remote sensing

MDPI

Article

A Lightweight Object Detector Based on Spatial-Coordinate Self-Attention for UAV Aerial Images

Chen Liu [1], Degang Yang [1,*], Liu Tang [1], Xun Zhou [2] and Yi Deng [1]

[1] College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China
[2] Party School of Yibin Committee of Communist Party of China, Yibin 644000, China
* Correspondence: yangdg@cqnu.edu.cn

Abstract: Object detection is one of the most widespread applications for numerous Unmanned Aerial Vehicle (UAV) tasks. Due to the shooting angle and flying height of the UAV, compared with general scenarios, small objects account for a large proportion of aerial images, and common object detectors are not extremely effective in aerial images. Moreover, since the computing resources of UAV platforms are generally limited, the deployment of common detectors with a large number of parameters on UAV platforms is difficult. This paper proposes a lightweight object detector YOLO-UAVlite for aerial images. Firstly, the spatial attention module and coordinate attention module are modified and combined to form a novel Spatial-Coordinate Self-Attention (SCSA) module, which integrates spatial, location, and channel information to enhance object representation. On this basis, we construct a lightweight backbone, named SCSAshufflenet, which combines the Enhanced ShuffleNet (ES) network with the proposed SCSA module to improve feature extraction and reduce model size. Secondly, we propose an improved feature pyramid model, namely Slim-BiFPN, where we construct new lightweight convolutional blocks to reduce the information loss during the feature map fusion process while reducing the model weights. Finally, the localization loss function is modified to increase the bounding box regression rate while improving the localization accuracy. Extensive experiments conducted on the VisDrone-DET2021 dataset indicate that, compared with the YOLOv5-N baseline, the proposed YOLO-UAVlite reduces the number of parameters by 25.8% and achieves gains of 10.9% in mAP0.50. Compared with other lightweight detectors, both the mAP and the number of parameters are improved.

Keywords: object detection; aerial images; attention mechanism; lightweight network; unmanned aerial vehicle (UAV)

Citation: Liu, C.; Yang, D.; Tang, L.; Zhou, X.; Deng, Y. A Lightweight Object Detector Based on Spatial-Coordinate Self-Attention for UAV Aerial Images. *Remote Sens.* **2023**, *15*, 83. https://doi.org/10.3390/rs15010083

Academic Editor: Riccardo Roncella

Received: 14 October 2022
Revised: 1 December 2022
Accepted: 21 December 2022
Published: 23 December 2022

1. Introduction

Compared with fixed surveillance cameras on the ground, cameras on UAV are easy to deploy and have a wider field of view. With the development of deep learning, UAV are tightly integrated with computer vision technology, making UAV less dependent on manual control and more intelligent, efficient, and convenient, widely used in smart agriculture [1], rescue search [2], and smart cities [3]. Object detection is not only a fundamental task in computer vision, but also a core problem in UAV applications.

Deep-learning-based object detectors currently are classified into two categories: one is the two-stage object detector, which includes the R-CNN [4], Fast-RCNN [5], mask R-CNN [6], etc. The R-CNN method, established by Girshick et al., is the first two-stage object detection algorithm that generates object candidate regions, then extracts features using a convolutional neural network, and applies a support vector machines classifier for classification. Mask R-CNN adds another parallel branch for pixel-level object instance segmentation. It detects not only the bounding box of an object but also the exact pixel containing the object. Another strategy is the single-stage object detector, consisting of anchor-based networks and anchor-free networks. The anchor-based networks include

YOLO [7] and SSD [8]. This type of algorithm considers object detection to be a regression problem, forgoing the phase to generate proposal boxes and reduce computation and time consumption. SSD is the first single-stage detector that is mapping the equivalent to a current two-stage detector while maintaining real-time speed. Joseph et al. proposed YOLO in 2015, which transforms the detection problem into a regression problem. YOLO reduces the computational cost by dropping the step of generating proposal frames, but the model is not highly effective at detecting small objects due to using only the last layer of feature maps. YOLOv2 [9] and YOLOv3 [10] replace GoogLeNet [11] as the backbone with darknet-19 and darknet-53, respectively, and use global average pooling and Batch Normalization (BN) [12] to improve network convergence. YOLOv4 [13] uses Complete Intersection over Union (CIoU) [14] loss for prediction frame filtering to improve the convergence of the model. YOLOv5 [15] uses a Feature Pyramid Network (FPN) [16] and Pixel Aggregation Network (PAN) [17] structure in the neck network. Due to the lighter model size, its accuracy is comparable to YOLOv4, but its speed is better. Moreover, anchor-free networks such as NanoDet [18]. NanoDet is a single-stage anchor-free object detection model that removes all convolutions from the PAN, and the generalized focal loss removes the center-ness branch. Thus, a lightweight model is achieved while maintaining high accuracy, thus achieving a lightweight model while maintaining high accuracy.

Although these object detectors perform well, these detectors typically focus on detection in general scenarios rather than aerial images, which is considerably different from general scenes. Figure 1 shows the detected objects of the general scenarios dataset MS COCO [19] and the aerial images dataset VisDrone-DET2021 [20]. There are considerably more small objects in the VisDrone-DET2021 dataset compared with MS COCO. "Small" here does not mean that the actual instance size is small, but rather the small scale of the objects in the image. Large objects have rich features and are easier to learn by neural networks, while small objects will lose part of their features during downsampling. The network is not able to learn sufficient small object features, resulting in a poor network fit to small objects, and therefore directly applying the common detectors to aerial images is not extremely effective.

(a) MS COCO (b) MS COCO

(c) VisDrone-DET2021 (d) VisDrone-DET2021

Figure 1. Comparison of example images from MS COCO and VisDrone-DET2021 datasets.

To solve this problem, many scholars have proposed their solutions. Yang et al. [21] proposed a Cluster Detection network for numerous small objects and uneven distribution of objects in aerial images. The network uses the methods of region clustering, slice detection, and scale adaptation to improve the running speed and small object detection

rate of the two-stage object detector on high-resolution aerial images. Wu et al. [22] proposed an object detector that fuses feature extraction, enhancement strategy, image pyramid network, and feature learning strategy. Pang et al. [23] proposed a Tiny-Net approach to object detection. Images can be processed in 29.4 seconds in a convolutional neural network consisting of a backbone Tiny-Net, an intermediate global attention block, a final classifier, and a detector. Tiny-Net is a lightweight residual structure that supports fast and powerful feature extraction from the input. In 2020, Cui et al. [24] proposed a context-aware module that can improve the recognition accuracy of small objects by using contextual information. YOLO-ACN [25] introduced an attention mechanism in the channel and spatial dimensions of each residual block to focus on small objects. The CIoU loss is employed to achieve accurate bounding box regression. Xu et al. [26] resulted in more accurate information in the predicted boxes by locating a quadrilateral to represent an object by learning the offsets of four points on a non-rotating rectangle. However, as the detection model requires angular information acquisition, it requires additional computational parameters, which are computationally expensive.

There are real-time and lightweight requirements for the application scenario of aerial image object detectors. Generally speaking, the computational resources of UAV platforms are limited, and a significant increase in the detection accuracy of the model would be at the expense of detection speed, which is a serious blow to real-time performance. Conversely, an over-emphasis on speed results in a significant loss of detection accuracy. Therefore, it is essential to balance the relationship between accuracy and speed. Considering the impact of the model on the detection speed, the faster single-stage object detector is more suitable for the UAV platforms, and the YOLOv5 detection method is finally selected. Regarding how to improve the efficiency of the object detector in aerial images, this study mainly focuses on the following two aspects: (1) reduce the size of the detector model through lightweight design and (2) improve the detection accuracy of small objects.

In consideration of the above discussion, in this paper, we design a lightweight object detector for aerial images, named YOLO-UAVlite. The main contributions of this study are recapitulated as follows:

1. The SCSA module is proposed by modifying the spatial and coordinate attention modules and combining their advantages to capture not only location information and channel information, but also spatial information. The module enhances the object representation and enables the model to locate the object area accurately. We design a lightweight backbone network, SCSAshufflenet, based on SCSA and ES to improve feature extraction and increase detection accuracy. Benefiting from the lightweight design, the model parameters are decreased.
2. A lightweight Slim-BiFPN structure is proposed to replace the original FPN. We design new convolutional blocks to reduce the number of convolutional layers, and decrease the computational complexity and parameter quantity of the detector, improving the efficiency of object detection.
3. The bounding box regression loss calculation model is modified to obtain higher quality boxes, which speeds up the convergence of the model and improves the positioning effect.

