remote sensing

MDPI

Article

Target Detection Method of UAV Aerial Imagery Based on Improved YOLOv5

Xudong Luo [1], Yiquan Wu [1,*] and Feiyue Wang [2]

[1] College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[2] Wuxi Gewu Intelligent Technology Co., Ltd., Wuxi 214016, China
* Correspondence: imagestrong@nuaa.edu.cn; Tel.: +86-137-7666-7415

Abstract: Due to the advantages of small size, lightweight, and simple operation, the unmanned aerial vehicle (UAV) has been widely used, and it is also becoming increasingly convenient to capture high-resolution aerial images in a variety of environments. Existing target-detection methods for UAV aerial images lack outstanding performance in the face of challenges such as small targets, dense arrangement, sparse distribution, and a complex background. In response to the above problems, some improvements on the basis of YOLOv5l have been made by us. Specifically, three feature-extraction modules are proposed, using asymmetric convolutions. They are named the Asymmetric ResNet (ASResNet) module, Asymmetric Enhanced Feature Extraction (AEFE) module, and Asymmetric Res2Net (ASRes2Net) module, respectively. According to the respective characteristics of the above three modules, the residual blocks in different positions in the backbone of YOLOv5 were replaced accordingly. An Improved Efficient Channel Attention (IECA) module was added after Focus, and Group Spatial Pyramid Pooling (GSPP) was used to replace the Spatial Pyramid Pooling (SPP) module. In addition, the K-Means++ algorithm was used to obtain more accurate anchor boxes, and the new EIOU-NMS method was used to improve the postprocessing ability of the model. Finally, ablation experiments, comparative experiments, and visualization of results were performed on five datasets, namely CIFAR-10, PASCAL VOC, VEDAI, VisDrone 2019, and Forklift. The effectiveness of the improved strategies and the superiority of the proposed method (YOLO-UAV) were verified. Compared with YOLOv5l, the backbone of the proposed method increased the top-one accuracy of the classification task by 7.20% on the CIFAR-10 dataset. The mean average precision (mAP) of the proposed method on the four object-detection datasets was improved by 5.39%, 5.79%, 4.46%, and 8.90%, respectively.

Keywords: unmanned aerial vehicle (UAV) aerial imagery; you only look once (YOLO); asymmetric convolutions; attention mechanism; K-Means++; non-maximum suppression (NMS)

Citation: Luo, X.; Wu, Y.; Wang, F. Target Detection Method of UAV Aerial Imagery Based on Improved YOLOv5. *Remote Sens.* **2022**, *14*, 5063. https://doi.org/10.3390/rs14195063

Academic Editor: Eufemia Tarantino

Received: 5 September 2022
Accepted: 6 October 2022
Published: 10 October 2022

1. Introduction

An unmanned aerial vehicle (UAV) has excellent convenience, stability, and safety. Because of its easy operation, flexible takeoff and landing, and wide detection range, it is frequently used in forestry and crop monitoring [1–4], traffic supervision [5], urban planning [6], municipal management [7,8], transmission line inspection [9,10], search and rescue [11–13], and other fields. In forestry and crop monitoring, the acquisition and analysis of farmland data is very important, as it helps growers to carry out efficient management. Examples include the precise spraying of pesticides, monitoring of tree growth, and timely harvesting of crops. Traditional methods that rely on the manual acquisition of data are time-consuming and labor-intensive, and they are prone to inaccurate data due to sampling bias and sparse measurement. A common alternative today is to use UAVs to capture the aerial imagery of farmland and then analyze the imagery to obtain the required information. Since the size of the objects in the image varies with the altitude of the

UAV, the objects appear in the image at different scales. Learning efficient representations for multiscale objects is an important challenge for object detection in UAV aerial images. Because of the rapid advancement of UAV technology, onboard cameras have become increasingly capable of capturing stable, high-resolution aerial images. This helps UAVs perform search-and-rescue missions over wide search areas or areas affected by natural disasters. Analyzing a large number of acquired aerial images in a short period of time is a huge and challenging job that would be quite stressful if performed manually. Therefore, an accurate and real-time UAV target detection method is urgently needed.

Traditional detection methods [14–18] traverse each image through a preset sliding window to extract features and then use a trained classifier for classification. They tend to require a lot of manpower and effort to process data, and it is difficult to uniformly set standards for features. In addition, traditional detection methods often face problems such as high time complexity, poor robustness, and strong scene dependence, which make them difficult to put into practical use. In recent years, with the continuously proposed target-detection methods based on Convolutional Neural Networks (CNNs), excellent detection results have been achieved. It has been demonstrated that these deep-learning-based algorithms are better suited for machine vision tasks. Depending on how the input image is processed, there are two types of object-detection methods: two-stage and one-stage detection. Their respective advantages can be summarized as good detection accuracy and calculation speed. Among them, R-CNN [19], Fast R-CNN [20], Faster R-CNN [21], Mask R-CNN [22], Cascade R-CNN [23], R-FCN [24], etc., are two-stage detection methods. DenseBox [25], RetinaNet [26], SSD series [27], YOLO series [28–33], etc., are one-stage detection methods.

During the flight of the UAV, the mounted device will transmit the captured images in real time, and this poses a challenge to the speed of the detection method. In addition, the objects contained in the images are mainly small objects, which are characterized by occlusion, blurring, dense arrangement, and sparse distribution, and they are often submerged in complex backgrounds. Due to the aforementioned problems, it is difficult for current detection methods to accurately locate and detect targets on UAV aerial images. They still have a lot of room for improvement. In recent years, YOLO series detection methods have been widely used in the detection of targets in UAV aerial images due to their superior speed and good accuracy. Chuanyang Liu et al. [10] proposed MTI-YOLO for targets such as insulators in power line inspection using UAVs. On the basis of YOLOv3-Tiny, MTI-YOLO expands the neck by adding a feature-fusion structure and SPP modules. It also adds the output layers of the backbone. The improvement of this method in the neck is relatively redundant, and the structure of the network needs to be optimized. Oyku Sahin et al. [34] analyzed the challenging problems in UAV aerial images. They extended the output layer of the backbone of YOLOv3 to detect objects of different scales in the image, increasing the original three detection layers to five. Such a structure plays a certain role in the feature-fusion part. However, this leads to overly large and complex detection models, thus increasing the cost of training and computation. Junos et al. [35] produced a UAV aerial imagery dataset targeting oil palm fruit. Based on YOLOv3-Tiny, they proposed a target-detection method for oil palm fruit. The method uses a densely connected neural network and Swish activation functions and adds a new detection layer. The activation function selected by this model is prone to performance degradation in deep networks, and the added feature layer undoubtedly slows down the detection speed of the model. Jia GUO et al. [9] proposed an improved YOLOv4 detection method for small targets such as anti-vibration hammers in transmission lines in UAV aerial images. To improve the ability of the network to extract features, the method adds Receptive Field Block (RFB) modules in the neck. In the proposed method, there is a lack of discussion on the location of adding modules, and the improved strategy is relatively simple. Yanbo Cheng et al. [36] proposed an improved YOLO method for image blur caused by the camera shaking during UAV aerial photography, exposure caused by uneven illumination, and noise during transmission. This method uses a variety of data-enhancement

