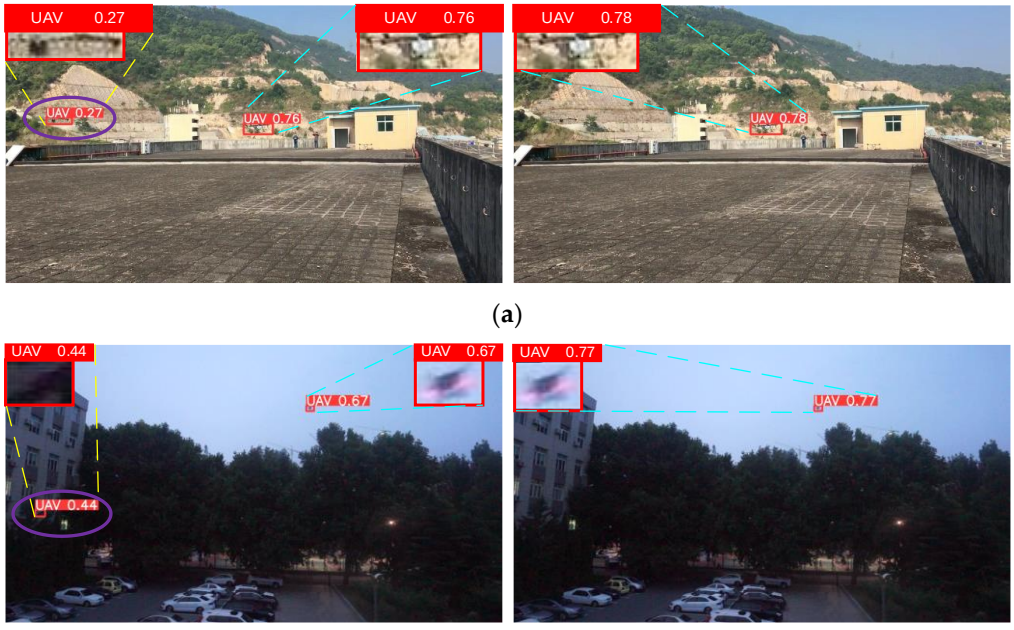
**Table 5.** Comparative experiments with the VisDrone2019 dataset.

| Model | Precision | Recall | F1 | mAP50 | mAP95 |
|---|---|---|---|---|---|
| YOLOv8n | 0.415 | 0.31 | 0.34 | 0.292 | 0.166 |
| Ours | 0.421 | 0.332 | 0.36 | 0.313 | 0.176 |

### *4.6. Visualization*

The figure below illustrates the labeled output image of our model on the DUT Anti-UAV dataset, providing a visual comparison of our proposed method with the baseline model under various conditions. We specifically selected this model for direct comparisons, considering the number of parameters in the proposed model and the minimal difference in GFLOPS compared to YOLOv8n to facilitate clear comparisons.

In Figures 11 and 12, we compare the output of our proposed model with the YOLOv8n baseline model. Figure 11 demonstrates a reduction in false detection rates attributed to the increased focus of the small target detection head on accurately identifying small targets. Simultaneously, the EMA enhances model robustness against changes in illumination and environmental interference, while the DDetect module improves local detail capture capabilities. Consequently, our proposed model performs better amidst ecological disturbances. For instance, in Figure 11b, the baseline model erroneously identifies a tree branch as a drone, a mistake not observed in our proposed model. This illustrates the superiority of our model for identifying small targets and navigating complex environments.




(**a**)

(**b**)

**Figure 11.** The outputs of the YOLOv8n baseline model (left panel) and the proposed model (right panel) were compared on sample images in (**a**) bright light and (**b**) low light scenarios. Those circled by purple ellipses are false detection objects. The model proposed in this paper performs better under the influence of different rays.

Figure 12 highlights the superiority of our proposed model in detecting drones under challenging conditions such as poor lighting, obstruction, and long distances. In the depicted scenario, the drone is small, distant, poorly illuminated, and partially obstructed in the image. The YOLOv8n baseline model struggled to detect this target drone. While both models may experience missed detections, our proposed model exhibits fewer missed detections due to its higher recall rate. For instance, in Figure 12a, the baseline model fails to detect drone targets in low light and at long distances, whereas in Figure 12b, it fails to

detect obscured drone targets. Conversely, our proposed model successfully detects these targets under such conditions.

