electronics

MDPI

Article

Lightweight Small Target Detection Algorithm with Multi-Feature Fusion

Rujin Yang [1], Jingwei Zhang [2], Xinna Shang [1,3,*] and Wenfa Li [1,4,*]

[1] Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing 100101, China; yangrujin6@gmail.com
[2] Vorovich Institute of Mathematics, Mechanics and Computer Science, Southern Federal University, Rostov-on-Don 344090, Russia; zhangjingwei@mail.ru
[3] College of Robotics, Beijing Union University, Beijing 100101, China
[4] Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China
* Correspondence: shangxinna@buu.edu.cn (X.S.); liwenfa@ustb.edu.cn (W.L.)

Abstract: Unmanned aerial vehicles (UAVs) are a highly sought-after technology with numerous applications in both military and non-military uses. The identification of targets is a crucial aspect of UAV applications, but there are challenges associated with complex detection models and difficulty in detecting small targets. To address these issues, this study proposes the lightweight L-YOLO algorithm for target detection tasks from a UAV perspective. The L-YOLO algorithm improves on YOLOv5 by improving the model's detection performance for small targets while reducing the number of parameters and computational effort. The GhostNet module replaces the relevant convolution in the YOLOv5 model to create a lightweight model. The EIoU loss is used as the loss function of the algorithm to accelerate convergence and improve regression accuracy. Furthermore, feature-level extensions based on YOLOv5 are implemented, and a new detection head is proposed to improve the model's detection accuracy for small targets. The size of the anchor boxes is redesigned to suit the small targets using the K-means++ clustering algorithm. The experiments were conducted on the VisDrone-2022 dataset, and the L-YOLO algorithm demonstrated a reduction in computational effort by 42.42% and number of parameters by 48.6% compared to the original algorithm. Furthermore, recall and mAP@0.5 improved by 2.1% and 1.4%, respectively. These results demonstrate that the L-YOLO algorithm not only has better detection performance for small targets but is also a lighter model, indicating promising prospects for target detection from a UAV perspective.

Keywords: UAV target detection; lightweighting; GhostNet module; EIoU loss; K-means++

Citation: Yang, R.; Zhang, J.; Shang, X.; Li, W. Lightweight Small Target Detection Algorithm with Multi-Feature Fusion. *Electronics* **2023**, *12*, 2739. https://doi.org/10.3390/electronics12122739

Academic Editor: Fernando De la Prieta Pintado

Received: 5 June 2023
Revised: 13 June 2023
Accepted: 15 June 2023
Published: 20 June 2023

1. Introduction

Advancements in unmanned aerial vehicle technology have enabled its widespread use in various industries, such as environmental surveys, forest fire prevention, and maritime rescue. Hence, target detection, a crucial component of UAV applications, has become a research hotspot in recent years.

Anchor-based and anchor-free target detection algorithms are currently the two types of deep-learning-based target detection algorithms. A two-stage anchor-based targeted detection algorithm differs from one-stage anchor-based target detection algorithms. One-stage target detection algorithms, such as the Single Shot MultiBox Detector (SSD) [1] and the You Only Look Once (YOLO) series [2–12], are fast but lack accuracy, particularly for small targets. As a result of its two-stage detection algorithm, the region-based convolutional neural networks class outperforms single-stage algorithms in terms of target detection accuracy. This type of algorithm has been represented in various studies, including those referenced in citations [13–16]. However, this improved accuracy comes at the cost of slower detection speed. Therefore, depending on the specific needs of the user, it may be necessary to consider both the accuracy and speed when selecting a target detection algorithm. Anchor-free

target detection algorithms follow a new approach to corner point detection, represented by CornerNet [17] and CenterNet [18], instead of using pre-defined frames.

Due to the operational environment of UAVs, target objects for detection are often small, and therefore, target detection algorithms applied to UAVs must be efficient in detecting small targets. However, conventional UAVs have limited processing power, making it challenging to deploy algorithms with large network sizes and computations. As a result, the number of parameters and computational power of the target detection algorithms embedded in UAVs must be considered. Lightweight neural networks optimized for low-power embedded devices, such as the MobileNet [19–21] and ShuffleNet [22,23] series, have been developed in recent years, but their detection performance is significantly lower. Therefore, it is crucial to simplify the algorithm model while ensuring its effectiveness in detecting small targets.

YOLOv5 is a highly popular single-stage target detection algorithm that has been gaining traction in recent years. It is widely used for target detection tasks, such as object recognition in images and videos. This algorithm is designed to be fast and efficient, making it an ideal choice for real-time applications where speed is crucial. In this study, the YOLOv5 model is combined with the lightweight network module, GhostNet [24], to reduce the number of parameters and computational effort, and the loss function is also modified. To improve the model's detection performance for small objects, a new feature prediction layer is designed and implemented. The study uses images from the VisDrone-2022 [25] dataset, obtained entirely by drones, as the detection target. The structure of this paper includes an introduction to related work, a description of the improvement methods adopted, a detailed demonstration of the method's effectiveness through experiments, and a summary.

2. Related Work

The detection of small targets is difficult due to their small size and low pixel density. Researchers have explored data augmentation techniques, contextual information, and multi-scale feature learning to enhance the performance of neural networks in detecting small targets. One proposed data augmentation method for addressing the issue of the limited number and diversity of small targets in a dataset is copy-pasting [26], which involves randomly duplicating small targets in the image. However, copy-pasting often results in issues such as scale and background mismatches, compromising the integrity of the image. To address these problems, researchers have proposed an adaptive copy-pasting method called AdaResampling [27]. Scale Match [28] also adjusts the scale of external datasets based on the scale of small targets in the dataset and integrates them into the training set to improve the feature representation of small targets.

Contextual information refers to the relationship between the pixels of a specific target and its neighboring objects, such as the contextual information around a person's eyes including their eyebrows and nose. The contextual feature information around a target can be useful for object recognition during detection. The SODet [29] backbone network utilizes the global computational properties of the Transformer [30] to establish connections between objects in an image that is relatively far from a particular target while using convolutional neural networks (CNN) to extract local information from the image. The Feature-Fused SSD [31] algorithm reconstructs the image back into pixel space through deconvolution, thus visualizing and finding the most suitable and effective receptive field as a small target for feature fusion, thereby enhancing the connection between contexts and improving the detection accuracy of the algorithm for small targets. Combining FA-SSD [32] extracts contextual information from the surrounding pixels of a small target and connects it with contextual features in tandem to enrich the features of small targets, thus enabling the model to better detect targets.