methods such as affine transformation, Gaussian blur processing, and grayscale transformation to strengthen the data preprocessing capability of the YOLOv4, which is used to alleviate the problem of difficult training due to a small amount of data. The downside is that this method lacks targeted modifications to the structure of the model itself. Based on YOLOv5, Wei Ding et al. [7] added a Convolutional Block Attention Module (CBAM) to distinguish buildings of different heights in UAV aerial images. The backbone of the improved model enhances the feature-extraction capability, but it should be noted that the amount of computation will increase due to the addition of other modules. Xuewen Wang et al. [37] proposed the LDS-YOLO detection method in view of the characteristics of small targets and insignificant details of dead trees in UAV aerial images. This method is improved on the basis of YOLOv5. A new feature-extraction module is constructed; the SoftPool method is introduced in the SPP module; and the traditional convolutions are replaced with depth-wise separable convolutions. This method gives a good performance. Although the depth-wise separable convolution used reduces the parameters of the model, it is easier to fail to learn the target features due to insufficient samples during training. To address the problem of the poor detection performance of damaged roads in UAV aerial images, YuChen Liu et al. [6] presented the M-YOLO detection method. This method replaces the backbone of YOLOv5s with MobileNetv3 and introduces the SPPNet network structure, which is beneficial to improving the detection speed of the model. It should be noted that the increase in speed is often accompanied by a sacrifice in detection accuracy. Based on YOLOv5s, Rui Zhang et al. [8] proposed a defect detection method for wind turbine blades in UAV aerial images, named SOD-YOLO. SOD-YOLO adds a small object detection layer, uses the K-Means algorithm to cluster to obtain anchor boxes, and adds CBAM modules to the neck. Furthermore, the use of a channel pruning algorithm reduces the computational cost of the model, while increasing detection speed. However, this method has not overcome the problem that the initial anchor boxes tend to be local optimal solutions due to K-Means clustering.

To summarize, when improving the model, it is important to balance the relationship between detection accuracy and computation speed. A good detection method should try to take into account the above two points. The most popular YOLO series detection method is YOLOv5, which is based on YOLOv4 and has four versions: s, m, l, and x. YOLOv5x is large in size and computationally complex. YOLOv5s and YOLOv5m are faster, but they are not accurate enough. YOLOv5l performs well in terms of speed and precision and is similar to YOLOv4 in terms of total parameters and total floating-point operations per second (FLOPS). For the above reasons, we modified YOLOv5l according to the characteristics of UAV aerial images to improve the detection performance of the model. This paper focuses on the following two points: (1) due to the abundance of small targets in UAV aerial images, there are situations such as occlusion, blur, dense arrangement, and sparse distribution, and they are often submerged in complex backgrounds. Therefore, it is essential to comprehensively improve the ability of the backbone to extract features. (2) During the flight of the UAV, the mounted device will transmit the captured images in real time, so it is necessary to pay attention to the detection speed and calculation cost of the model. The main improvement strategies in this paper are as follows:

1. Modifications to the backbone of YOLOv5. The residual blocks in the upper, middle, and lower layers of the backbone of YOLOv5 are improved with asymmetric convolutional blocks. After the Focus module, an Improved Efficient Channel Attention (IECA) module is added. The SPP module is improved by using grouped convolutions.
2. Use the K-Means++ algorithm to cluster different datasets to get more accurate anchor boxes. In the postprocessing of the model, Efficient Intersection over Union (EIOU) is used as the judgment basis for non-maximum suppression (NMS). We named this new NMS method EIOU-NMS.

The rest of this paper consists of the following: Section 2 gives a brief overview of the YOLOv5 and details the improvement strategies for the YOLO-UAV. Section 3 presents the experimental environment, parameter settings, used datasets, and evaluation indicators.

Detailed experimental steps, experimental results, and images for visualization are given to verify the effectiveness of the improved strategies and the superiority of the proposed method. Section 4 summarizes the proposed improvement strategies and compares the YOLO-UAV with similar recent studies. Section 5 concludes this paper and points out the future work ideas.

2. Method

2.1. YOLOv5 Algorithm Description

YOLOv5 changes the width and depth of the model by adjusting the parameters. According to its size, from small to large, it is divided into YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The structure of YOLOv5 can be divided into three parts, namely the backbone, neck, and head. The backbone is also known as a feature-extraction network. When the image is input, feature extraction will be performed by the backbone. The input image first goes through the Focus module. This module obtains a corresponding eigenvalue for every other point and concatenates the four independent feature layers to obtain the final result. At this time, the width and height information of the image will be concentrated into the channel, which solves the problem of information loss caused by downsampling. YOLOv5 uses the SiLU [38] activation function, which can be seen as a smoothed ReLU [39] activation function. SiLU has no upper bound but has a lower bound and is nonmonotonic. It still maintains good performance on deep networks, and this is beneficial for the model to improve the fitting effect by increasing the depth. The backbone of YOLOv5 contains the SPP module, which performs feature extraction through the max-pooling of different pooling kernel sizes, expanding the receptive field of the model. To fuse feature information at different scales, the neck will use three feature maps of different sizes extracted by the backbone for feature fusion. This part still uses the PANet [40] structure, which, based on the FPN [41], adds a channel from the shallow network to the deep network. This helps to combine the location and semantic information of shallow and deep features, thereby improving the utilization of information and speeding up the efficiency of information dissemination. The head of the model can be seen as the classifier and regressor of YOLOv5. Through a 1×1 convolution, it is judged whether there is an object in the feature map corresponding to it. During training, Mosaic data augmentation is used, which enriches the background of detected objects and helps to improve the efficiency of batch normalization. Label smoothing is used, as it helps mitigate the risk of model overfitting and improves generalization. An adaptive anchor box approach is used, which facilitates the automatic setting of the initial anchor box size when changing different datasets. The structure of YOLOv5 is shown in Figure 1.

2.2. Algorithm Design and Improvement

The images transmitted by UAVs in real time contain an abundance of small targets, and there are situations such as occlusion, blur, dense arrangement, and sparse distribution. In addition, the complex background also brings challenges to the detection of objects. We improved the YOLOv5l based on the characteristics and practical needs of UAV aerial images to increase the model's detection performance in small targets and complex background environments. The improvement strategies mainly focus on the backbone of the model, which comprehensively improves the feature-extraction capability. Additionally, the setting of anchor boxes and the suppression of redundant prediction boxes are also improved.