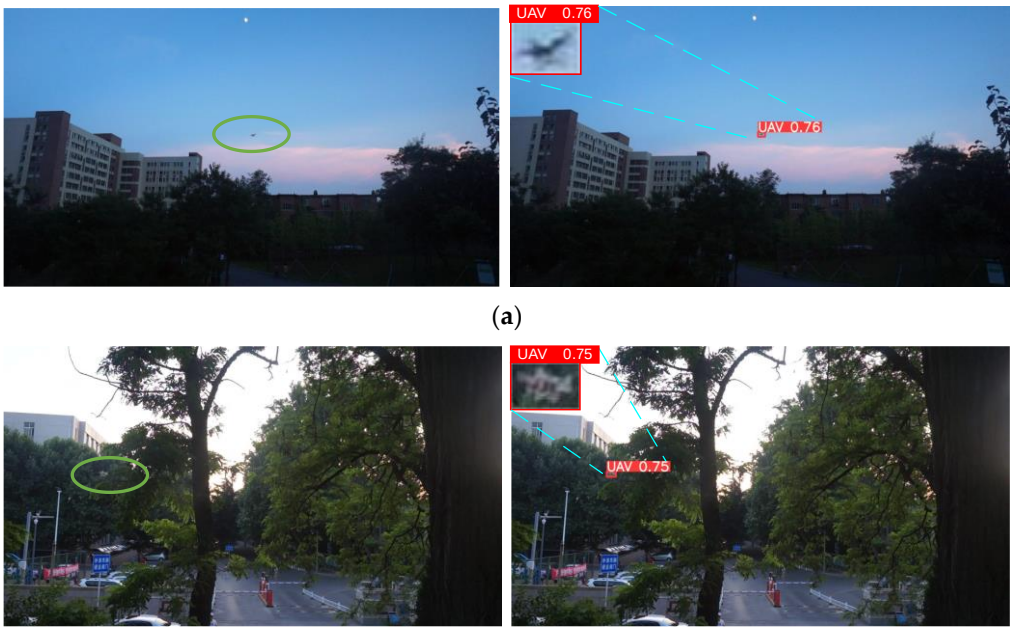



**Figure 12.** The outputs of the YOLOv8n baseline model (left panel) and the proposed model (right panel) were compared on sample images in (**a**) low light and (**b**) occlusion. Those circled by green ellipses are missed detection objects. The model is better regarding insufficient light and detecting obstructed objects.

## 5. Conclusions

The study introduces and evaluates an efficient UAV image detection model (EDGS-YOLOv8) using datasets such as DUT Anti-UAV, Det-Fly UAV, and VisDrone for small target detection. Compared to YOLOv8, our method offers the following enhancements: (a) improved network neck to enhance UAV detection accuracy by integrating a new detection head for small UAV targets, removing the insensitive detection head for small UAVs, and significantly reducing parameters while maintaining accuracy; (b) inclusion of EMA attention mechanisms in the neck network; (c) replacement of the C2f module and traditional convolution with the C3Ghost module and ghost convolution in the neck, reducing model complexity and enhancing inference speed; and (d) Utilization of the designed DDetect detection head instead of the original Detect, which not only reduces computational complexity but also improves the model's ability to capture details and further enhances detection accuracy. These modifications aim to enhance the model's accuracy in detecting target UAVs while reducing model complexity, resulting in a more efficient model. The proposed lightweight model achieves high detection accuracy with a size of only 4.23 MB, making it suitable for embedded devices and mobile terminals. We tested our proposed model with several benchmark YOLOv8 models and some recent UAV target detection models in the DUT Anti-UAV public dataset. Our method improves mAP50 by 1.2% compared to the high-precision model YOLOv8m, while the model volume and GFLOPS are smaller, and the FPS is also 41.2% higher than YOLOv8m. In addition, our method is smaller in terms of model volume and GFLOPS while achieving higher accuracy compared to the real-time and small model YOLOv8n. Using the Det-Fly anti-UAV image dataset and VisDrone2019, the proposed model's effectiveness and performance of small target detection are further evaluated. The proposed model performed better than the baseline YOLOv8n in accuracy, recall, 50, and 95. Both have been greatly improved, further proving the proposed model's advantages. In the future, the FPS of the model can be

further enhanced through new modifications to meet the higher-speed UAV identification requirements. To sum up, the UAV detection model proposed in this paper has good performance in real-time and precision and meets the requirements of rapid detection of UAVs in a real environment. The model can still maintain a high processing speed with limited computing resources, making it easy to deploy and apply in embedded systems to provide a feasible solution for real-time target monitoring of uncrewed aerial vehicles and other scenarios.