As a result of developing methods for extracting useful information from images of various sizes and feature maps of various scales, researchers have been able to improve the accuracy of detection of small targets by performing feature extraction and predicting fea-

tures from multiple scaled feature maps. The SSD algorithm detects objects by performing softmax classification and position regression on multiple feature scales, with each scale responsible for detecting objects of different sizes. DSSD [33] builds on SSD by replacing its original VGG16 [34] backbone network with Resnet-101 [35], which has a deeper network level and stronger feature expression ability, and by adopting a feature fusion method to fuse the feature information of different layers together. Similarly, FPN [36] adds an upsampling and side connection structure to SSD, significantly improving target detection accuracy. PANet [37] shortens the information transfer distance between bottom-level and top-level features using a bottom-up path augmentation method, while BiFPN [38] applies a bidirectional path to each feature layer for feature fusion and repeats the fusion process multiple times to achieve higher-level multi-scale feature enhancement. QueryDet [39] uses a novel query algorithm to radically speed up the process of object detection based on the feature pyramid.

Embedded devices often have limited computing power and storage space, making it difficult to deploy large neural network models. To address this issue, researchers have focused on developing lightweight neural network designs. The MobileNet series, developed by Google, uses depthwise separable convolution as the basic unit to create efficient and lightweight CNN models. The ShuffleNet series, developed by Megvii Technology, achieves a balance between model performance and computational load with low memory and computing power. PP-LCNet [40], proposed by the Baidu team, is a lightweight CPU network that improves the performance of lightweight models on multi-tasking.

3. L-YOLO

Owing to the limitations of a UAV's own on-board processor and power losses, there are few parameters for target detection algorithms applied to UAVs. As UAVs often operate at high altitudes and the scale of target detectors varies highly, algorithms embedded in UAVs need to consider the detection performance of small targets while ensuring conventional target detection. Therefore, enhancing the algorithm model to simultaneously meet the requirements of low power consumption and efficient small target detection is a problem that must be addressed.

The speed at which YOLOv5 detects targets is good, but its accuracy is not as good as for a typical two-stage detection algorithm. Two objectives are achieved in this study with the L-YOLO algorithm proposed in this study as an improvement to the YOLOv5 algorithm:

(1) Make the algorithm model more suitable for embedded devices with limited hardware conditions by reducing its parameters.

(2) Improve small target detection performance by further optimizing the model.

3.1. L-YOLO Model

In this study, a small target detection model named L-YOLO is proposed. This model uses the YOLOv5 detection algorithm as its basis. As part of L-YOLO, GhostNet is introduced into the backbone network and neck of the YOLOv5 model, and additionally, loss function and feature prediction layers are modified. The model of L-YOLO is shown in Figure 1.

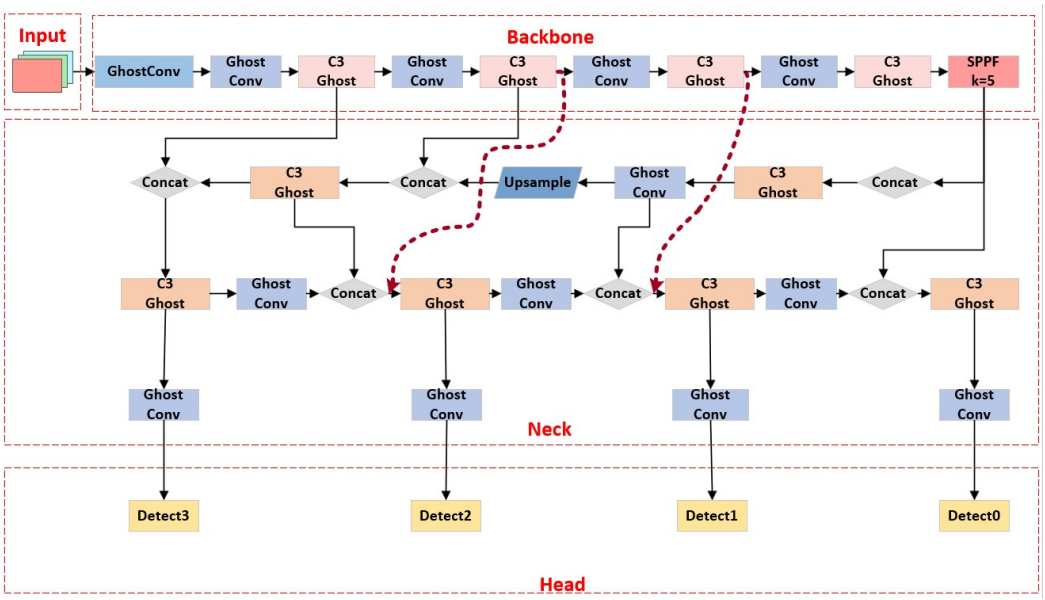


Figure 1. L-YOLO model structure.

3.2. L-YOLO Model

Given the specific limits of the drone, the computing power of the deep learning model embedded in it is relatively limited. In the original network structure of YOLOv5, a large amount of redundant data is generated when extracting image features, which occupies hardware storage space, reduces computing speed, and does not meet the requirements of rapid detection. For the algorithm model to adapt to the UAV equipment, this study introduces the GhostNet network structure, which is specially designed for mobile equipment.

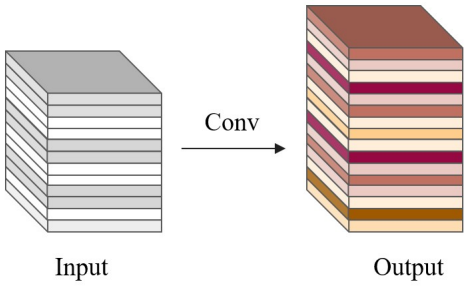
The majority of convolution operations begin with point convolution for dimensionality reduction and end with depth convolution for feature extraction. In addition to extracting more feature information from an input image, CNN-trained neural networks also generate more redundant feature maps. In addition to enhancing the performance of a model, performing numerous convolution operations increases memory and computing resource consumption. Most lightweight networks today achieve lightweight effects by removing some redundant features. GhostNet combines standard convolution and linear operations while maintaining the original network's output feature map and channel size. In this way, parameterization and computation are simplified.