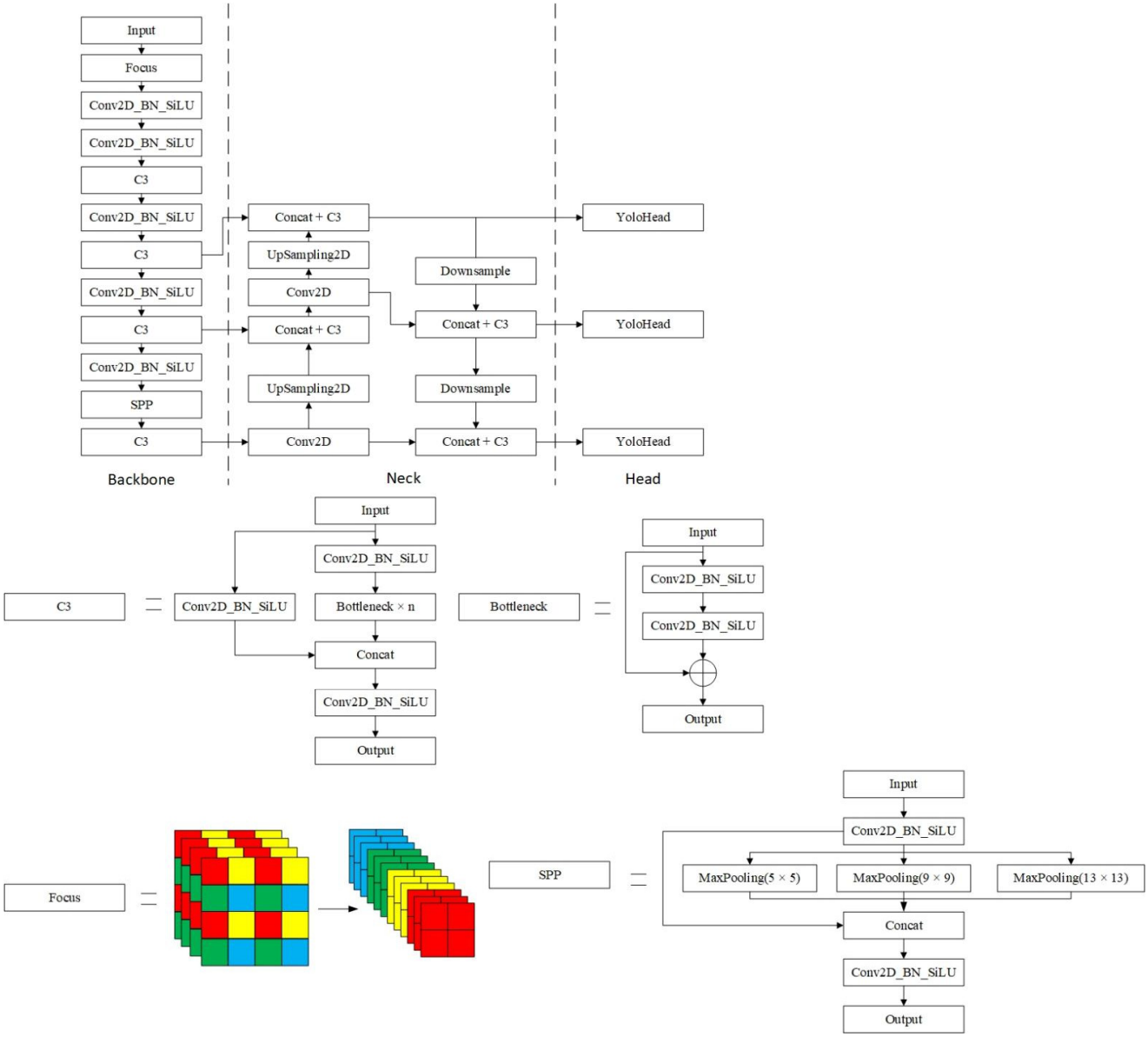

Figure 1. The Structure of YOLOv5.

2.2.1. Improvements to the Feature Extraction Module in Backbone

In the backbone of YOLOv5, four downsampling operations are performed, using convolutions with kernel size 3 × 3 and stride 2. After each downsampling, the features of the input feature map are extracted, using the C3 block. The core of the C3 block is the residual structure executed multiple times inside, which is a key step in feature extraction. In view of the characteristics of small targets, dense arrangements, sparse distribution, and complex backgrounds in UAV aerial images, we improved the C3 block to strengthen the feature-learning ability.

Information corresponding to different levels is extracted in each layer of the convolutional neural network. With the continuous deepening, more abstract information will be extracted, and there will be more ways of combining information between different levels. When dealing with the problems of gradient vanishing and gradient explosion, regularization can help deepen the network and alleviate the gradient problem. However, it will cause performance degradation, resulting in an increased error rate. Kaiming He et al. [39] proposed a residual structure to solve the abovementioned degradation problem and also reduce the influence caused by the vanishing gradient. As shown in Figure 2a, the residual

structure of the ResNet is implemented by using the "shortcut connection" method. Its simple identical mapping adds no additional parameters or computational complexity. Such a residual structure is easy to optimize, and the method of increasing the network depth can be easily adopted to improve the detection accuracy. Gao Huang et al. [42] proposed the structure of the Dense Convolutional Network (DenseNet), which connects each layer to other layers by using the shortcut connection approach. The dense blocks in DenseNet are shown in Figure 2b. This structure does not require repeatedly learning the existing features. It reduces the parameters, computational cost, and storage overhead of the model. DenseNet has strong generalization and very good resistance to overfitting, especially when the training data are relatively scarce. Yunpeng Chen et al. [43] pointed out that ResNet achieves implicit feature reuse but lacks the ability to extract new features. The DenseNet network will continuously explore new features, but the structure is redundant. In order to enjoy the respective advantages of the abovementioned networks at the same time, a Dual-Path Network (DPN) is proposed. This is a new connection method that realizes effective feature extraction and feature reuse. The structure of the DPN is shown in Figure 2c. It takes the residual structure as the backbone and adds a dense convolutional path. This network has higher parameter efficiency, a lower computational cost and memory consumption, and is easy to optimize. In a recent study, ShangHua Gao et al. [44] proposed a convolutional module called Res2Net through a hierarchical structure. The Res2Net module is shown in Figure 2d. The number of channels is first compressed with 1×1 convolution, and the channels are divided into multiple subsets. The original 3×3 convolution is then replaced by connecting a smaller group of convolutional blocks in a residual-like hierarchical style for finer feature extraction. Finally, feature fusion is accomplished by using 1×1 convolution to obtain the final result. The Res2Net module enhances the ability to represent features at multiple scales through channel splitting, while expanding the receptive field of the model. It shows the importance of this new dimension of scale in the network.

The core of extracting features in different modules is the 3×3 convolution inside. Using convolutions with larger kernel sizes, such as 5×5 or 7×7, for replacement is beneficial to expand the receptive field of the model, which helps to learn more efficient feature dependencies in a wider range of feature maps. Therefore, if the geometric size of the convolution kernel is reduced, part of the ability to extract features will be sacrificed. However, convolutions with larger kernels always require expensive computational costs and also increase the risk of vanishing gradients. In response to the aforementioned problems, Christian Szegedy et al. [45] pointed out that the complexity of operations can be reduced and the training speed can be accelerated by appropriate convolution decomposition. We can replace the $n \times n$ convolution with a combination of $1 \times n$ and $n \times 1$. Using such asymmetric convolutions can achieve the same receptive field and effectively reduce the computational complexity of the model. Xiaohan Ding et al. [46] proposed an asymmetric convolution block (ACB). This module strengthens square convolutions by using one-dimensional asymmetric convolutions. As shown in Figure 3, it consists of three layers of parallel convolutionals with kernel sizes of 3×3, 1×3, and 3×1, respectively. The horizontal and vertical one-dimensional convolutions in the ACB module explicitly enhance the central skeleton of the square convolutions, and the addition of the output results of the three-layer convolution makes the extracted features more robust. The ACB module can still give a good performance in the case of input rotational deformation.