The remainder of this study is organized as follows: Section 2 reviews related works about lightweight neural networks, attention mechanisms, and ghost convolution. Details of the proposed model are presented in Section 3. Section 4 presents the implementation details of the experiments, giving the results and discussion of ablation and comparison experiments. Finally, the conclusions are given in Section 5.

2. Related Work

2.1. Lightweight Neural Network

The single-stage detector reduces the model size but still fails to achieve real-time detection of UAV aerial images due to a large number of parameters. To tackle the problem of large-scale neural network models, numerous scholars have proposed their own

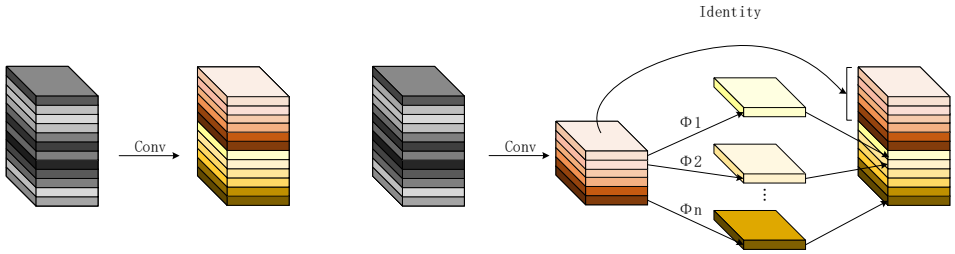
lightweight networks. In 2017, Howard et al. [27] proposed Depthwise Separable Convolution (DSC) in MobileNetV1, which decomposes the standard convolution into depthwise and pointwise convolution. The magnitude reduces the number of operations and parameters. In the same year, ShuffleNetV1 [28], based on DSC, has added group convolution and channel shuffle module, and the calculation amount and detection time were better than MobileNetV1. In 2018, ShuffleNetV2 [29] proposed four principles for designing a lightweight network and introduced channel split. The split structure replaces the add operation with the concatenate operation, reducing the number of model parameters. Based on the lightweight network ShuffleNetV2, Paddle [30] improved the network structure and proposes ES as backbone network, which outperforms ShuffleNetV2. In our work, we replace the original backbone darknet-53 with ES to reduce the model parameters.

2.2. Attention Mechanism

The attention mechanism adopted in computer vision allows the model to focus on significant areas of the input object, and the model tends to pay more attention to local information in the image that is helpful for judgment. Attention is divided into two main strategies: channel attention and spatial attention. The classic channel attention mechanism, Squeeze-and-Excitation Networks (SE) [31], is used to solve the loss of convolutional pools by modeling channel relationships. The Efficient Channel Attention Module [32] removes the fully connected layer and adopts a one-dimensional convolutional layer to reduce model redundancy, which can capture the information of cross-channel interactions. However, both SE and Efficient Channel Attention Module ignore spatial information, which is significant for capturing object structure in object detection tasks. The Convolutional Block Attention Module (CBAM) [33] integrates channel attention with spatial attention. The location information is obtained by using large-scale convolutions, but it only captures local relationships and not long-range dependencies on the feature map. Meanwhile, CBAM requires expensive computing, which is not suitable for lightweight networks. The coordinate attention [34] module embeds the location information into the channel attention, bringing a significant gain for intensive prediction tasks. In this paper, we modify CBAM and coordinate attention for small objects, and combine them to form a new attention module, which aims to capture the area of interest and improve the detection accuracy of our object detector.

2.3. Ghost Convolution

Currently, most convolution operations are point-by-point convolution for dimensionality reduction, followed by deep convolution for feature extraction. After training is complete, neural network models trained with traditional convolutional layers frequently produce numerous redundant feature maps. While this operation ensures a sufficient understanding of the input, it requires extensive convolutional layer calculation, which increases the calculated volume and the number of model parameters. GhostNet [35] prefers an inexpensive linear operation to obtain feature maps, thus reducing the model parameters and computational effort. Figure 2 shows the difference in structure between the standard convolution and the ghost convolution (ghostconv). The first regular convolution is performed, but with a reduced number of output channels, followed by a series of simple linear operations to generate more feature maps. Eventually, the two sets of feature maps are concatenated together in the specified dimension. Replacing standard convolution with ghostconv reduces the size of the model and the running time. Therefore, ghostconv is used in the SCSA to avoid generating a wide range of weight parameters.



(a) standard convolution (b) ghost convolution

Figure 2. The standard convolution versus ghost convolution.

3. Proposed Method

The overall structure of our YOLO-UAVlite detector is based on the YOLOv5-N object detector. We propose a novel attention module SCSA and add it to the ES module, which generates a lightweight backbone network SCSAshufflenet to enhance the feature extraction capability of the backbone network. Then, in the neck network, an improved feature pyramid model, called Slim-BiFPN, is proposed to reduce the model parameters and loss of information during the feature fusion process. Finally, we modify the loss function to obtain higher quality bounding boxes. Figure 3 shows the structure of the YOLO-UAVlite detector, and Table 1 summarizes the main parameters.

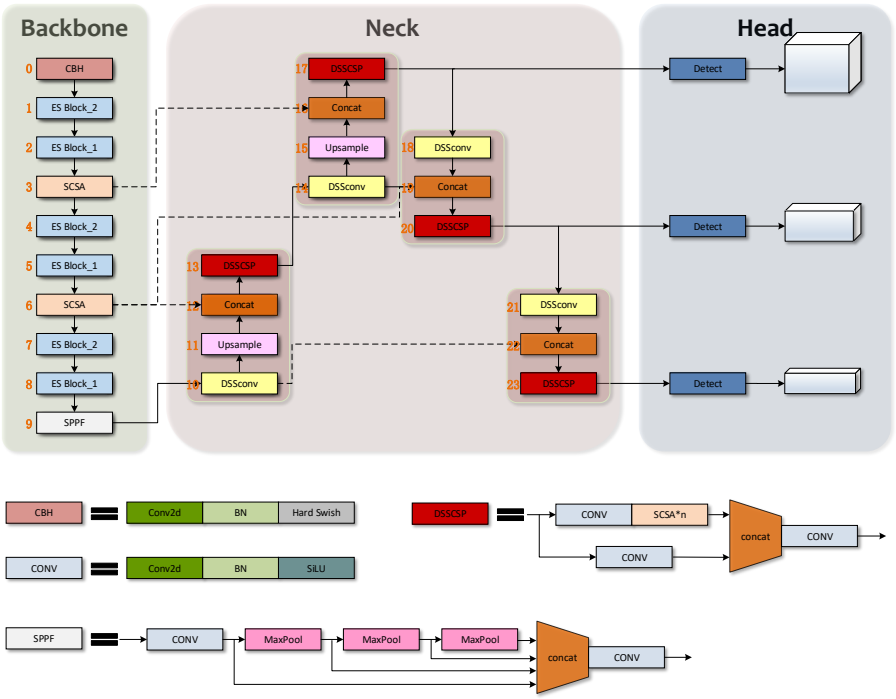


Figure 3. The structure of the YOLO-UAVlite detector. The backbone is SCSAshufflenet; the Neck uses the Slim-BiFPN structure; and the number of each module is marked with orange numbers on the left of the module.

As can be seen in Figure 3 and Table 1, the 640 × 640 size color image is first fed into the backbone, where it is downsampled by several convolutional layers and pooling layers. The first layer of the backbone is the CBH module, consisting of a convolutional layer, a BN layer, and a Hard Swish [36] activation function layer, aimed at reducing the computational effort of the model and accelerating the training speed. The ES Block extracts feature maps of different sizes by convolutional downsampling, and the attention mechanism strengthens object representation. The Spatial Pyramid Pooling-Fast (SPPF) module aims to

convert arbitrary-sized feature maps into fixed-sized feature vectors, which can expand the receptive field. Extract abstract features from images of sizes 160 × 160, 80 × 80 and 40 × 40. Then, the extracted features are fed into the neck, where the Slim-BiFPN approach is used for feature fusion. Low-level features contain more positional, detailed information, but are less semantic and more noisy due to the fewer convolutions they undergo. High-level features have more semantic information and are less sensitive to detail. By fusing three different levels of features, the advantages of multiple features are complemented to obtain more robust and accurate recognition results. Finally, there are three output tensors in the head for large, medium, and small object detection.

Table 1. Main modules and parameters of the YOLO-UAVlite algorithm.