The Huawei Noah's Ark Laboratory has proposed GhostNet, a lightweight network for feature extraction. The Ghost module can generate more features with less computation. Figure 2a is an ordinary convolution structure, and Figure 2b is the convolution structure in the Ghost module. The Ghost module divides the original convolution into two parts, first generating a small number of feature maps using fewer convolution kernels, then using resource-intensive linear operations to produce the remaining feature layers, and finally stitching all the feature layers together to expand the target feature map.

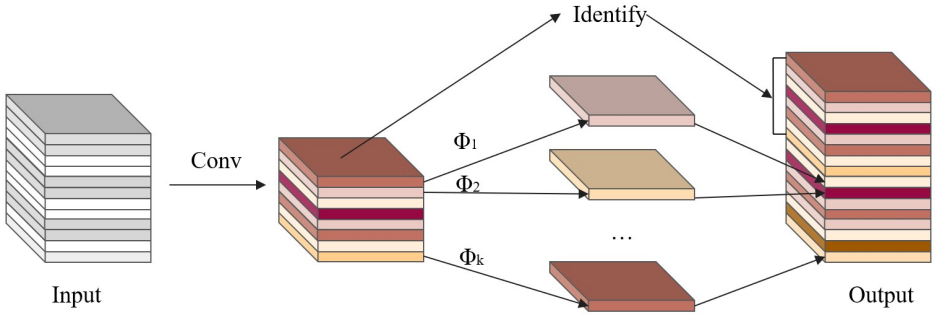
Ordinary convolution calculations convolve three channels simultaneously and provide a single value as output. Depthwise separable convolution splits the traditional convolution into two steps. First, the three channels are convoluted to obtain three separate values, and then, these three values are passed through a pointwise kernel with a size of $1 \times 1 \times 3$ to obtain the final value. For images of $H \times W$ with the same size, when the offset is not considered, the parameters and calculations are as follows:

$$P_T = c \times L \times L + c \times X \tag{1}$$

$$F_T = H \times W \times c \times L \times L + c \times X \times H \times W \tag{2}$$



(**a**) Ordinary convolutional layers



(**b**) The Ghost Module

Figure 2. Illustration of the normal convolution and Ghost convolution modules when the input and output are the same.

In the above equation, the convolution kernel size is represented by L, the input channel size is represented by C, the output channel size is indicated by X, and the input map size is H. The length of the input feature map is W, and the width is H. However, any convolution kernel can be of any size.

If each basic feature corresponds to S redundant features, then the kernel of a Ghost convolution is $D \times D$. For a GhostNet convolution, assuming the bias parameter is set to zero, the following parameters and calculations are generated:

$$P_{\text{ghost}} = X/S \times c \times L \times L + (S-1)/S \times X \times D \times D \tag{3}$$

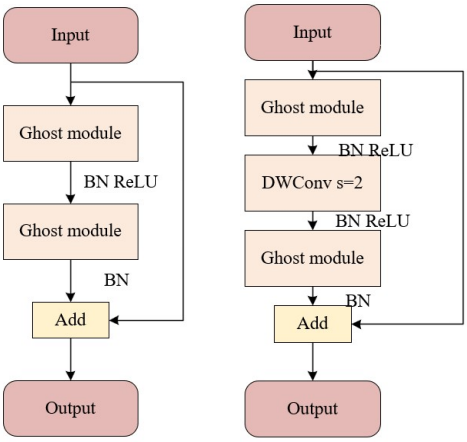
$$F_{\text{ghost}} = X/S \times H \times W \times c \times L \times L + (S-1)/S \times H \times W \times X \times D \times D \tag{4}$$

This gives the ratio of the number of parameters to the amount of computation for Ghost convolution versus conventional convolution, which can be expressed as:

$$R_P = \frac{P_{\text{ghost}}}{P_T} = \frac{X/S \times c \times L \times L + (S-1)/S \times X \times D \times D}{X \times c \times L \times L} \approx \frac{1}{S} \tag{5}$$

$$R_F = \frac{F_{\text{ghost}}}{F_T} = \frac{X/S \times c \times W \times H \times L \times L + (S-1)/S \times X \times W \times H \times D \times D}{X \times c \times W \times H \times L \times L} \approx \frac{1}{S} \tag{6}$$

The Ghost bottleneck was constructed based on the strengths and features of the Ghost module, as shown in Figure 3. It borrows the residual block structure from the ResNet model, integrating multiple convolutions and shortcuts. As shown in Figure 3a, the step size is 1, and after the two ghost modules, a batch normalization layer is added, followed by a Rectified Linear Unit activation function. Figure 3b utilizes a two-step convolution algorithm to downsample between two Ghost modules.



(**a**) Step size of 1 (**b**) Step size of 2

Figure 3. Ghost bottleneck structure.

Therefore, by introducing the GhostNet module, the original model has fewer parameters and requires less computing effort. In Figure 4, we see the model after GhostNet is introduced.

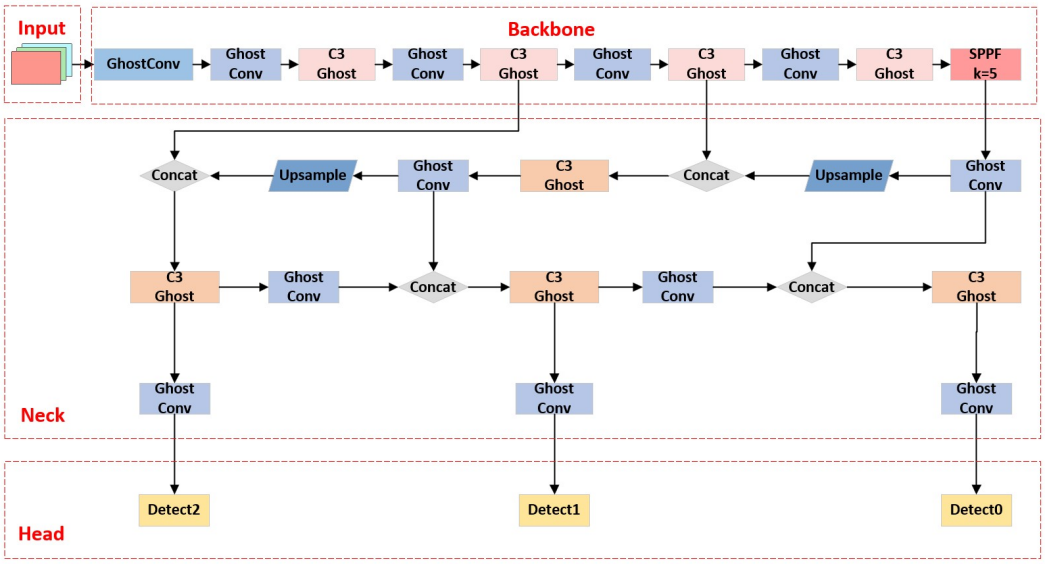


Figure 4. Achieving lightweight models.