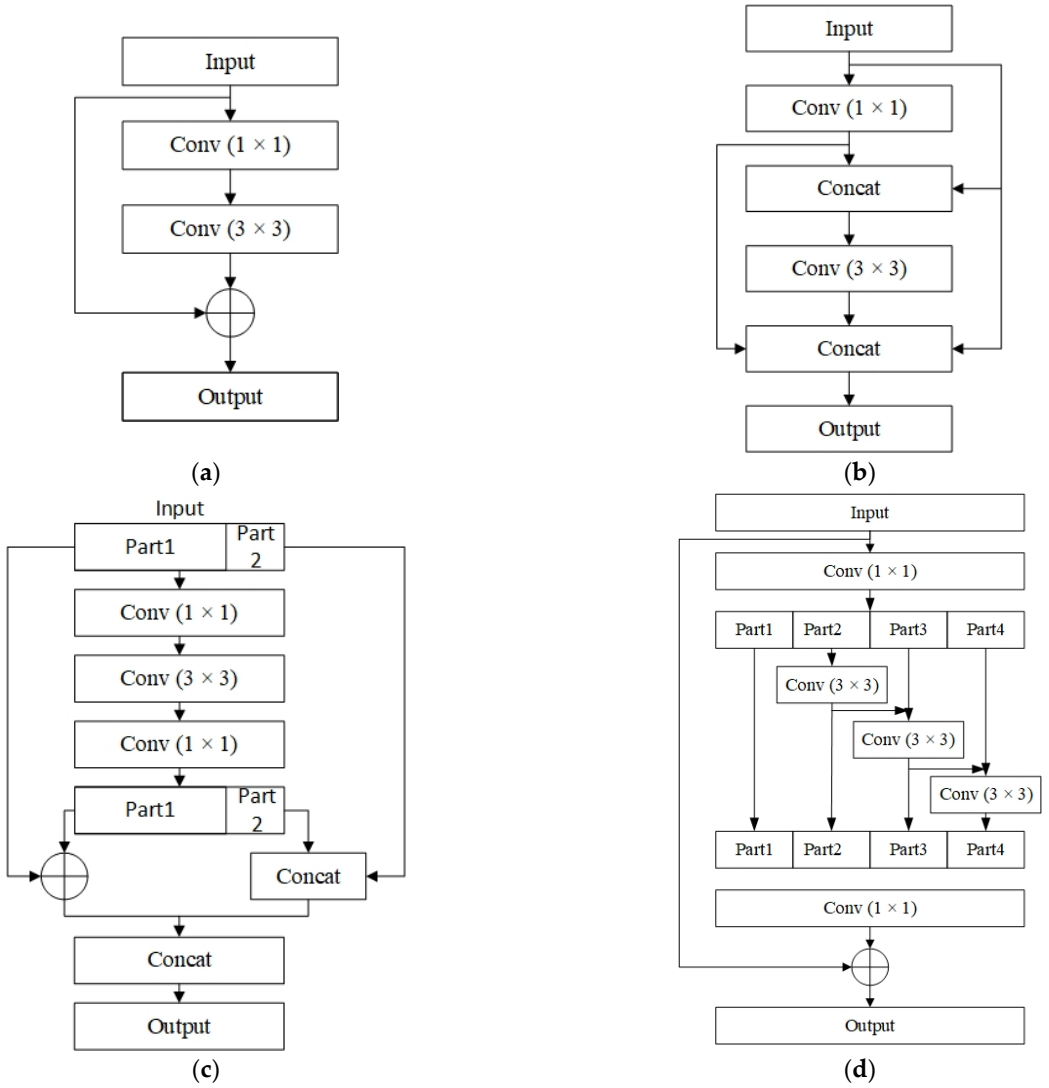

Figure 2. Components of different network structures: (**a**–**d**) basic modules that make up ResNet, DenseNet, DPN, and Res2Net, respectively.

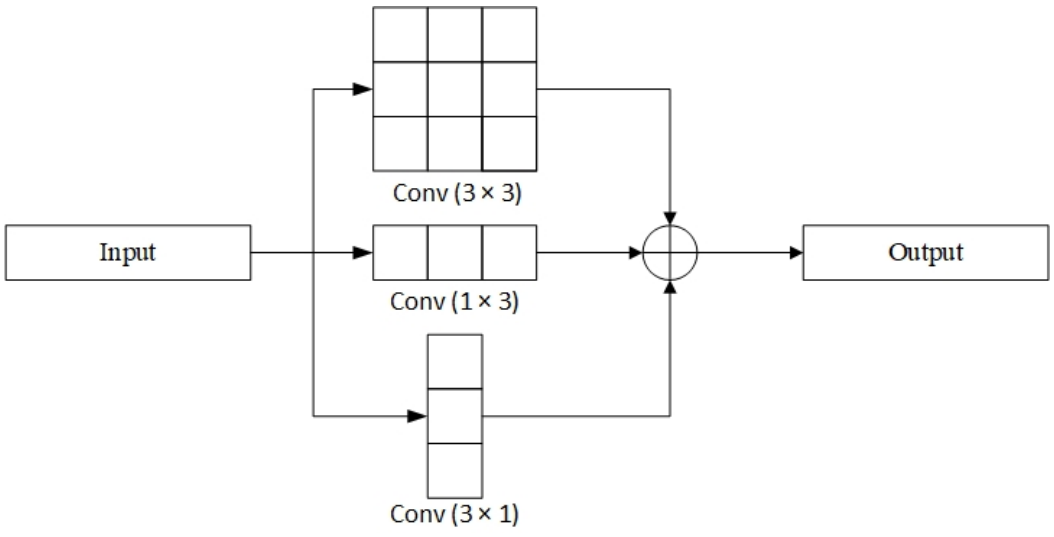

Figure 3. The structure of the ACB module.

Inspired by asymmetric convolution, we improved ResNet, DPN, and Res2Net and proposed three feature-extraction modules. They are named the Asymmetric ResNet (ASResNet) module, Asymmetric Enhanced Feature Extraction (AEFE) module, and Asymmetric Res2Net (ASRes2Net) module, respectively. Among them, the ASResNet and ASRes2Net modules are shown in Figure 4a,c, and they use ACB to replace the original 3×3 convolution. The improved module obtains the outputs of standard convolution and asymmetric convolutions and then adds them, which helps to enhance the feature-extraction ability of the network. The 1×3 and 3×1 convolution groups in the module are equivalent to a standard square 3×3 convolution. The overall effect of the module can be seen as an expansion of the previous convolution kernel size, that is, expanding the 3×3 kernel to a 5×5 size. Such improvements help to capture dependencies between signals on a larger scale, leading to more efficient feature representations. The AEFE module is shown in Figure 4b. We propose a new DPN-based network topology. When the feature map is input to this module, the number of channels is first compressed with 1×1 convolution. Then we use the ACB module to extract features. Immediately after that, it is split into two paths. One concatenates the extracted features with the compressed feature map, and the other is added with the input feature map after adjusting the number of channels. Finally, the results from the two paths are concatenated together, and 1×1 convolution is used for feature fusion to obtain the final output result. This module takes full advantage of residual networks and dense convolutional networks. This facilitates feature reuse and extraction, while also improving model generalization. We replaced residual blocks at different locations in the backbone of YOLOv5 with improved modules. The ASResNet modules were replaced in the first and second layers of the backbone to enhance the learning ability of the network. At the third layer of the backbone, the AEFE module, was used for replacement in order to extract more new features for subsequent processing. In the original backbone, the last layer contains the SPP module to expand the receptive field. This is consistent with the role of the ASRes2Net module, so the ASRes2Net module was replaced as the fourth layer of the backbone. The aforementioned improvements to the backbone enhance the ability of feature extraction in UAV aerial imagery.

2.2.2. Add the Channel Attention Module

The attention mechanism originated from the study of human vision [47]. Due to the bottleneck of information processing, we need to selectively focus on specific parts of the visual area and have to ignore certain information. This helps to take full advantage of existing visual-information-processing capabilities. In recent years, a variety of deep-learning fields have made extensive use of the attention mechanism. In image processing, it effectively promotes the network to focus on specific local information through a variety of implementation forms.

In a recent study, a new attention module named the IECA [48] has was proposed. It not only alleviates the inefficiency of the Squeeze-and-Excitation (SE) module [49] caused by acquiring all channel dependencies but also makes full use of the gains brought by different pooling methods. Figure 5 shows the IECA module. To obtain channel information, the input feature map is first processed by using mean-pooling and max-pooling. Then the number of adjacent channels is determined by 1D convolution and summed to obtain the corresponding attention map. Finally, the Sigmoid function is used to map the attention map to the range of 0 to 1, and then it is multiplied with the input to obtain the final output result.