Number	Module	Output	Repeat Times	Params
	Input	3 × 640 × 640		
0	CBH	32 × 320 × 320	1	928
1	ES Blcok_2	64 × 160 × 160	1	11,808
2	ES Blcok_1	64 × 160 × 160	2	20,736
3	SCSA	64 × 160 × 160	1	10,797
4	ES Blcok_2	128 × 80 × 80	1	43,072
5	ES Blcok_1	128 × 80 × 80	5	190,720
6	SCSA	128 × 80 × 80	1	37,901
7	ES Blcok_2	256 × 40 × 40	1	163,968
8	ES Blcok_1	256 × 40 × 40	2	291,840
9	SPPF	256 × 40 × 40	1	164,608
10	DSSconv	128 × 40 × 40	1	32,237
11	Upsample	128 × 80 × 80	1	0
12	Concat	256 × 80 × 80	1	0
13	DSSCSP	128 × 80 × 80	1	60,461
14	DSSconv	64 × 80 × 80	1	8989
15	Upsample	64 × 160 × 160	1	0
16	Concat	128 × 160 × 160	1	0
17	DSSCSP	64 × 160 × 160	1	15,933
18	DSSconv	64 × 80 × 80	1	23,325
19	Concat	128 × 80 × 80	1	0
20	DSSCSP	128 × 80 × 80	1	60,461
21	dDSSconv	128 × 40 × 40	1	89,581
22	Concat	256 × 40 × 40	1	0
23	DSSCSP	256 × 40 × 40	1	169,997
	Detect	3	1	20,295
	Total			1,417,657

3.1. SCSAshufflenet Backbone

The backbone network extracts feature maps of different sizes by adopting multiple convolutions and pooling from the input image. To avoid using too numerous convolutional layers and pooling layers, which will cause model parameters to be redundant, we propose the SCSAshufflenet backbone network to replace the original backbone network, which is composed of ES and SCSA. The ES removes some convolutional layers and reduces the model parameters. At the same time, the SCSA makes the backbone pay more attention to the features of small object objects and assigns more weights to small object features during the downsampling process, in order to obtain richer small object features. The SCSA and ES will be described in detail below, respectively.

3.1.1. Spatial-Coordinate Self-Attention

To improve the model's ability to extract useful features, we modify the spatial attention and coordinate attention and combine them to form new attention which integrates spatial, location, and channel information to capture the interrelationships between different types of information. As shown in Figure 4, our SCSA consists of two blocks: a spatial attention block and a coordinate attention block.

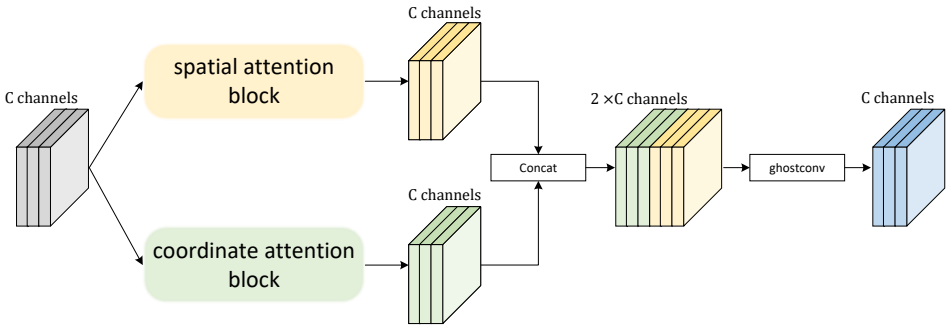
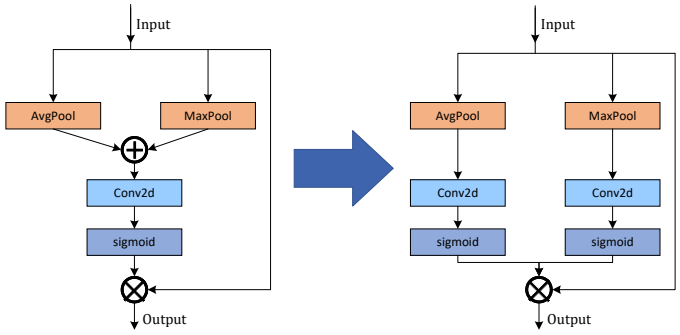


Figure 4. Structure of SCSA module.

As can be seen in Figure 5, we modified the spatial attention block in CBAM by convolving the MaxPool branch and the AvgPool branch, respectively, and then using the activation function on them to obtain two different focus points. Compared with the original attention module, more spatial feature information can be obtained. The final spatial attention block output feature map is formulated by multiplying these weights with the initial input feature map, as follows:

$$Y_{sa} = X \otimes \sigma(F_3(AvgPool(X)) + F_3(MaxPool(X))) \quad (1)$$

Here, Y_{sa} and X denote the spatial attention block output feature map and input feature map, respectively. $\otimes$ denotes the element-wise multiplication, σ is the sigmoid activation function. F_3 represents a convolution operation with the filter size of 3×3. $AvgPool$ and $MaxPool$ represent the average pooling operation and the max-pooling operation, respectively.



(a) spatial block (b) modified spatial block

Figure 5. Structure of spatial block and modified spatial block.

Coordinate attention decomposes the channel attention into two one-dimensional features and integration of features along both X and Y spatial directions, and efficiently integrates spatial coordinate information into the generated attention map. We modify the coordinate attention to make the feature map have different receptive fields through convolution, further enhancing the interdependence between the location information and channel relations. Figure 6 shows the structure of the modified coordinate attention block. Specifically, for the input feature map X ($C \times H \times W$), we obtain different receptive fields after the ghostconv module, so as to learn high dimensional features, and then fuse with the original input to obtain O. For the feature map O ($C \times H \times W$), the pooled with dimensions $(H, 1)$ and $(1, W)$ are used to augment the horizontal and vertical features of each channel,

respectively. Therefore, the output of the cth channel with width W and height H can be written, respectively, as follows:

$$Z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} O_c(h, i) \tag{2}$$

$$Z_c^w(w) = \frac{1}{H} \sum_{0 \le j \le W} O_c(h, j) \tag{3}$$

where Z_c and O_c are the output and the input of the cth channel, respectively. H and W denote the height and width of the input feature maps, respectively. Next, the two feature maps Z^h and Z^w are concatenated, and the concatenated feature map is reduced by the 1×1 convolution operation, yielding

$$f = \delta\left(F_1\left(\left[Z^h, Z^w\right]\right)\right) \tag{4}$$

where $f \in R^{C/r \times (H+W)}$ is the intermediate feature map of spatial information in the horizontal and vertical directions, δ denotes the nonlinear activation function, and F_1 represents the 1×1 convolution operation. Then, f is divided into two separate tensors f^h and f^w, as shown in the following formula:

$$f^h \in R^{C/r \times H} \tag{5}$$

$$f^w \in R^{C/r \times W} \tag{6}$$

where r is the reduction radio. f^h and f^w are transformed by two 1×1 convolutions to the same number of channels as O, thus yielding the following:

$$g^h = \sigma\left(F_1\left(f^h\right)\right) \tag{7}$$

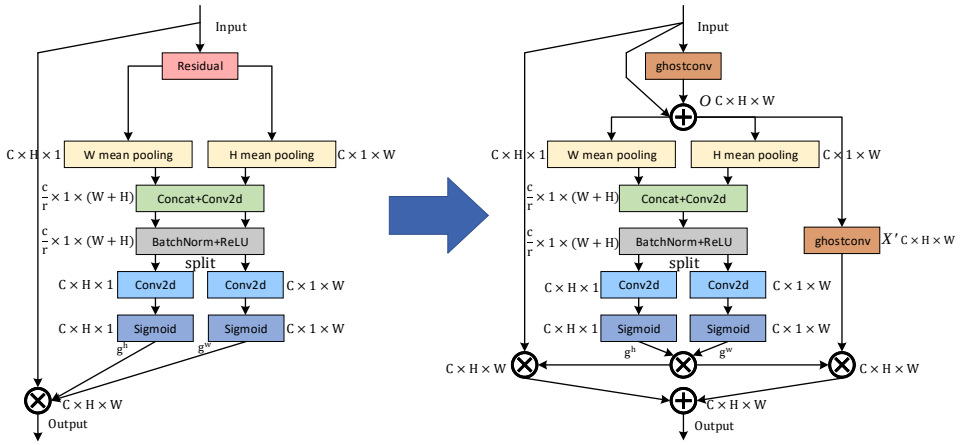
$$g^w = \sigma(F_1(f^w)) \tag{8}$$

Here, g^h and g^w are the attention weights in the H and W directions, respectively. σ is the sigmoid function, and F_1 represents the 1×1 convolution operation. For input O, we establish long-distance connections to provide information flow between different layers and improve the ability of feature information to communicate across layers, yielding

$$X' = \left(F_{ghost}(O)\right) \tag{9}$$

where F_{ghost} represents ghostconv operation. The final coordinate attention block output feature map is computed as:

$$Y_{sa} = X \times g^h \times g^w + X' \times \left(1 - g^h \times g^w\right) \tag{10}$$



(a) coordinate attention block (b) modified coordinate attention block

Figure 6. Structure of coordinate block and modified coordinate block.