3.3. Loss Function

In YOLOv5, the boundary loss is calculated using the Complete-IoU (CIoU) loss in order to determine the distance between the true bounding box and the predicted bounding box. This takes into account not only the overlapped area between the predicted and real frames but also the distance between their central points and their aspect ratios. The formula for this is as follows:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{7}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{8}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{9}$$

$\rho^2\left(b, b^{gt}\right)$ is the Euclidean distance between the two frames, IoU is the intersection ratio between them, c is the diagonal length of the smallest outer rectangle between them, v is the positive equilibrium parameter, and $\frac{w^{gt}}{h^{gt}}$ and $\frac{w}{h}$ is the aspect ratio consistency between the two frames.

It can be seen from Equation (8) that the penalty for this item in CIoU is no longer effective when the aspect ratio of the predicted frame satisfies $\left\{\left(w = kw^{gt}, h = kh^{gt}\right) \mid k \in R^{+}\right\}$, despite the fact that CIoU loss considers the distance between the centroids of the real and predicted frames, the overlap area, and the aspect ratio. Furthermore, we have

$$\frac{\partial v}{\partial w} = \frac{8}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right) \cdot \frac{h}{w^2 + h^2} \tag{10}$$

$$\frac{\partial v}{\partial h} = -\frac{8}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right) \cdot \frac{w}{w^2 + h^2} \tag{11}$$

Therefore,

$$\frac{\partial v}{\partial w} = -\frac{h}{w}\frac{\partial v}{\partial h} \tag{12}$$

The above equation shows that $\frac{\partial v}{\partial w}$ and $\frac{\partial v}{\partial h}$ are inversely related; that is, when the value of w or h increases during training, the other value is bound to decrease. Efficient-IoU (EIoU) loss is used as the loss function of the algorithm in this study to solve the two problems described above. According to CIoU loss, EIoU loss introduces information about the real and predicted frames' lengths and widths. Its formula is as follows:

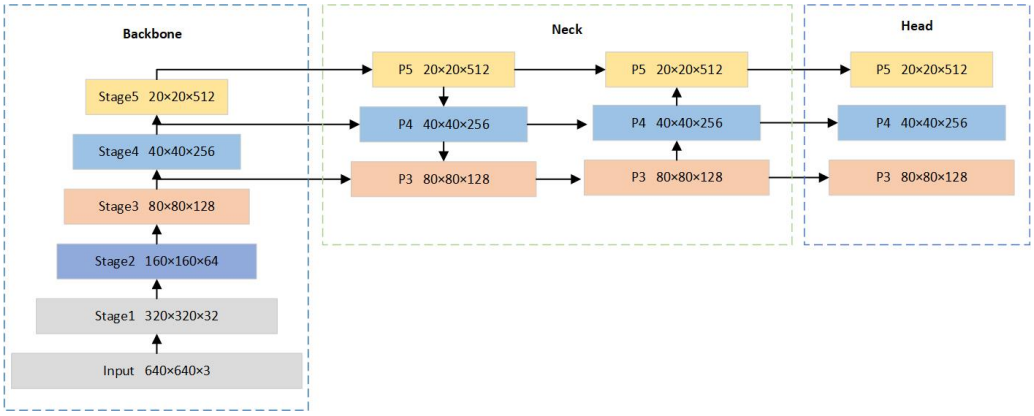
$$L_{EIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \frac{\rho^2\left(w, w^{gt}\right)}{{c_w}^2} + \frac{\rho^2\left(h, h^{gt}\right)}{{c_h}^2} \tag{13}$$

In this equation, c_w and c_h represent the width and height of the smallest bounding box covering the ground truth and predicted boxes, respectively. By using the EIoU loss as a basis, we can split the aspect ratio loss into the predicted width and height as well as a minimum bounding box. This results in faster convergence, better regression accuracy, and a focus on high-quality anchor boxes during regression. In addition, EIoU loss introduces Focal loss into its bounding box regression task, which optimizes sample imbalances.

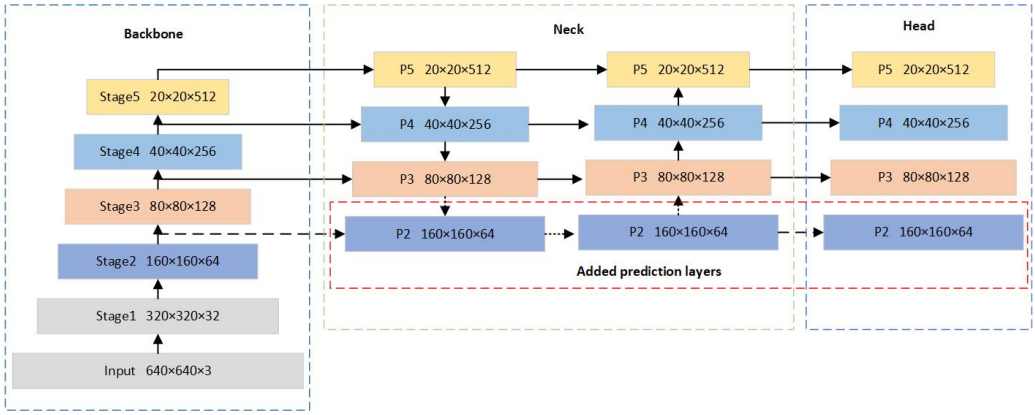
3.4. Prediction Feature Layer

Small sample sizes and the relatively high downsampling multiplier of the model contribute to the poor detection of small targets in YOLOv5. Due to the difficulty of learning features of small targets, shallower feature maps should incorporate a small target detection layer.