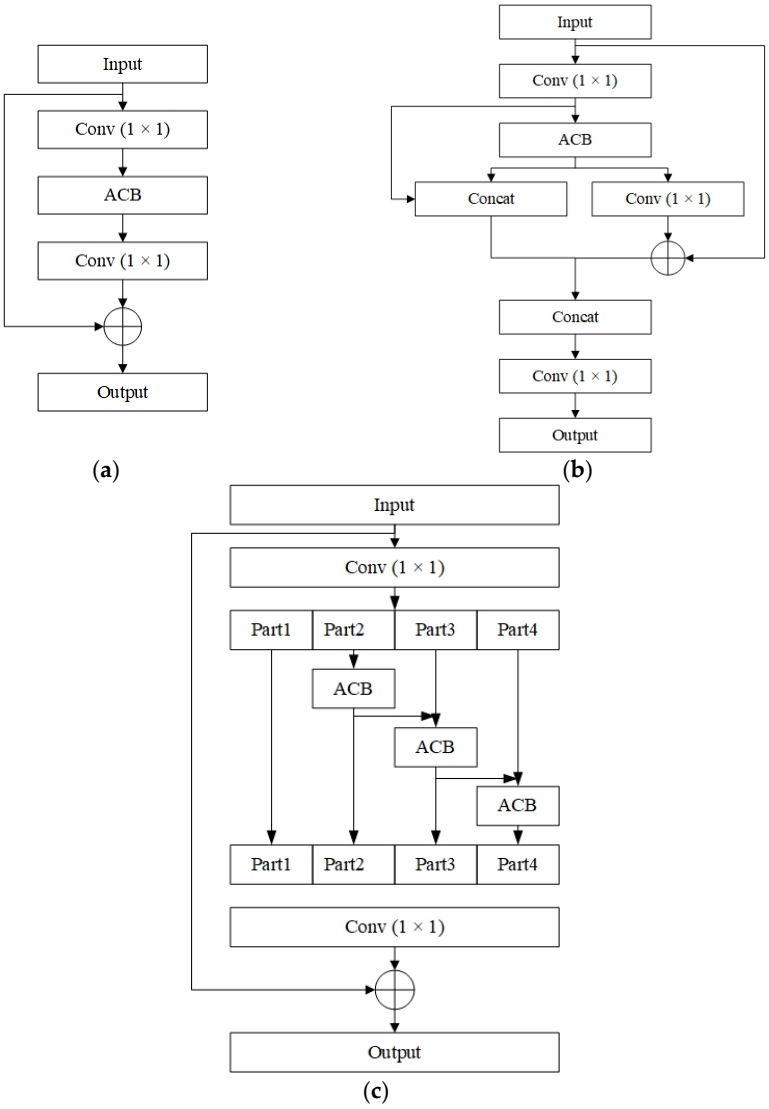

Figure 4. Three proposed feature extraction modules: (**a**–**c**) ASResNet module, AEFE module, and ASRes2Net module, respectively.

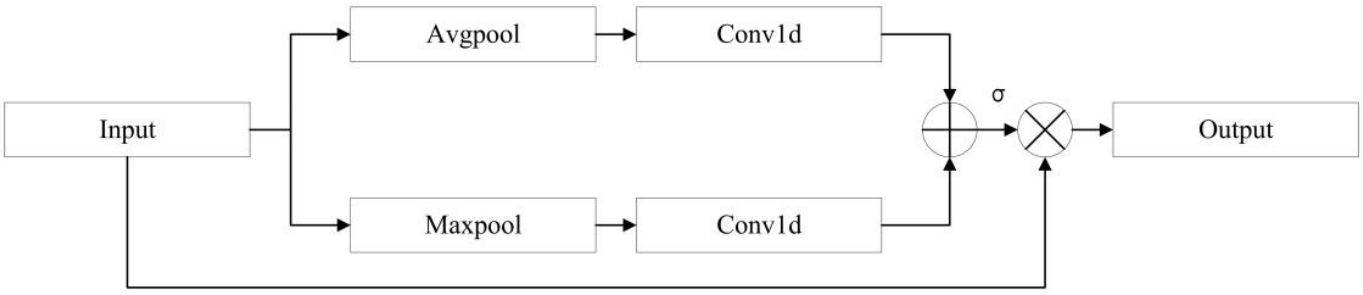

Figure 5. Structure of the IECA module.

In the backbone of YOLOv5, the Focus module slices the image, which integrates the size information of the image into the channel. This module expands the input RGB three-channel image to 12 channels by concatenating, which is a four-fold increase in channels. The advantage of this processing is that a downsampled feature map can be obtained without information loss. Because the number of feature map channels has been expanded multiple times in this process, and the interdependence between channels is more complex, it is necessary to add a channel attention module after Focus. Based on

the preceding analysis, we added an IECA module to assist the network in emphasizing important features, while suppressing irrelevant ones. Such an improvement is beneficial for suppressing the interference caused by complex backgrounds, as this is especially important in UAV aerial images.

2.2.3. Improvements Made to the SPP Module

In general, classification layers in convolutional neural networks are made up of fully connected layers. Such a structure requires a fixed number of features, resulting in the input image having to meet a certain required size. Kaiming He et al. [50] proposed the SPP module to deal with this constraint. This module effectively avoids distortion problems caused by operations such as the cropping, scaling, or stretching of image areas. In YOLOv3-SPP [29], the SPP module is improved based on the idea of a spatial pyramid. The improved module concatenates the outputs of multiple max-pooling layers. These layers have different pooling kernel sizes, fusing local and global features. This helps to expand the receptive field of the model and enhance the expressive power of the feature map. It is suitable for situations where the size of the objects in the image to be detected has a large difference.

The SPP module has been applied in all subsequent versions of YOLO. In addition, SPP has no shortage of improvements to it. Guohua Gao et al. [51] pointed out that the concatenation of max-pooling output results in the SPP module will reduce the resolution of the image and easily lose local information. Therefore, two dilated convolutional layers are added to the original module, which expands the space size and helps to capture multiscale global information at different sampling rates. Xuewen Wang et al. [37] pointed out that the operation of max-pooling is easy to highlight the strong responses in the input, but it will ignore the detailed features. To ensure that small targets are not missed, the SoftPool method was introduced. This method is a variant of the pooling operation, which prevents information loss as much as possible during the pooling process and is more friendly to the detection of small targets. Zongsheng Wu et al. [52] introduced atrous convolutions in the SPP module to improve the detection of small objects. Atrous convolutions with kernel sizes of 3×3 and dilation-rate sizes of 2, 5, and 9 are added after the max-pooling layers with pooling kernel sizes of 3, 5, and 9, respectively. The addition of atrous convolution expands the receptive range of feature maps, making it easier to capture rich contextual information and improve the detection effect of small targets.

The shortcomings of the improved SPP modules in the abovementioned can be summarized as follows: (1) After adding atrous convolutions, sparse sampling will affect the continuity of the output results, resulting in a lack of correlation between feature points. (2) Compared with mean-pooling and max-pooling, the SoftPool operation has high computational complexity. This can lead to longer model training and prediction times and even increase the risk of overfitting. (3) The newly added concatenated layers and convolution blocks will undoubtedly increase the additional computational burden and reduce the operation speed of the model.

We propose a new SPP module called the GSPP module. This module replaces the original two convolutions with grouped convolutions. Set the "group" parameter to 32. The GSPP module is shown in Figure 6. The addition of grouped convolution reduces the number of parameters, thus making the module more efficient. Additionally, grouped convolution acts like regularization, reducing the risk of model overfitting and improving the detection accuracy of the model. We compared the computational complexity of GSPP and SPP modules. When the shape of the input feature map is $13 \times 13 \times 1024$, the total parameters of GSPP are 84,992, and the total FLOPS is 14.62 MFlops. The total parameters of SPP are 2,624,512, and the total FLOPS is 443.8 MFlops. In summary, the GSPP module gives a better performance in terms of computational complexity and detection accuracy.