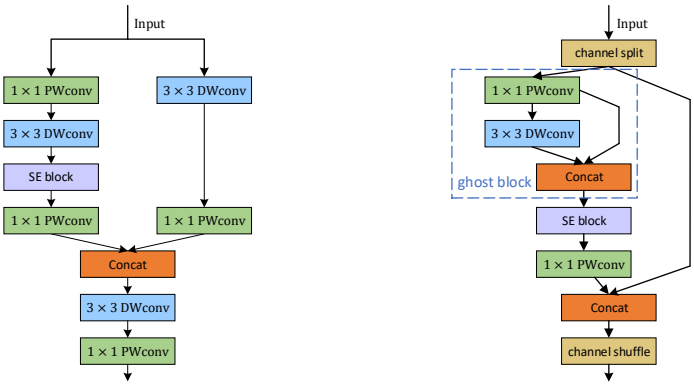
The spatial attention and coordinate attention feature map are concatenated and reduced the dimension of the concatenation result by 1×1 convolution. The output of the SCSA module can be formulated as follows:

$$Y_{scsa} = F_1[Y_{sa} + Y_{ca}] \tag{11}$$

where Y_{sa} and Y_{ca} denote the spatial attention block feature map and the coordinate attention block feature map, respectively. [] is a concatenation operation. The proposed SCSA module integrates spatial information, location information, and channel information, and has better positioning capability for dense aerial imagery, thereby improving the detector accuracy. In this study, the SCSA module is added to ES. Benefiting from the partnership with the SCSA module, the backbone network is still lightweight but more object-focused. It is worth noting that, when the network is deep and the feature maps are small, using the attention mechanism to force the extraction of some features may lose some meaningful feature information. Therefore, as shown in Figure 3, the SCSA module is applied to L3 and L6, but not L9 in the backbone network.

3.1.2. Enhanced ShuffleNet

The original backbone network darknet-53 of YOLOv5-N has 53 layers. The structure of the BottleneckCSP module in the backbone network is computationally complex, and the inference time on hardware devices with limited computational resources cannot meet the application requirements. Therefore, we replace darknet-53 with ES. As shown in Figure 7, ES is divided into two types of channels, with the ES Block_1 outputting the same number of channels and the ES Block_2 increasing the number of channels. Due to the addition of the channel split module at the beginning of the ES Block_1, the input feature map is split into two branches, each with half the number of channels. Compared with the standard convolution, ghost block has smaller parameters and less computation and generates more feature maps, thereby reducing the weight parameters. The SE block is good at weighing the network channels and obtaining better features. After convolution, the two branches are concatenated. When the stride is 2, the input feature map is sent to the two branches for convolution, and the feature map becomes half the size of the initial input feature map, which is output after concatenation and convolution. The final output feature map is twice as many channels as the initial input feature map.



(a) ES Block_1 (b) ES Block_2

Figure 7. The structure of ES Block_1 and ES Block_2.

3.2. Slim-BiFPN

The original YOLOv5 adopts the FPN structure on the neck for feature extraction and fusion. Small object features will lose some information after the low-pass filtering effect of the multiple convolution operation. Moreover, the original convolutional block parameters are redundant, which is not conducive to model lightweight. The proposed lighter Slim-BiFPN replaces the FPN, and the DSC layer is used to replace the original YOLOv5-N convolutional block, reducing the number of convolutional kernels and shrinking the size of the convolutional kernels, resulting in a significant reduction in the number of parameters. Simultaneously, the SCSA is introduced into the convolutional block to adjust the feature map weights during upsampling and reduce the feature loss. Specifically, we construct new modules, called DSSconv and DSSCSP. Therefore, we propose Slim-BiFPN based on BIFPN and construct new SCSAconv and SCSACSP modules, The former reduces the computational cost and parameters, and speeds up inference by replacing the original convolutional block with depthwise convolution (DWconv) and pointwise convolution (PWconv) parts, while the latter uses stacked SCSA to enhance multi-scale feature fusion to reduce non-gradient feature loss to ensure that sufficient features are extracted.

3.2.1. DSSconv

For standard convolution, assuming that the input feature map $X \in R^{C\times H\times W}$, C denotes the number of channels of the input feature map, and H and W denote the height and width of the input feature map, respectively. The output feature map is computed as:

$$Y = F_k(X) \tag{12}$$

where $Y \in R^{C'\times H'\times W'}$ denotes the output feature map, and F_k represents the $k \times k$ convolution operation. C' denotes the number of channels of the output feature map, and H' and W' denote the width and height of the output feature map, respectively. Therefore, the parameters required for the standard convolution are calculated as follows:

$$P_{standardconvolution} = k \times k \times C \times C' \tag{13}$$

Compared with standard convolution, the DSC can effectively reduce the model parameters and computational effort, and the difference in results from standard convolution is minimal. The DSC module is divided into two processes: channel-by-channel two-dimensional depthwise convolution DWconv in the depth direction, then followed by a 1×1 pointwise convolution to compress and expand the channels to achieve the effect of standard convolution. Specifically, in the DWconv operation, the channels and convolution kernels correspond one-to-one, meaning that a convolution kernel has only one channel, the number of convolution kernels is the same as the number of the previous layer channels, the convolution kernel shape is $k \times k \times 1$, and the feature map shape changes

from $C \times H \times W$ to $C' \times H' \times W'$. Therefore, the parameters required for the DWconv module are calculated as follows:

$$P_{DWconv} = k \times k \times 1 \times C \tag{14}$$

Then, the PWconv operation is employed in the feature map after the DWconv module. The number of convolution kernels is the same as the number of output channels, and the convolution kernel shape is $1 \times 1 \times C$. The feature map shape changes from $C \times H' \times W'$ to $C' \times H' \times W'$. Therefore, the parameters required for the PWconv module are calculated as follows:

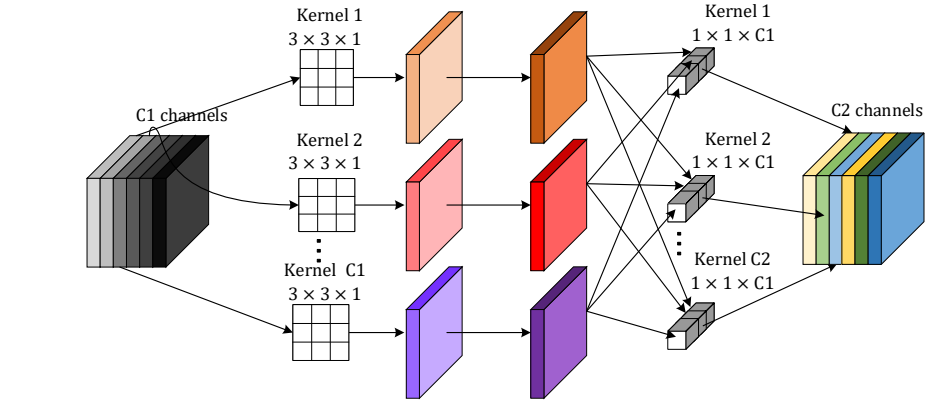
$$P_{PWconv} = 1 \times 1 \times C \times C' \tag{15}$$

The ratio of parameters required for the DSC module relative to the standard convolution module can be calculated as follows:

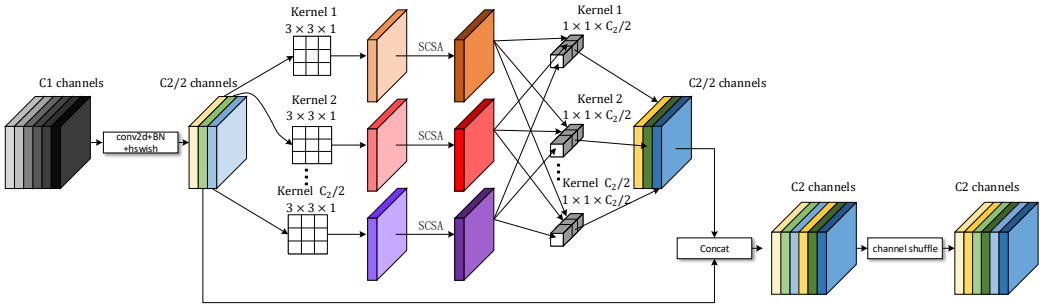
$$r_c = \frac{k \times k \times 1 \times C + 1 \times 1 \times C \times C'}{k \times k \times C \times C'} = \frac{1}{C'} + \frac{1}{k^2} \tag{16}$$

It can be seen that the number of DSC module parameters is $\frac{1}{C'} + \frac{1}{k^2}$ times that of standard convolution from the results. In this paper, $k = 3$, so the number of Slim-BiFPN parameters can be reduced by replacing standard convolution with DSC.