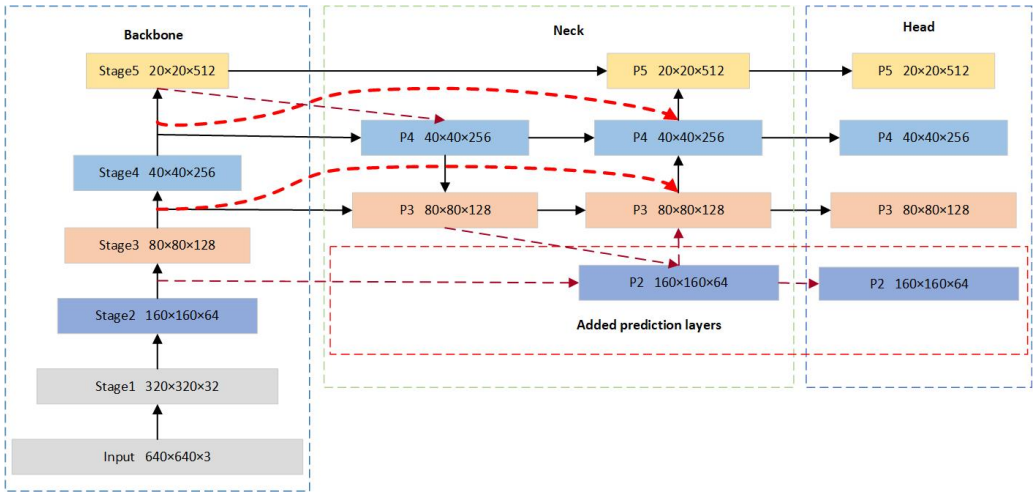
YOLOv5 originally performed feature prediction only in the last three C3 layers, as shown in Figure 5a. However, the detection of small targets is inadequate as it loses feature information during the continuous downsampling process. Hence, this study adds a feature prediction layer, as shown in Figure 5b. Predictions in the newly added layer are more precise, and small objects are less likely to be downsampled, which helps the model to gain insight. Inspired by BiFPN, this study improves the connection method of the feature fusion layer, as shown in Figure 5c. In the original PANet model, bottom-up and top-down path aggregation were used to improve multi-scale feature fusion. However, the input features of the bottom-up feature fusion stage had no original output features from the backbone network. Using cross-connection, this study removes nodes that do not contribute to feature fusion and adds skip connections between input and output nodes of the same scale to fuse more features. We consider each bidirectional path as a layer on the same feature scale. Higher-level feature fusion is achieved by reusing the same layer multiple times.



(**a**) YOLOv5 simplified structure diagram



(**b**) Simplified structure diagram of the prediction layer



(**c**) Simplified structure diagram of the model for high-level feature fusion

Figure 5. Implementation process of high-level feature fusion.

For the detection of small targets, the feature fusion network is enhanced with a second feature layer, as shown in Figure 5b. However, retaining extra shallow semantic information in the network leads to the loss of deep semantic information. The cross-scale connectivity approach adopted in this study can fuse more feature information without increasing the computational cost.

3.5. Anchor Box

The YOLOv5 algorithm obtains the anchor box size through edge clustering with the K-means [41] algorithm on the MS COCO [42] dataset, which is dominated by large and medium targets. This study used the VisDrone-2022 dataset, which contains a large number of small targets, so the anchor box size is not suitable for the dataset, as screening out inappropriate bounding boxes by the YOLO detection head would severely affect the model. To address this issue, this study modified the size of the anchor box in the VisDrone-2022 dataset using the K-means++ [43] clustering algorithm, which improved the model's detection accuracy for small targets.

By optimizing the selection of initial points, the K-means++ clustering algorithm improves the accuracy of classification results compared to K-means. For the VisDrone-2022 dataset, this study used K-means++ clustering to calculate anchor box sizes. Using this method, we selected the first cluster center randomly from the dataset and then chose the remaining cluster centers based on the distance between each sample x_i in the dataset and the initialized cluster centers, indicated by $D(x)$. Once the cluster centers were determined, we used the following formula for the relationship:

$$P(\mathrm{x}) = \frac{D(x)^2}{\sum_{i=1}^{N} D(x_i)^2} \tag{14}$$

The point with the highest probability value was chosen as the next clustering center. Each clustering center was selected in this manner until K were selected. In the dataset, each sample was assigned to the class with the smallest distance from the K cluster centers based on its distance to the K cluster centers. Continuous updates were performed until the cluster centers were fixed in their positions.

4. Experiments

An Intel Xeon Gold 5118 CPU@2.30 GHz CPU and NVIDIA Quadro P5000 16 G GPU were used in this experiment for model training, and the same platform was used for model test inference. The software ran on the Windows operating system and included Python 3.8.13, PyTorch 1.9.0, and the Cuda11.1 deep learning framework.

For the validation of the proposed method, the following experiments were conducted on VisDrone-2022:

(1) L-YOLO ablation experiments: L-YOLO, which is proposed in this study, is based on YOLOv5, with several improvements. Ablation experiments were conducted on the VisDrone-2022 dataset to verify the effect of each improvement on the detection process.

(2) Our comparison experiments with the most advanced target detection algorithms demonstrated L-YOLO's effectiveness.

4.1. Dataset

A traditional dataset has a relatively small proportion of small targets and an uneven distribution of them. As a result of uneven distributions, the model is biased toward learning large and medium targets during training. The VisDrone-2022 dataset, a professional dataset with predominantly small objects, was used to address this issue. A random selection of VisDrone-2022 images is shown in Figure 6.

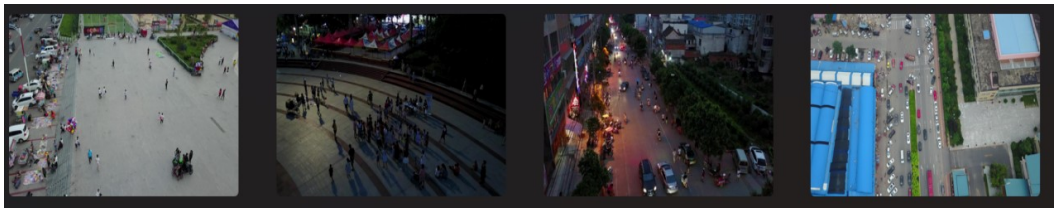


Figure 6. VisDrone-2022 images selected at random.