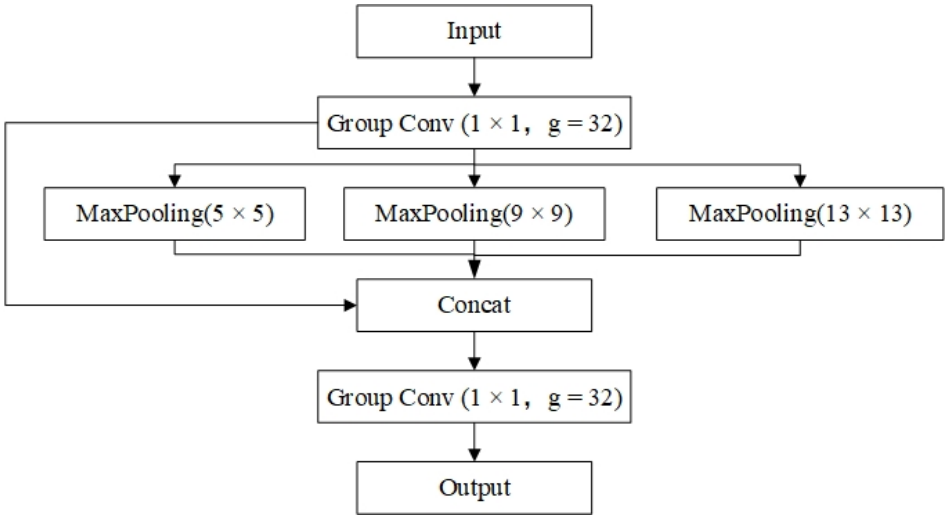

Figure 6. Structure of the GSPP module.

2.2.4. Get New Anchor Boxes Using the K-Means++ Algorithm

In YOLOv5, some anchor boxes with a picture size of 640 × 640 pixels and obtained from the COCO dataset are saved by default. Anchor boxes are clustered by using the K-Means algorithm and adjusted during training, using a genetic algorithm. The K-Means algorithm randomly selects a set of points as the initial cluster centers. This results in the convergence being heavily dependent on the center initialization, and the clustering results of different initial centers may be completely different. The K-Means++ [53] algorithm has been proposed for this problem. The basic idea is that the initial cluster centers should be as far apart as possible. The purpose is to make the randomly selected center points no longer tend to the local optimal solution but tend to the global optimal solution as much as possible. Because of the characteristics of the targets in UAV aerial images, it is not suitable to use anchor boxes that have been preset based on natural images. To make them more accurate, we used the K-Means++ algorithm to cluster the used dataset. Table 1 shows the steps of the K-Means++ algorithm.

Table 1. The steps of the K-Means++ algorithm.

	Step Description
Step 1	The first cluster center is selected at random after moving the centers of all marked rectangles in the dataset to the origin of the coordinate system.
Step 2	Calculate the shortest distance of each sample to the currently known cluster center and the probability that each sample is selected as the next cluster center. Then, using the roulette method, choose the next cluster center.
Step 3	Step 2 needs to be repeated until the required number of cluster centers is selected.
Step 4	Calculate the distance between the center of each sample and the cluster center, and then divide each sample into the closest cluster.
Step 5	Calculate the average of all sample widths and heights for each cluster as the new cluster center.
Step 6	Steps 4 and 5 need to be repeated until the cluster center movement is less than a predetermined value or the number of calculations meets the requirements.

In the next section of experiments, we show the anchor boxes and visualization results obtained by clustering using different datasets.

2.2.5. Suppressing Redundant Prediction Boxes Using the EIOU-NMS Method

Target-detection algorithms use non-maximum suppression (NMS) as a necessary postprocessing step to get rid of redundant prediction boxes for the same object. Adopting a suitable NMS method is not only beneficial to improving the prediction efficiency but

can also improve the detection accuracy. The greedy NMS method measures the degree of overlap between the two prediction boxes by using the Intersection over Union (IoU). By calculating the *IoU* value between the predicted box with the highest score and the other boxes, the parts with a higher degree of overlap than expected are removed. The traditional NMS method is not conducive to target detection in UAV aerial images because objects are frequently arranged densely and obscured from one another. It is easy to delete occluded objects by mistake, reducing the recall rate of the model.

Aiming at the shortcomings of greedy NMS, the common improvement methods are Soft-NMS [54] and DIOU-NMS [55]. Soft-NMS does not directly zero out the prediction score; it also takes the calculated *IoU* value as the input of the Gaussian penalty function and multiplies the result with the initial score as the new score for this prediction box. The new score was adjusted for the degree of overlap. Since the penalty function used is continuous, sudden changes in the sorted list in detection are avoided. The DIOU-NMS uses Distance–Intersection over Union (DIoU) to measure the distance between the highest scoring prediction box and other prediction boxes on the same object. In this way, when suppressing redundant boxes, the distance between their center points is also involved, thereby effectively avoiding the conflict between the prediction boxes of overlapping targets.

During the operation of Soft-NMS, a Gaussian penalty function is added. The function is shown in Equation (1), where b is the prediction box with the highest score, b_i is other prediction boxes on the same object, σ is a constant, and D represents the final result after NMS. The exponential operation included in it is not only computationally complex but also affects the speed of postprocessing. The value of σ cannot be obtained by an adaptive method, so it is necessary to repeatedly test to find the optimal value. DIOU-NMS uses *DIoU* to measure the degree of overlap between boxes, but the new improved *IoU* variant may produce better results than *DIoU*:

$$f(x) = e^{-\frac{IoU(b,b_i)^2}{\sigma}}, \ \forall b_i \notin D \tag{1}$$

We propose a new method that uses *EIoU* as the judgment basis for NMS, called EIOU-NMS. As defined in Equations (2)–(4), where S_i is the prediction score of different target categories; B is the prediction box with the highest score; B_i is other prediction boxes on the same object; ε is the threshold; ρ^2 is the Euclidean distance; b, w, and h are the prediction box's center point, width, and height, respectively; and c, c_w^2 and c_h^2 are the diagonal distance, width, and height of the circumscribed rectangles of the two prediction boxes. The *EIoU* [56] calculation method is shown in Equation (5), which adds the loss of width and height on the basis of *DIoU*. This makes it necessary to pay attention not only to the distance between two center points but also to the difference between width and height when suppressing redundant prediction boxes. These improvements enable EIOU-NMS to better measure the degree of coincidence of prediction boxes, which is more conducive to suppressing redundant prediction boxes:

$$S_i = \begin{cases} S_i \ , \ IoU - R_{EIoU}(B, B_i) < \varepsilon \\ 0 \ , \ IoU - R_{EIoU}(B, B_i) \geq \varepsilon \end{cases} \tag{2}$$

$$IoU = \frac{area(B \cap B_i)}{area(B \cup B_i)} \tag{3}$$

$$R_{EIoU}(B, B_i) = \frac{\rho^2(b, b_i)}{c^2} + \frac{\rho^2(w, w_i)}{c_w^2} + \frac{\rho^2(h, h_i)}{c_h^2} \tag{4}$$

$$EIoU = IoU - \frac{\rho^2(b, b_i)}{c^2} - \frac{\rho^2(w, w_i)}{c_w^2} - \frac{\rho^2(h, h_i)}{c_h^2} \tag{5}$$

In summary, the improved model is shown in Figure 7. YOLO-UAV is improved on the basis of YOLOv5l. The improvement parts are mainly in the backbone of the model. In addition, the setting of anchor boxes and the suppression of redundant boxes were

also improved. The structure of YOLO-UAV is divided into the backbone, neck, and head. When the shape of the input image is 416 × 416 × 3. First, extract features through the backbone and output three feature maps with shapes of 52 × 52 × 256, 26 × 26 × 512 and 13 × 13 × 1024. Then feature fusion through the neck is carried out to strengthen feature extraction. Finally, the final prediction result is obtained by the postprocessing operation of the head.