However, the channel information of the feature map is separated during the DSC module, which reduces the feature extraction ability of the network. Therefore, the channel shuffle operation is adopted to communicate information between the two branches. Inspired by Mobilenetv3 [36], we introduce the SCSA module in the DSC to adjust the weight of each channel, aiming to improve the balanced feature map. We refer to this modified convolutional module as DSSconv, and Figure 8 shows the structure of the DSSconv module. Specifically, the input features are reduced to half of the final output channel number by the CONV module (conv2d+ BN+Hard Swish activation function); then, a DWconv operation with a 3×3 convolution kernel is applied. Next, after the features are adjusted by the SCSA module with the weights of each channel, the new features are then generated by the 1×1 PWconv module. Finally, the feature maps are concatenated with the initial CONV feature map, and a channel shuffle module is used to exchange feature information on different channels.



(a) the DSC module



(b) the DSSconv module

Figure 8. The structure of DSC and DSSconv modules; (**a**) the DSC module; (**b**) the DSSconv module.

3.2.2. DSSCSP

In the original FPN structure, the BottleneckCSP module is used to learn the feature map, but too many convolutions in the BottleneckCSP may lead to a loss of features and redundant parameters. Therefore, we propose DSSCSP instead of BottleneckCSP. Figure 3 shows the structure of the DSSCSP. Specifically, the initial feature map of DSSCSP is input to two branches, the first branch learns features via CONV and stacked SCSA, and the second branch directly learns features via CONV module. Finally, the output feature maps from the two branches are concatenated together and output through a single CONV module. We do not use DSSconv in the DSSCSP, although this will reduce the model parameters, but will significantly deepen the network layer of Slim-BIFPN and significantly increase the inference time.

3.3. Optimization of Loss Function

The loss function of YOLOv5 mainly consists of confidence loss, classification loss, and bounding box loss. In the original YOLOv5, regression prediction of the bounding box by using the CIoU function is defined as follows:

$$\begin{aligned} L_{CIoU} &= 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \beta v, \\ IoU &= \frac{\left|B \cap B^{gt}\right|}{\left|B \cup B^{gt}\right|}. \end{aligned} \tag{17}$$

where b and b^{gt} denote the center points of the predicted box and ground truth box, respectively. ρ represents the Euclidean distance between b and b^{gt}, and c represents the diagonal length of the smallest rectangle covering b and b^{gt}. β denotes the trade-off parameter, and v is used to measure the consistency of the aspect ratio between b and b^{gt}, which are defined as follows:

$$\begin{aligned} \beta &= \frac{v}{(1 - IoU) + v}, \\ v &= \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2. \end{aligned} \tag{18}$$

where w^{gt} and h^{gt} represent the width and height of the ground truth box, respectively. w and h represent the width and height of the predicted box, respectively. $\alpha-$IoU [37] introduces parameter α into the original IoU parameters to adjust the performance. The $\alpha-$CIoU is defined as follows:

$$L_{\alpha-CIoU} = 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}\left(b, b^{gt}\right)}{c^{2\alpha}} + (\beta v)^{\alpha} \tag{19}$$

There are numerous small objects in the aerial images. The smaller the object, the closer the predicted box's center point is to the ground truth box, and thus the loss will be significantly lower than that of the large object when calculating the loss, which is not conducive to the calculation of the loss of small objects. In this paper, by increasing the Euclidean distance between the predicted box and ground truth box, amplifying the loss weight of large and medium objects, and expanding the difference, optimize the loss calculation for small objects. The α is the parameter that boosts the loss value and gradient of the high IoU object, thus improving the regression effect of the box and accelerating convergence. Through experimentation, we found that the best results are obtained when α is 3, so in our work, α = 3. Finally, the IoU loss function in this paper is defined as follows:

$$L_{IoU} = 1 - IoU^{\alpha} + \frac{\left(\rho^{2\alpha}\left(b, b^{gt}\right)\right)^{2}}{c^{2\alpha}} + (\beta v)^{\alpha} \tag{20}$$

4. Experiment Results and Discussion

4.1. Dataset

All the experiments are conducted on the public object detection benchmark VisDrone-DET2021. The dataset is mainly marked with ground humans and daily vehicles, which are captured by UAV at different shooting angles and heights, consists of 8899 images and 382,005 labels, of which 6471 images are used for training and 548 images for validation. There are 10 categories in the dataset, namely pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle.

4.2. Evaluation Criteria

Mean Average Precision (mAP) is used to measure accuracy, which is the average of the various types of accuracy in the dataset, by plotting a curve against accuracy and recall, and the area of the curve against the coordinate axis is the AP value. The Precision, Recall, AP, and mAP are calculated respectively as follows:

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$$AP = \int_{0}^{1} P(r)dr \tag{23}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{24}$$

where True Positive (TP) is the number of positive samples being correctly classified, False Positive (FP) is the number of negative samples being incorrectly classified, and False Negative (FN) is the number of positive samples being incorrectly classified. n indicates the total number of detection categories.

In this study, we follow the evaluation protocol of MS COCO dataset, including mAP (IoU threshold average), mAP0.50 (IoU threshold = 0.50 mAP), and mAP0.75 (IoU threshold = 0.75 mAP). Three criteria are also included, namely AP-s, AP-m, and AP-l, corresponding to small, medium, and large scale AP results, respectively. The higher the mAP and the AP value, the better the detection effect of the object detector. The size of the

model is evaluated in terms of the number of parameters, and the fewer the parameters, the lighter the detector. Frames per second (fps) is used for the real-time efficiency evaluation, to ensure similarity to real-world usage scenarios, and our calculation section includes post-processing, such as input/output and non-maximum suppression. The larger the fps, the better the real-time efficiency of the detector.

4.3. Implementation Details

Our YOLO-UAVlite detector is based on the deep learning framework of PyTorch, and the experimental code is based on the improvement of the YOLOv5 project version 6.1 of ultralytics. We choose adaptive moment estimation (Adam) as the optimizer. The batch size, momentum, and weight decay are set to 16, 0.937, and 0.0005, respectively. To avoid the loss value being too large to affect the stability of convergence at the beginning of training, we first run three epochs of warm-up training. During warm-up, the momentum of the optimizer is set to 0.8. After warm-up training, the learning rate is decayed adopting a cosine annealing function, where the initial learning rate and the minimum learning rate are set to 0.001 and 0.00001, respectively. Finally, we train the model for 300 epochs. When loading the data, all image resolution is uniformly adjusted to 640 × 640, and data preprocessing adopts mosaic data augmentation. All experimental algorithms in this paper are migrated implementations of the official model, trained from scratch for 300 epochs starting from batch size 16, with the remaining hyperparameters as the default.

The operating environment for all experimental algorithms is as follows: the CPU is Intel i9-10900K, the GPU is NVIDIA GeForce RTX 3090 (24GB), and the Operating System is Windows 64-bit. The frameworks are Python 3.7.0, PyTorch 1.11.0, and CUDA 11.3.

4.4. Results and Discussion

4.4.1. Ablation Experiments

To verify the effectiveness of the three improved methods, we conducted ablation experiments. As shown in Table 2 and Figure 9, the results of the ablation experiments show that each method achieves some improvement. Based on the baseline model YOLOv5-N, we propose SCSAshufflenet as the backbone, replacing the original FPN with Slim-BiFPN, and modifying the loss function. The detection results are shown in Figure 10, where it can be seen that each of the improvement methods improves the detection accuracy. The ablation experiments were conducted with the same training configuration for a fair comparison.

Table 2. The experimental results of ablation of the algorithm module.

Method	SCSAshufflenet	Slim-BiFPN	Loss Function	mAP0.50	mAP	mAP0.75	AP-s	AP-m	AP-l	Params(m)	FPS
YOLOv5-N				0.257	0.130	0.116	0.068	0.200	0.272	1.77	72.0
M1	√			0.359	0.188	0.175	0.117	0.278	0.319	1.66	49.7
M2		√		0.261	0.135	0.123	0.071	0.201	0.272	**1.28**	**89.3**
M3			√	0.264	0.139	0.136	0.077	0.222	0.285	1.28	71.6
M4	√	√		0.364	0.194	0.181	0.125	0.278	0.317	1.41	63.5
M5	√	√	√	**0.366**	**0.203**	**0.197**	**0.129**	**0.293**	**0.334**	1.41	63.3

Bolded text is the best value for the column.