The VisDrone-2022 dataset was collected by the Machine Learning and Data Mining Laboratory at Tianjin University. Compared to MS COCO, this dataset contains twice as many small objects, thus making it suitable for detecting small targets. For each scale, Table 1 displays the percentages of targets based on these data.

Table 1. MS COCO and VisDrone-2022 scale target comparison (%).

Size (%)	MS COCO	VisDrone-2022
Small	41.43	87.77
Medium	34.33	11.97
Large	24.24	0.26

4.2. Ablation Experiment

As the UAV can only carry target detection algorithms with few parameters and low power consumption, this study proposes a lightweight L-YOLO model. This model features the GhostNet module as the neck and backbone networks. Upon incorporating the lightweight modules, to ensure the algorithm's detection performance, we also combined the original model with ShuffleNetV2, MobileNetV3, PP-LCNet, and GhostNet modules for a comparison experiment. The position and number of lightweight modules inserted in the model were consistent, and the experiment was conducted on the VisDrone-2022 dataset; the results are shown in Table 2.

Table 2. Experimental comparison of YOLOv5 in combination with different lightweight modules.

Models	R (%)	mAP@0.5 (%)	Parameters	GFLOPs
YOLOv5s	34.5	33.9	7.2M	16.5
ShuffleNetV2-YOLOv5s	22.5	19	2.7M	6.5
MobileNetV3-YOLOv5s	26.2	20.5	3.9M	7.3
PP-LCNet-YOLOv5s	27.4	27	3.8M	8.2
GhostNet-YOLOv5s	31.4	30.6	3.6M	8.1

As seen in Table 2, although the ShuffleNetV2, MobileNetV3, and PP-LCNet modules reduce the number of parameters and the computational complexity of the original model, the detection performance is also reduced by a large amount. In contrast, the algorithm model combined with the GhostNet module sacrifices recall and mAP values to a lesser extent but reduces the number of parameters and computational complexity. ShuffleNetV2 reduces the parameters of the original model to 2.7 M, and the calculation amount is reduced to 6.5 G, providing the best lightweight effect among the four modules. However, it also has the greatest impact on the detection of the model, reducing the recall rate by 12% and the mAP value by 14.9%. The introduction of the GhostNet module reduces the recall rate of the model by 3.1%, the mAP by 3.3%, the number of parameters by 50%, and the amount of calculations by 50.9%. This enables the model to achieve a lightweight effect, and the detection performance of the model is only slightly reduced. Therefore, the GhostNet module was inserted into the backbone network and neck of the algorithm model, replacing the initial complex convolution structure of the original algorithm.

When the GhostNet module was introduced, the algorithm's detection performance slightly degraded. Adding the feature detection layer to the model improved the performance of the detection algorithm by modifying the loss function. The ablation experiments performed on the VisDrone-2022 dataset were used to verify the effectiveness of the improvements proposed in this study. As a fair evaluation, this study kept the parameters of each variable consistent; the experimental results can be found in Table 3.

Table 3. Comparison of results of ablation experiments.

Methods	GhostNet	EIoU Loss	New Prediction Layer	New Anchor	R (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Parameters	GFLOPs
YOLOv5s baseline					34.5	33.9	18.2	7.2 M	16.5
M1	✓				31.4	30.6	15.6	3.6 M	8.1
M2		✓			35.8	35.3	19.2	7.2 M	16.5
M3			✓		39.4	39.2	22.3	7.3 M	19
M4				✓	35.6	34.7	18.9	7.2 M	16.5
M5	✓	✓			31.9	31.8	16.3	3.6 M	8.1
M6	✓	✓	✓		36.3	35	19.1	3.7 M	9.5
M7	✓	✓	✓	✓	36.6	35.3	19.2	3.7 M	9.5

All the proposed methods in this study were compared against YOLOv5s as the baseline, and the results showed that they all improved its efficiency. The first method involved replacing the convolutional blocks in the original model with GhostNet modules; owing to this, the model achieved the effect of lightweighting, but with a slight reduction in the detection performance.

The second method was to use EIoU loss as the loss function of the model. Model parameters are usually not changed by changing the loss function. The introduction of EIoU loss enhanced the detection ability of the model and increased the recall rate and mAP@0.5 by 1.3% and 1.4%, respectively.

The third method was to add a new feature prediction layer. As this was based on the original feature layer with an additional small target prediction layer and changes in the connection method, it led to an increase in the number of parameters and computation of the original model. The number of parameters of the model increased from 7.2 to 7.3 M, and the computation volume increased from 16.5 to 19 (Table 3). However, the detection ability of the model improved significantly, the recall rate increased by 4.9%, and mAP@0.5 increased from 33.9% to 39.2%.

For the VisDrone-2022 dataset, the fourth method involved resizing the anchor box using the K-means++ algorithm. The comparison results show that this method had no impact on the number of parameters and calculations of the model, but it did improve the detection performance of the model. The recall rate increased from 34.5% to 35.6%, and mAP@0.5 also increased by 0.8%.

The next three methods focused on developing a lightweight model with higher detection performance. The methods used in this study improved the efficiency of the model, and the final model not only reduced the number of parameters from 7.2 to 3.7 M but also reduced the number of calculations from 16.5 to 9.5 compared to the original YOLOv5s model (Table 3). Additionally, the detection performance of the model significantly improved, and the recall rate and mAP@0.5 increased by 2.1% and 1.4%, respectively, which fully proves the effectiveness of the proposed method.

4.3. Comparative Experiment

Experiments were conducted to compare L-YOLO with other state-of-the-art target detection algorithms to demonstrate the superiority of L-YOLO over other algorithms, and the results are shown in Table 4.

Table 4. Comparison with different target detection algorithms.