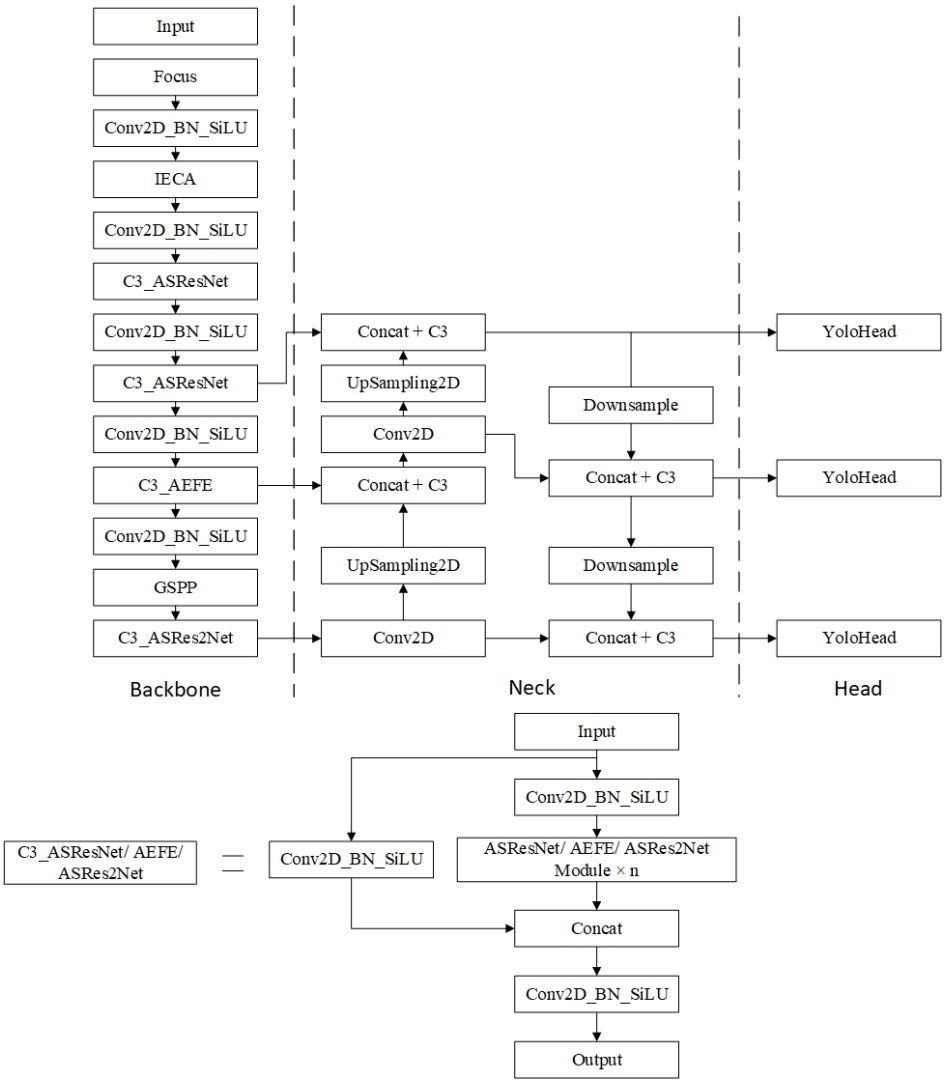

Figure 7. Structure of the YOLO-UAV.

3. Experiments and Results

In this section, we discuss a series of experiments we conducted on image-classification datasets, generic-object-detection datasets, and UAV aerial image datasets, including CIFAR-10 [57], PASCAL VOC, VEDAI [58], VisDrone 2019 [59], and Forklift [48]. The experiments are divided into five parts: (1) the anchor boxes obtained by K-Means++ algorithm clustering on different datasets are given, and the clustering results are visualized; (2) since the improvements mainly focus on the backbone of the model, ablation experiments were performed on the image classification and detection tasks, respectively, to verify the effectiveness of the improvement strategies; (3) the proposed method is compared to several other advanced detection methods to verify its superiority; (4) comparative experiments were carried out on three UAV aerial image datasets to verify the superiority of the proposed method on UAV aerial imagery; and (5) three NMS methods are compared

on multiple datasets to verify the effectiveness of the proposed EIOU-NMS method in suppressing redundant prediction boxes.

3.1. Experimental Environment and Training Parameter Settings

As shown in Tables 2 and 3, the experimental environment and some uniform parameter criteria set in experiments are given. If there is no special description later, the parameter settings in the table are used by default.

Table 2. Experimental environment.

Environment	Versions or Model Number
CPU	i7-10700k
GPU	RTX 2070 SUPER
OS	Windows 10
Python	3.8.12
Pytorch	1.8.1
Torchvision	0.9.1
OpenCV-Python	4.5.5.64

Table 3. Parameter criteria.

Input Size	Optimizer	Momentum	Batch Size	Training Epoch	Training, Validation, and Test Set Ratio
416 × 416	SGD	0.937	4	100	8:1:1

3.2. Dataset

The CIFAR-10 is a small dataset for image classification. The dataset has an image size of 32 × 32 pixels and has 10 categories, including 50,000 training images and 10,000 testing images. It will be used for ablation experiments for the classification task of the backbone.

The PASCAL VOC includes the VOC2007 and VOC2012 datasets. Among them, the VOC2007 dataset contains 20 object categories and 4952 annotated images. This dataset was used for ablation experiments, comparative experiments of other methods, and comparative experiments of different NMS methods.

VEDAI is a dataset for vehicle detection in aerial images. Among them, the color image sub-dataset of 512 × 512 pixels contains eight categories, except "other", with a total of 1246 annotated images. This dataset was used for comparative experiments on UAV aerial images and comparative experiments of different NMS methods.

The VisDrone 2019 dataset contains a large number of objects to be detected, some of which are very small due to the perspective of the UAV. This dataset has 7019 annotated images in total. It is divided into 10 categories, some of which have relatively similar characteristics. This dataset was used for comparative experiments on UAV aerial images and comparative experiments of different NMS methods.

The Forklift dataset is a forklift-targeted dataset based on UAV aerial imagery established by us. Initially, there were 1007 annotated images in the dataset. The number of images was then expanded to 2022, and images similar to the natural horizontal viewing angle were replaced. This part of the shooting task was completed by two professional UAV pilots. The UAVs used were DJI Mavic 2, Mavic 3, and Jingwei M300RTK, and the mounted cameras are Zenmuse P1 and Zenmuse H20T. The UAV was flying at an altitude of between 100 and 150 m when filming. We annotated the obtained UAV aerial images and invited two pilots to check and correct them. This dataset was used for comparative experiments on UAV aerial images and comparative experiments of different NMS methods. Figure 8 shows some example images from the VEDAI, VisDrone 2019, and Forklift datasets.