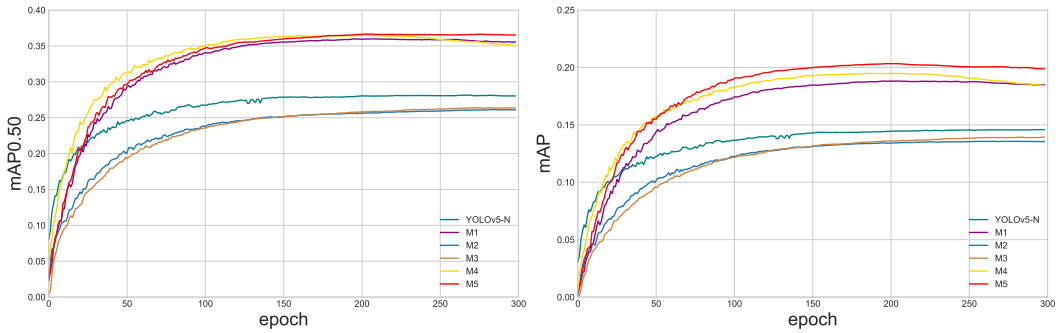


Figure 9. Comparison of experimental mAP0.5 and mAP for ablation. M1: SCSAshufflenet backbone network; M2: Slim-BiFPN; M3: Loss Function; M4: SCSAshufflenet + Slim-BiFPN; M5: SCSAshufflenet + Slim-BiFPN + Loss Function.

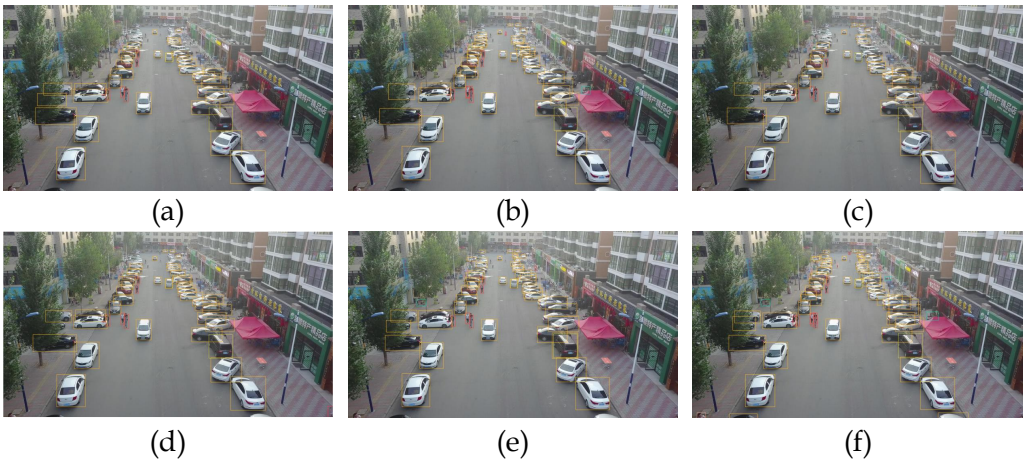


Figure 10. Detection examples of different ablation experiments in aerial image scenes. (**a**) YOLOv5-N; (**b**) M1: SCSAshufflenet; (**c**) M2: Slim-BiFPN; (**d**) M3: Loss function; (**e**) M4: SCSAshufflenet + Slim-BiFPN; (**f**) M5: SCSAshufflenet + Slim-BiFPN + Loss Function.

To verify the feasibility of the SCSAshufflenet backbone, YOLOv5-N was selected as a baseline architecture, and the original backbone darknet-53 is replaced with SCSAshufflenet. As can be seen from rows 1 and 2 of Table 2, compared with the YOLOv5-N, the model parameters are reduced by 6.2% after setting SCSAshufflenet to the backbone, while the mAP0.50 is increased by 10.2 percentage points. The AP results at small, medium, and large scales achieve gains, and the smaller the object, the more significant the improvement. Considering the distribution of object sizes in aerial images where small scales dominate, the proposed SCSAshufflenet strategy achieves the desired effect.

To demonstrate the role of Slim-FPN, we replace the original the FPN with Slim-FPN and leave the rest of the model unchanged. As can be seen from rows 1 and 2 of Table 2, after replacing FPN with Slim-BiFPN, the model parametric count drops from 1.77 m to 1.28 m, reducing the number of parameters by 27.6%. At the same time, the fps is increased by 17.3, indicating that the real-time performance of the model has also been improved. In addition, the accuracy of the model with various types of AP values is slightly improved. The mAP0.50 improved from 25.7% to 26.1%, and the AP-s and the AP-m improved by 0.3% and 0.1%, respectively. As described in Section 3.2, Slim-BiFPN uses a lighter DSSconv with fewer model parameters, and DSSCSP reduces feature loss, especially for small object features, which explains why the highest improvement in AP values is observed for small objects.

Modifying the loss function usually only affects the training process, but does not or rarely does not affect the inference time of the network. In this paper, we use the new loss function to obtain a higher-quality box. Rows 1 and 4 of Table 2 showed that, after using

the new loss function, we were able to achieve gains of 0.7%, 0.9%, and 2.0% in mAP0.50, mAP, and mAP0.75, respectively. The fps has changed slightly from 72.0 to 71.6.

To verify the interplay of various aspects of improvement, we replaced both SCSAshufflenet and Slim-BiFPN into YOLOv5-N, and as can be seen in rows 1 and 5 of Table 2, both AP values improved and all model parameters decreased compared with the original YOLOv5-N. Interestingly, by comparing rows 2, 3, and 5, we find that using SCSAshufflenet or Slim-BiFPN alone does not balance accuracy and speed. When using SCSAshufflenet alone, the model parameter count remains at a high level, and the fps decreases significantly, although the mAP0.50 increases from 25.7% to 35.9%. Similarly, when using Slim-BiFPN alone, the model lightness is improved, but the mAP0.50 and other AP values are not significantly increased. This confirms what was mentioned in Section 3—that the model accuracy improvement is mainly due to the SCSAshufflenet backbone, while the lightness is mainly due to Slim-BiFPN.

Finally, all the improvements are applied to verify the enhancement of each part compared to the benchmark model YOLOv5-N in all aspects and to verify the balance between accuracy and lightness. As can be seen from rows 1 and 6 of Table 2, the simultaneous use of the three improvements results in greater enhancements in both accuracy and lightness than the use of one of the improvements alone, achieving a balance between accuracy and speed trade-offs and achieving higher accuracy on UAV platforms with limited computational resources.

4.4.2. Comparisons with State-of-the-Art in Aerial Images Detection

To further validate the performance of the proposed YOLO-UAVlite in aerial images, the YOLO-UAVlite is compared with other lightweight detectors on the VisDrone-DET2021 dataset. The compared detectors are the anchors-free detector NanoDet-M-1.5X, PP-PicoDet-M [30], YOLOv6-N [38] and YOLOv7-Tiny [39], and the anchors-based detectors are YOLOv3-Tiny [10], YOLOv4-Tiny [13], YOLOv5-N, YOLOv5-S, and YOLOv5-Lites-S [40]. We did not choose a two-stage detector due to its large parameters and relatively slow detection speed compared to lightweight detector, which cannot be deployed to UAV. Therefore, there is no comparison between the two-stage detector and the lightweight detector in the aerial image scenario. To effectively compare the performance of the proposed algorithms, the training environments and datasets are exactly the same for all algorithms. The comparison results between YOLO-UAVlite and the other algorithms are shown in Figure 11. Table 3 records the performance comparison of YOLO-UAVlite with other state-of-the-art methods.

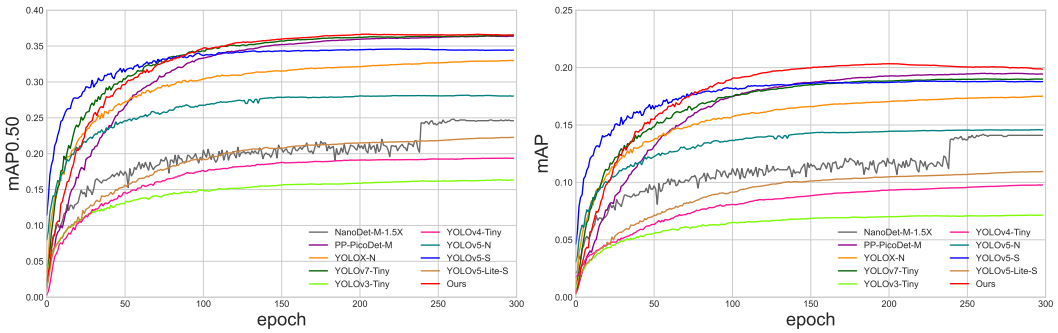


Figure 11. Comparison of mAP0.5 and mAP of different lightweight detectors.