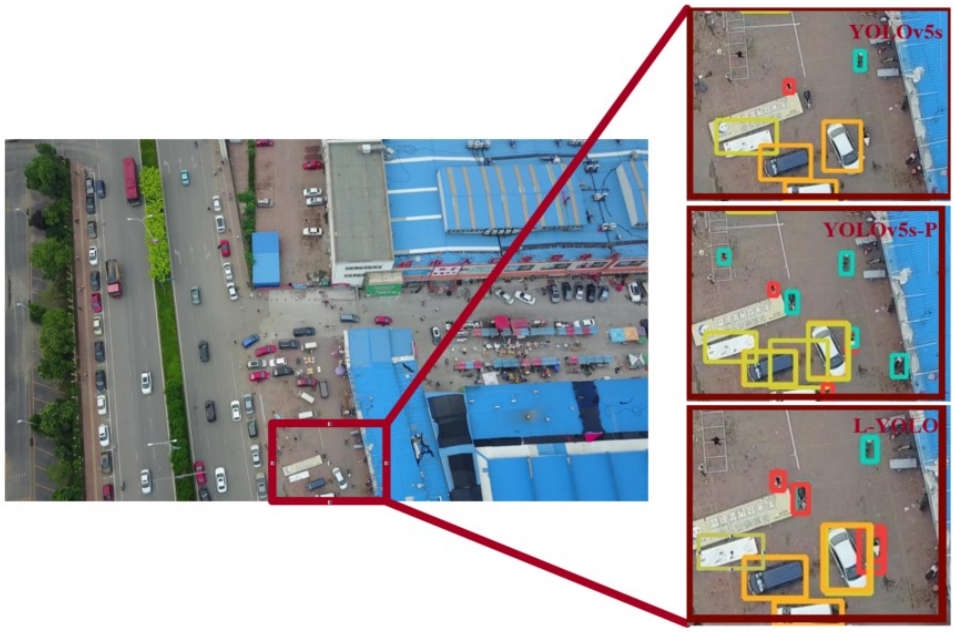
Methods	R (%)	mAP@0.5 (%)	Parameters	GFLOPs
SSD	35.5	23.9	24.5M	87.9
RetinaNet (ResNet-18)	37.9	21.2	19.8 M	93.7
YOLOv3	34.8	32.3	63 M	157.3
YOLOv5s	34.5	33.9	7.2 M	16.5
YOLOv5m	37.9	37.8	21.2 M	49
YOLOX-s	39.6	33.8	9.0 M	26.8
YOLOv7	39	34.5	36.9 M	104.7
YOLOv8s	39.8	39	11.2 M	28.5
L-YOLO (ours)	36.6	35.3	3.7 M	9.5

The results from Table 4 show that L-YOLO not only surpasses YOLOv5s in detection performance but also reduces the amount of parameters and calculations by a significant amount. The performance of L-YOLO compared with the early detection algorithms such as SSD and RetinaNet is high in all aspects. Although the recall of L-YOLO is slightly lower compared to YOLOX-s and YOLOv7, its mAP value is higher, and the number of parameters and computation is much lower. Compared to the latest YOLOv8s, L-YOLO has a slightly weaker detection performance, but the number of parameters and the number of computations are only about one-third of that of YOLOv8s. In summary, L-YOLO not only meets the lightweight requirement; it also has a strong detection performance.

Visual comparisons were made between images captured from different scenes in the VisDrone-2022 test set to determine L-YOLO's detection performance. The results are shown in Figure 7, where group (a) is a graph of detection results of YOLOv5s, group (b) is a graph of the detection results after changing the feature prediction layer, and group (c) is the detection result of the proposed L-YOLO model. We marked the main differences of the images with positive red boxes and used numbers to label them. Plot (d) shows randomly selected images from the results of a comparison of visualizations, with areas enlarged to facilitate visual comparison. Graphs comparing the detection effects of the groups show that group (b) shows the most effective detection effect, proving that the small target detection layer proposed in this study improves detection on small targets and reduces missed detections. The detection effect of L-YOLO, although inferior to that of group (b), is better than that of YOLOv5s. The same number in the diagram represents the same area. This shows that this study does reduce the detection effect of the model after the model is lightened, but by modifying the loss function and other aspects of optimization, the detection effect of L-YOLO exceeds that of the original YOLOv5s, which fully proves the effectiveness of the proposed method.

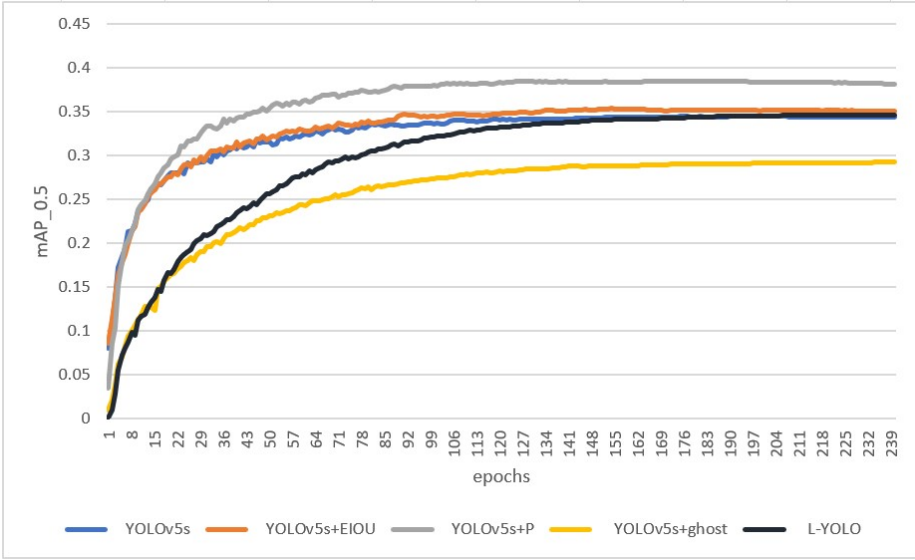
Figure 8 shows the experimental performance of different models, where YOLOv5s+P represents the model after adding the new detection layer proposed in this study. The results show that the detection performance of L-YOLO exceeds that of YOLOv5, proving that the proposed method reduces the computational and parametric quantities of the model and improves the detection performance of the model for small targets.