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

**Funding:** This research was funded by the Defense Industrial Technology Development Program (grant number JCKYS2022DC10).

**Data Availability Statement:** No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

## References

1. Wan, D.; Zhao, M.; Zhou, H.; Qi, F.; Chen, X.; Liang, G. Analysis of UAV patrol inspection technology suitable for distribution lines. *J. Phys. Conf. Ser.* **2022**, *2237*, 012009. [CrossRef]
2. Zhao, J.; Yang, W.; Wang, F.; Zhang, C. Research on UAV aided earthquake emergency system. *IOP Conf. Ser. Earth Environ. Sci.* **2020**, *610*, 012018. [CrossRef]
3. Zeybek, M. Accuracy assessment of direct georeferencing UAV images with onboard global navigation satellite system and comparison of CORS/RTK surveying methods. *Meas. Sci. Technol.* **2021**, *32*, 065402. [CrossRef]
4. 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]
5. Vattapparamban, E.; Güvenç, I.; Yurekli, A.I.; Akkaya, K.; Uluağaç, S. Drones for smart cities: Issues in cybersecurity, privacy, and public safety. In Proceedings of the 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), Paphos, Cyprus, 5–9 September 2016; pp. 216–221.
6. Mekdad, Y.; Aris, A.; Babun, L.; El Fergougui, A.; Conti, M.; Lazzeretti, R.; Uluagac, A.S. A survey on security and privacy issues of UAVs. *Comput. Netw.* **2023**, *224*, 109626. [CrossRef]
7. Mohammed, M.A.; Abd Ghani, M.K.; Arunkumar, N.; Hamed, R.I.; Abdullah, M.K.; Burhanuddin, M. A real time computer aided object detection of nasopharyngeal carcinoma using genetic algorithm and artificial neural network based on Haar feature fear. *Future Gener. Comput. Syst.* **2018**, *89*, 539–547. [CrossRef]
8. Yu, Z.; Shen, Y.; Shen, C. A real-time detection approach for bridge cracks based on YOLOv4-FPM. *Autom. Constr.* **2021**, *122*, 103514. [CrossRef]
9. Xie, J.; Huang, S.; Wei, D.; Zhang, Z. Multisensor Dynamic Alliance Control Problem Based on Fuzzy Set Theory in the Mission of Target Detecting and Tracking. *J. Sens.* **2022**, *2022*. [CrossRef]
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
11. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
12. Hu, Y.; Wu, X.; Zheng, G.; Liu, X. Object detection of UAV for anti-UAV based on improved YOLO v3. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8386–8390.
13. Zhai, H.; Zhang, Y. Target detection of low-altitude uav based on improved yolov3 network. *J. Robot.* **2022**, *2022*, 4065734. [CrossRef]
14. Dadrass Javan, F.; Samadzadegan, F.; Gholamshahi, M.; Ashatari Mahini, F. A modified YOLOv4 Deep Learning Network for vision-based UAV recognition. *Drones* **2022**, *6*, 160. [CrossRef]
15. Delleji, T.; Slimeni, F.; Fekih, H.; Jarray, A.; Boughanmi, W.; Kallel, A.; Chtourou, Z. An upgraded-YOLO with object augmentation: Mini-UAV detection under low-visibility conditions by improving deep neural networks. *Oper. Res. Forum* **2022**, *3*, 60. [CrossRef]
16. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.