Table 3. Comparison with state-of-the-art lightweight detectors.

Method	mAP0.50	mAP	mAP0.75	AP-s	AP-m	AP-l	Params(m)	FPS
anchors-free								
NanoDet-M-1.5X (2020) [18]	0.248	0.142	0.142	0.064	0.230	0.316	2.08	72.0
PP-PicoDet-M (2021) [30]	0.363	0.195	0.183	0.123	0.284	0.314	2.15	67.0
YOLOv6-N (2022) [38]	0.290	0.152	0.142	0.084	0.233	0.270	4.30	**127.3**
YOLOv7-Tiny (2022) [39]	0.364	0.190	0.173	0.114	0.278	0.357	6.19	82.0
anchors-based								
YOLOv3-Tiny (2018) [10]	0.149	0.063	0.049	0.032	0.103	0.149	6.06	103.7
YOLOv4-Tiny (2020) [13]	0.194	0.098	0.077	0.045	0.133	0.187	8.86	78.0
YOLOv5-N (2020) [15]	0.257	0.130	0.119	0.068	0.200	0.279	1.90	72.0
YOLOv5-S (2020) [15]	0.329	0.176	0.164	0.103	0.259	**0.368**	7.20	53.1
YOLOv5-Lite-S (2021) [40]	0.195	0.092	0.079	0.049	0.145	0.208	1.64	57.1
YOLO-UAVlite (ours)	**0.366**	**0.206**	**0.197**	**0.129**	**0.293**	0.334	**1.41**	63.3

Bolded text is the best value for the column.

For the VisDrone-DET2021 dataset, our YOLO-UAVlite achieves 36.6% mAP0.50 with 1.41m model parameters. As shown in Figure 12, the proposed method outperforms all the compared models, achieving state-of-the-art detection accuracy with the least amount of model parameters. From these two indicators, it can be seen that YOLO-UAVlite is easier to deploy on UAV platforms and achieves higher accuracy and better detection results. Compared with the baseline architecture YOLOv5-N, the proposed model reduces the number of parameters by 25.8%, while the mAP0.50 is increased by 10.9%; in particular, the detection effect of small objects is considerably increased, and AP-s is 1.9 times larger than YOLOv5-N. Compared with the larger YOLOv5-S, the number of parameters is only one-fifth that of YOLOv5-S, and the mAP0.50 is 3.7% higher. Compared with YOLOv3-Tiny, YOLOv4-Tiny, and YOLOv5 Lite-S, the proposed model achieves the highest performance scores in terms of mAP0.50, mAP, mAP0.75, and small, medium, and large scale AP values, while minimizing the number of model parameters. Compared with the anchor-free detectors NanoDet-M-1.5X and YOLOv6-N, the mAP0.50 shows an increase of 11.8% and a decrease of 7.6%, respectively, and the parameters are greatly reduced by 32.2% and 67.2%, respectively. YOLO-UAVlite has slightly higher mAP0.50 than PP-PicoDet-M and YOLOv7-Tiny, but the number of model parameters of YOLO-UAVlite is 65.6% of PP-PicoDet-M and 22.8% of YOLOv7-Tiny, respectively.

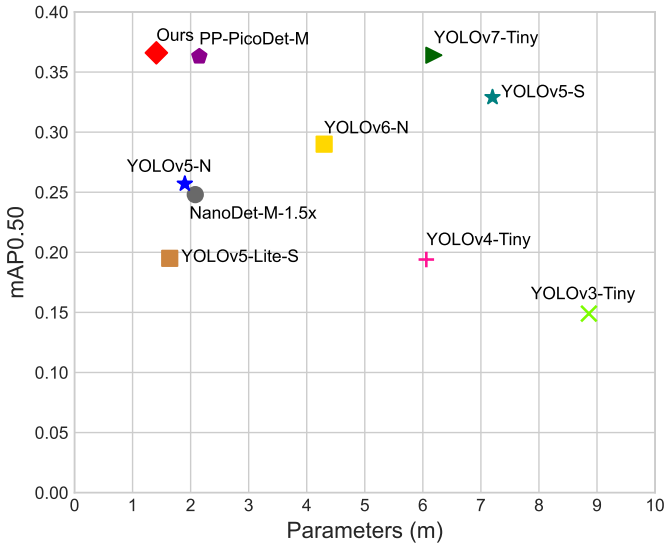


Figure 12. Comparison of mAP0.50 and parameters of different lightweight detectors.

By comparing our model with the AP-s, AP-m, and AP-l for each model, we found that the Ap-s lead is the highest, indicating that our model gains are mainly from small-scale objectives. Although the AP-l of our method is lower than that of YOLOv5-S and YOLOv7-Tiny, the recognition accuracies for small-scale objects and medium-scale objects are 12.9% and 29.3%, respectively, which are significantly higher than other methods, proving that our YOLO-UAVlite detector has advantages in aerial image scenes with a large number of small objects. In addition, we also investigate the computational cost of the proposed and compared methods. Compared with the baseline YOLOv5-N, the model inference speed is decreased by 8.7 fps, but the accuracy of each scale is considerably increased. The fastest detector is YOLOv6-N, which reaches 127.3 fps. Although the proposed YOLO-UAVlite model drops to 63.3 fps, it still meets the speed requirements on the UAV platforms. This shows that our proposed YOLO-UAVlite has elevated accuracy for multi-scale object detection and achieves better accuracy and performance trade-off.

To adequately illustrate the applicability of our model to different image scenes, we present object detection results under typical scene conditions. The test results are shown in the figure. As shown in Figure 13, the overall detection rate of YOLO-UAVlite is higher than that of the other detectors, especially in the long-distence view, where it can still correctly detect more objects. As can be seen from Figure 14, our YOLO-UAVlite can detect the actual objects more accurately in the object cluster with high overlap, with fewer overlapping boxes and lower false alarms than the comparison detectors.

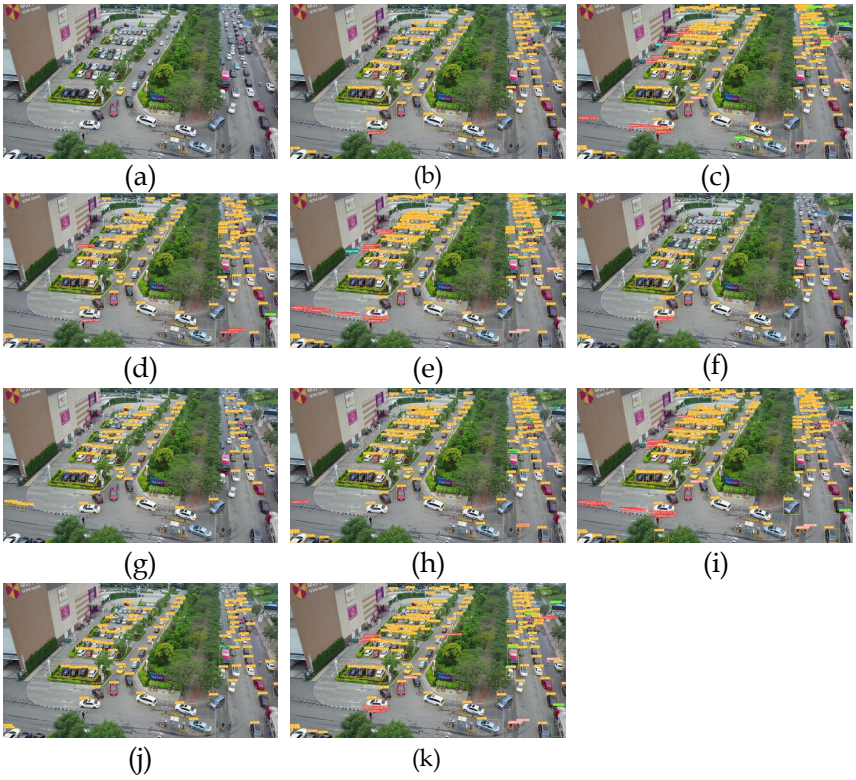


Figure 13. Detection examples of different algorithms in aerial image scenes. (**a**) original image; (**b**) NanoDet-M-1.5X; (**c**) PP-PicoDet-M; (**d**) YOLOv6-N; (**e**) YOLOv7-Tiny; (**f**) YOLOv3-Tiny; (**g**) YOLOv4-Tiny; (**h**) YOLOv5-N; (**i**) YOLOv5-S; (**j**) YOLOv5-Lite-S; (**k**) ours.