(**a**) YOLOv5s (**b**) YOLOv5s+P (**c**) L-YOLO

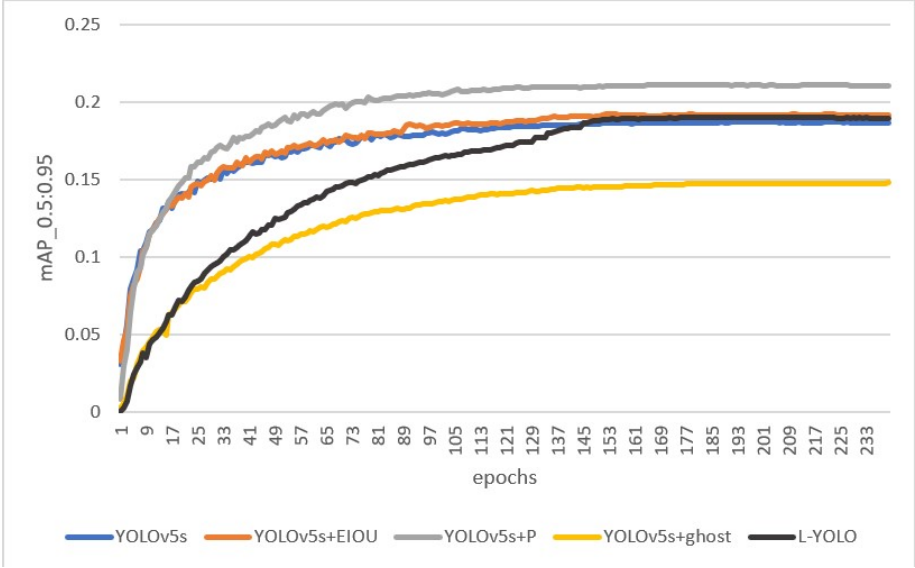


(**d**) Visualization versus zoom-in

Figure 7. Comparison of algorithm visualization effects.



(**a**) mAP@0.5 for the experiment.



(**b**) mAP@0.5:0.95 for the experiment.

Figure 8. mAP0.5, mAP0.5:0.95 for the experiment.

5. Conclusions

The small internal storage capacity and limited computing power of embedded devices make them difficult to use for large-scale data storage. Therefore, the target detection algorithms applied to embedded devices must be lightweight while demonstrating high detection performance. To address the above issues, this study proposes the L-YOLO algorithm using YOLOv5 as a baseline. In this study, the GhostNet module is introduced into the algorithm model, the loss function of the original model is modified, a new predictive feature layer is proposed, and an anchor box suitable for small targets is redesigned using the K-means++ clustering algorithm. This study tests the proposed algorithm on the VisDrone-2022 dataset. The experimental data show that, compared to the YOLOv5, the number of calculations of L-YOLO was reduced by 42.42%; this resulted in a 48.6% reduction in parameters. Simultaneously, the recall rate increased from 34.5% to 36.6%, and the mAP@0.5 also increased by 1.4%, proving that the proposed method not only reduces the number of parameters and number of calculations but also improves the detection performance of the model.

Author Contributions: Funding acquisition, W.L.; investigation, R.Y.; methodology, R.Y.; project administration, R.Y. and J.Z.; resources, W.L. and X.S.; software, R.Y.; supervision, W.L. and X.S.; writing—original draft, R.Y.; writing—review and editing, R.Y., X.S. and J.Z. 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 (Grant Nos. 61972040).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. 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; pp. 21–37.
2. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
3. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
4. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
5. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
6. Jocher, G. Yolov5. Code Repository. 2022. Available online: https://www.github.com/ultralytics/yolov5 (accessed on 14 June 2023).
7. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
8. 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. *arXiv* **2022**, arXiv:2207.02696.
9. JOCHER. Network Data. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 14 June 2023).
10. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
11. Huang, X.; Wang, X.; Lv, W.; Bai, X.; Long, X.; Deng, K.; Dang, Q.; Han, S.; Liu, Q.; Hu, X.; et al. PP-YOLOv2: A practical object detector. *arXiv* **2021**, arXiv:2104.10419.
12. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. PP-YOLOE: An evolved version of YOLO. *arXiv* **2022**, arXiv:2203.16250.
13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
14. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1440–1448.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; p. 28.
16. Liang, T.; Bao, H.; Pan, W.; Pan, F. Traffic sign detection via improved sparse R-CNN for autonomous vehicles. *J. Adv. Transp.* **2022**, *2022*, 3825532 . [CrossRef]
17. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
18. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.
19. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
20. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
21. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 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.
22. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.

23. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
24. 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, 14–19 June 2020; pp. 1580–1589.
25. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7380–7399. [CrossRef] [PubMed]
26. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Augmentation for small object detection. *arXiv* **2019**, arXiv:1902.07296.
27. Chen, C.; Zhang, Y.; Lv, Q.; Wei, S.; Wang, X.; Sun, X.; Dong, J. Rrnet: A hybrid detector for object detection in drone-captured images. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4917–4926.
28. Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; Han, Z. Scale match for tiny person detection. In Proceedings of the IEEE/CVF Winter conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1257–1265.
29. Zhao L.; Liu, S.P. Small Target Detection Algorithm Based on Adaptive Fusion of Global and Local Image Features. 2022. Available online: https://xueshu.baidu.com/usercenter/paper/show?paperid=1d2w06s0an6r0rw01k660ex0kj632154&site=xueshu_se (accessed on 14 June 2023)
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; p. 30.
31. Cao, G.; Xie, X.; Yang, W.; Liao, Q.; Shi, G.; Wu, J. Feature-fused SSD: Fast detection for small objects. In Proceedings of the Ninth International Conference on Graphic and Image Processing (ICGIP 2017), Qingdao, China, 14–16 October 2017; Volume 10615, pp. 381–388.
32. Lim, J.S.; Astrid, M.; Yoon, H.J.; Lee, S.I. Small object detection using context and attention. In Proceedings of the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Republic of Korea, 13–16 April 2021; pp. 181–186.
33. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
35. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
36. 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.
37. 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.
38. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
39. Yang, C.; Huang, Z.; Wang, N. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13668–13677.
40. 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.
41. MacQueen, J. Classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Los Angeles, LA, USA, 1 January 1967; pp. 281–297.
42. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
43. Arthur, D.; Vassilvitskii, S. K-means++ the advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans LA, USA, 7–9 January 2007; pp. 1027–1035.