17. 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]
18. Ma, J.; Huang, S.; Jin, D.; Wang, X.; Li, L.; Guo, Y. LA-YOLO: An effective detection model for multi-UAV under low altitude background. *Meas. Sci. Technol.* **2024**, *35*, 055401. [CrossRef]
19. Zhang, X.; Fan, K.; Hou, H.; Liu, C. Real-time detection of drones using channel and layer pruning, based on the yolov3-spp3 deep learning algorithm. *Micromachines* **2022**, *13*, 2199. [CrossRef] [PubMed]
20. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
21. Sun, H.; Yang, J.; Shen, J.; Liang, D.; Ning-Zhong, L.; Zhou, H. TIB-Net: Drone detection network with tiny iterative backbone. *IEEE Access* **2020**, *8*, 130697–130707. [CrossRef]
22. Dai, G.; Hu, L.; Fan, J.; Yan, S.; Li, R. A deep learning-based object detection scheme by improving YOLOv5 for sprouted potatoes datasets. *IEEE Access* **2022**, *10*, 85416–85428. [CrossRef]
23. Wang, L.; Cao, Y.; Wang, S.; Song, X.; Zhang, S.; Zhang, J.; Niu, J. Investigation into recognition algorithm of helmet violation based on YOLOv5-CBAM-DCN. *IEEE Access* **2022**, *10*, 60622–60632. [CrossRef]
24. 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* **2022**, *7*, 13. [CrossRef]
25. Bai, B.; Wang, J.; Li, J.; Yu, L.; Wen, J.; Han, Y. T-YOLO: A lightweight and efficient detection model for nutrient buds in complex tea-plantation environments. *J. Sci. Food Agric.* **2024**, *104*, 5698–5711. [CrossRef]
26. Zhou, X.; Yang, G.; Chen, Y.; Li, L.; Chen, B.M. VDTNet: A High-Performance Visual Network for Detecting and Tracking of Intruding Drones. *IEEE Trans. Intell. Transp. Syst.* **2024**. [CrossRef]
27. Chen, J.; Ma, A.; Huang, L.; Li, H.; Zhang, H.; Huang, Y.; Zhu, T. Efficient and lightweight grape and picking point synchronous detection model based on key point detection. *Comput. Electron. Agric.* **2024**, *217*, 108612. [CrossRef]
28. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A modified YOLOv8 detection network for UAV aerial image recognition. *Drones* **2023**, *7*, 304. [CrossRef]
29. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. 2023. Available online: https://github.com/ultralytics/ultralytics/blob/main/CITATION.cff (accessed on 30 June 2023).
30. Jocher, G.; Stoken, A.; Borovec, J.; Chaurasia, A.; Changyu, L.; Hogan, A.; Hajek, J.; Diaconu, L.; Kwon, Y.; Defretin, Y. ultralytics/yolov5: v5. 0-YOLOv5-P6 1280 models, AWS, Supervise. ly and YouTube integrations. Zenodo. *Zenodo.* 2021. Available online: https://github.com/ultralytics/yolov5 (accessed on 18 May 2020).
31. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
32. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
33. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
34. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient multi-scale attention module with cross-spatial learning. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
35. Jaderberg, M.; Simonyan, K.; Zisserman, A. Spatial transformer networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef]
36. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
37. Zhang, C.; Kim, J. Object detection with location-aware deformable convolution and backward attention filtering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9452–9461.
38. Deng, L.; Yang, M.; Li, H.; Li, T.; Hu, B.; Wang, C. Restricted deformable convolution-based road scene semantic segmentation using surround view cameras. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 4350–4362. [CrossRef]
39. Liu, Z.; Yang, B.; Duan, G.; Tan, J. Visual defect inspection of metal part surface via deformable convolution and concatenate feature pyramid neural networks. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 9681–9694. [CrossRef]
40. Chen, G.; Wang, W.; He, Z.; Wang, L.; Yuan, Y.; Zhang, D.; Zhang, J.; Zhu, P.; Van Gool, L.; Han, J. VisDrone-MOT2021: The vision meets drone multiple object tracking challenge results. Proceedings of IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 2839–2846.
41. Wang, R.; Shivanna, R.; Cheng, D.; Jain, S.; Lin, D.; Hong, L.; Chi, E. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 1785–1797.
42. 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]
43. Rodriguez-Ramos, A.; Rodriguez-Vazquez, J.; Sampedro, C.; Campoy, P. Adaptive inattentional framework for video object detection with reward-conditional training. *IEEE Access* **2020**, *8*, 124451–124466. [CrossRef]

44. Li, J.; Ye, D.H.; Chung, T.; Kolsch, M.; Wachs, J.; Bouman, C. Multi-target detection and tracking from a single camera in Unmanned Aerial Vehicles (UAVs). In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4992–4997.
45. Jiang, N.; Wang, K.; Peng, X.; Yu, X.; Wang, Q.; Xing, J.; Li, G.; Zhao, J.; Guo, G.; Han, Z. Anti-UAV: A large multi-modal benchmark for UAV tracking. *arXiv* **2021**, arXiv:2101.08466.
46. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
47. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9759–9768.
48. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
49. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings Part I 14. pp. 21–37.
50. Zheng, Y.; Chen, Z.; Lv, D.; Li, Z.; Lan, Z.; Zhao, S. Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1020–1027. [CrossRef]
51. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019.