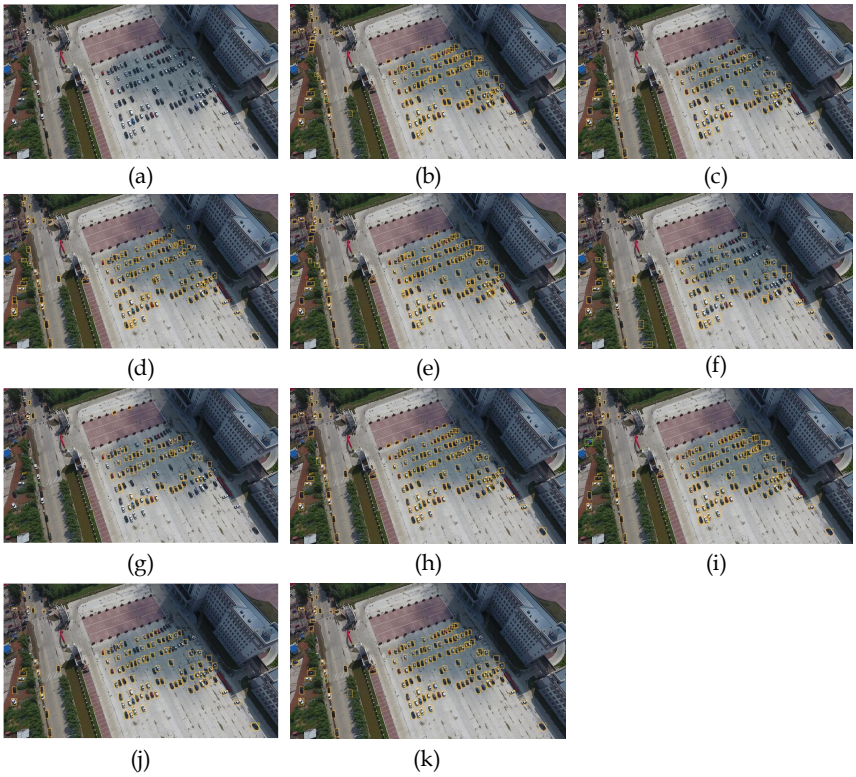


Figure 14. Detection examples of different algorithms in dense small object scenes. (**a**) Original image; (**b**) NanoDet-M-1.5X; (**c**) PP-PicoDet-M; (**d**) YOLOv6-N; (**e**) YOLOv7-Tiny; (**f**) YOLOv3-Tiny; (**g**) YOLOv4-Tiny; (**h**) YOLOv5-N; (**i**) YOLOv5-S; (**j**) YOLOv5-Lite-S; (**k**) ours.

5. Conclusions

In this study, a lightweight detector YOLO-UAVlite is proposed to tackle the challenge of small-scale object detection in aerial images. The network structure is improved based on the YOLOv5-N algorithm to improve the detection effect of small objects. We modify the spatial and coordinate attention and combine their advantages to generate a new attention, called SCSA, which integrates spatial, location, and channel information to enhance object representation. We propose a backbone network based on the SACA and ES. The improved backbone greatly increases the detection performance, especially for small objects. The proposed Slim-BiFPN considerably lightens the network, and the SCSA module is fused to reduce information loss and avoid accuracy loss. Finally, we expand the difference and reduce the loss weight for the small objects by the optimizing loss function. Our experiments show that YOLO-UAVlite increases mAP0.50 by 10.9% and reduces parameters by 25.8% compared with the original detector on the VisDrone-DET2021 dataset. The proposed model has parameters of 1.41 m, mAP0.50 of 36.6%, and the fps up to 63.3, which essentially meets the performance and accuracy requirements of UAV platforms. In the future, our work continues to be focus on the detection of dense small objects and practical test deployment work on UAV platforms.

Author Contributions: Conceptualization, C.L. and D.Y.; data curation, C.L. and L.T.; methodology, C.L., D.Y., and L.T.; software, C.L. and X.Z.; formal analysis, C.L. and Y.D.; validation, C.L., D.Y., and X.Z.; visualization, L.T., Y.D., and C.L.; writing—original draft preparation, C.L. and D.Y.; writing—review and editing, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: The work described in this paper was partially supported by the National Natural Science Foundation of China under Grant No. 61772295; the Natural Science Foundation Project of CQ CSTC under Grant No. cstc2018jcyjAX0470; the Applying Basic Research Program of Chongqing Education Committee under Grant No. KJZD-M202000501; the Chongqing Postgraduate Education Teaching Reform Research Project under Grant No. yjg193096; and the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant No. KJQN201800539.

Data Availability Statement: The data used to support the findings of this study are obtained from [20].

Acknowledgments: The authors thank the anonymous and editor reviewers for their critical comments and suggestions for improving the manuscript.

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

References

1. Wang, L.; Xiang, L.; Tang, L.; Jiang, H. A Convolutional Neural Network-Based Method for Corn Stand Counting in the Field. *Sensors* **2021**, *21*, 507.
2. Sambolek, S.; Ivasic-Kos, M. Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors. *IEEE Access* **2021**, *9*, 37905–37922.
3. Yifan.; Lu.; Jiaming.; Lu.; Songhai.; Zhang.; Peter.; Hall. Traffic signal detection and classification in street views using an attention model. *Comput. Vis. Media* **2018**, *4*, 253–266.
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014
5. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
6. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the International Conference on Computer Vision, Venice, Italy, 22–29 October
2017.
7. 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.
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
9. 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.

10. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
11. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
12. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning. PMLR, Lille, France, 7–9 July 2015; pp. 448–456.
13. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
14. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34; pp. 12993–13000.
15. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012.; Kwon, Y.; TaoXie.; Fang, J.; imyhxy.; Michael, K.; et al. ultralytics/yolov5: v6.1—TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference, 2022. Available online: https://zenodo.org/record/6222936#.Y5GBLH1BxPZ (accessed on 20 December 2022). https://doi.org/10.5281/zenodo.6222936.
16. 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.
17. Wang, W.; Xie, E.; Song, X.; Zang, Y.; Wang, W.; Lu, T.; Yu, G.; Shen, C. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 8440–8449.
18. RangiLyu. NanoDet-Plus: Super Fast and High Accuracy lightweight Anchor-Free Object Detection Model. 2021. Available online: https://github.com/RangiLyu/nanodet (accessed on 20 December 2022).
19. 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.
20. 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. https://doi.org/10.1109/TPAMI.2021.3119563.
21. Yang, F.; Fan, H.; Chu, P.; Blasch, E.; Ling, H. Clustered object detection in aerial images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 8311–8320.
22. Wu, X.; Hong, D.; Tian, J.; Chanussot, J.; Li, W.; Tao, R. ORSIm detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 5146–5158.
23. Pang, J.; Li, C.; Shi, J.; Xu, Z.; Feng, H. $mathcal R^2$ -CNN: Fast Tiny Object Detection in Large-Scale Remote Sensing Images. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 5512–5524.
24. Cui, L.; Lv, P.; Jiang, X.; Gao, Z.; Zhou, B.; Zhang, L.; Shao, L.; Xu, M. Context-aware block net for small object detection. *IEEE Trans. Cybern.* **2020**, *52*, 2300–2313.
25. Li, Y.; Li, S.; Du, H.; Chen, L.; Li, Y. YOLO-ACN: Focusing on small target and occluded object detection. *IEEE Access* **2020**, *8*, 227288–227303.
26. Xu, Y.; Fu, M.; Wang, Q.; Wang, Y.; Chen, K.; Xia, G.S.; Bai, X. Gliding vertex on the horizontal bounding box for multi-oriented object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 1452–1459.
27. 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.
28. 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.
29. 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.
30. Yu, G.; Chang, Q.; Lv, W.; Xu, C.; Cui, C.; Ji, W.; Dang, Q.; Deng, K.; Wang, G.; Du, Y.; et al. PP-PicoDet: A Better Real-Time Object Detector on Mobile Devices. *arXiv* **2021**, arXiv:2111.00902.
31. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
32. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks *arXiv* **2019**, arXiv:1910.03151.
33. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
34. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
35. 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.
36. 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, Korea, 27 October–2 November 2019 ; pp. 1314–1324.
37. He, J.; Erfani, S.; Ma, X.; Bailey, J.; Chi, Y.; Hua, X.S. Alpha-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20230–20242.

38. 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.
39. 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 preprint arXiv:2207.02696* **2022**.
40. Chen, X.; Gong, Z. YOLOv5-Lite: Lighter, faster and easier to deploy. https://doi.org/10.5281/zenodo.5241425.