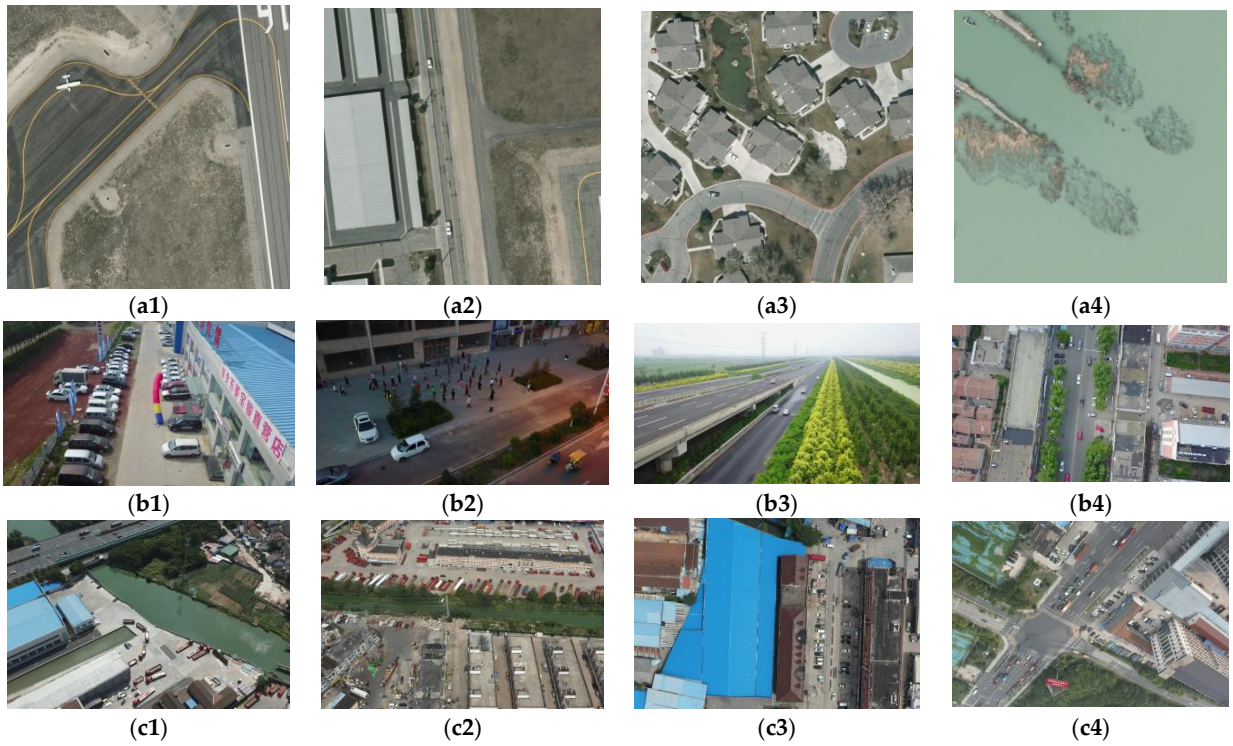

Figure 8. Some example images from the VEDAI, VisDrone 2019, and Forklift datasets: (**a1**–**a4**), (**b1**–**b4**), and (**c1**–**c4**) from the VEDAI, VisDrone 2019, and Forklift datasets, respectively.

3.3. Evaluation Indicators

The evaluation indicators to evaluate the performance of the detection method are Precision (*P*), Recall (*R*), *F*1 score, Average Precision (*AP*), and Mean Average Precision (*mAP*). They are calculated as shown in Equations (6)–(10), where *TP* is True Positive, *FP* is False Positive, *FN* is False Negative, and *C* is the total number of categories. Additionally, total parameters and total FLOPS are used to measure model size and computational complexity, and top-1 accuracy is used to measure image classification performance:

$$P = \frac{TP}{TP + FP} \tag{6}$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = \frac{2 \times R \times P}{R + P} \tag{8}$$

$$AP = \int P(R)dR \tag{9}$$

$$mAP = \frac{1}{C}\sum_{j}^{C} AP_j \tag{10}$$

3.4. Experimental Results

3.4.1. Results of Clustering Different Datasets

Clustering on PASCAL VOC, VEDAI, VisDrone 2019, and Forklift datasets was performed by using the K-Means++ algorithm. The number of cluster centers was set at nine. Table 4 lists the default anchor boxes and our obtained anchor boxes. Figure 9 shows the visualization results, where different colors represent different clusters, and "×" represents the cluster center. In subsequently mentioned experiments, the anchor boxes listed in the table were used.

Table 4. Default anchor boxes and anchor boxes obtained by clustering.

Dataset	Anchor Boxes
Default anchor boxes	(10, 13), (16, 30), (33, 23), (30, 61), (62, 45), (59, 119), (116, 90), (156, 198), (373, 326)
PASCAL VOC	(15, 26), (25, 69), (50, 44), (49, 127), (100, 94), (94, 201), (237, 169), (167, 299), (348, 347)
VEDAI	(17, 7), (8, 17), (21, 8), (17, 11), (13, 15), (11, 24), (17, 17), (31, 12), (26, 28)
VisDrone 2019	(1, 4), (2, 9), (5, 6), (5, 13), (10, 10), (8, 20), (19, 17), (15, 31), (34, 42)
Forklift	(3, 4), (4, 7), (6, 8), (4, 12), (6, 16), (9, 11), (12, 16), (8, 25), (17, 26)

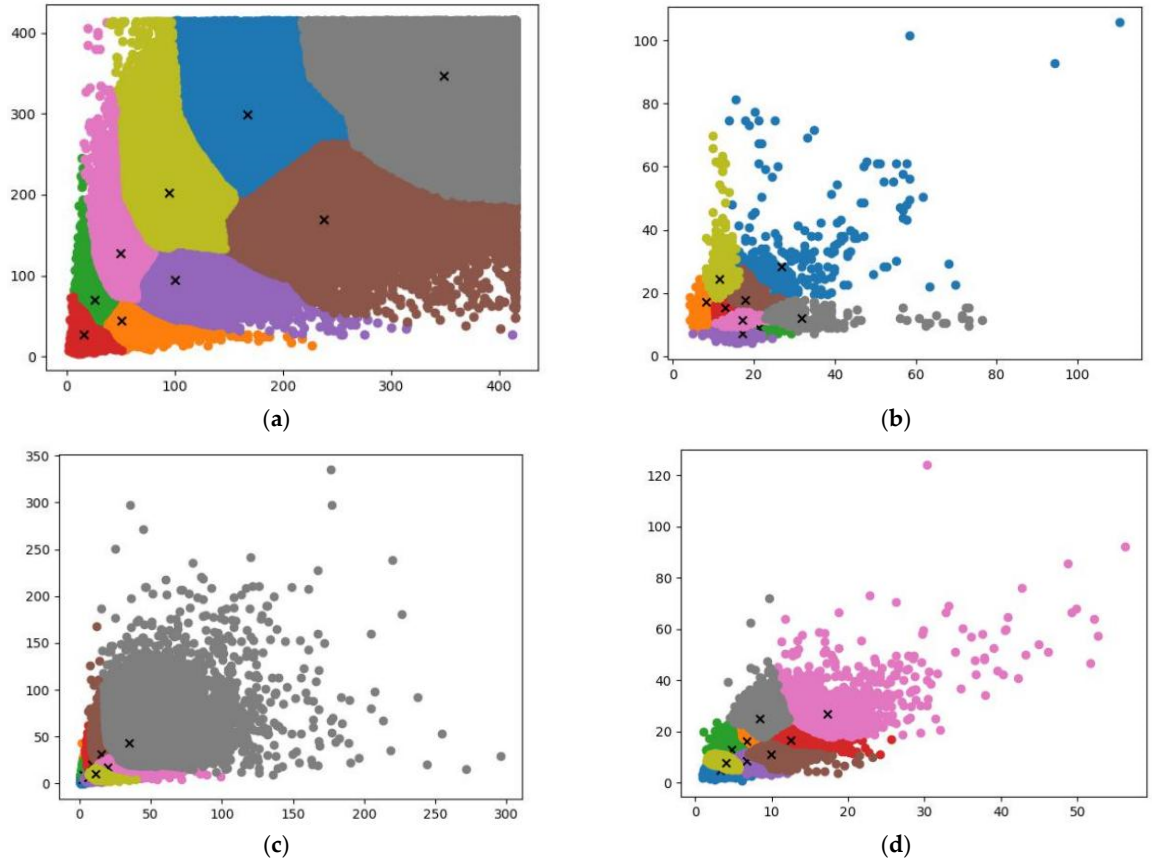

Figure 9. Visualization of K-Means++ clustering results for different datasets: (**a**–**d**) results of clustering on PASCAL VOC, VEDAI, VisDrone 2019, and Forklift datasets, respectively.

3.4.2. Ablation Experiments

We improved the backbone of YOLOv5l to strengthen the ability to extract features. Ablation experiments were conducted on the image classification and the detection tasks, respectively, to verify the efficacy of the improved strategies.

(1) Ablation experiments on classification tasks

In this part of the experiments, an adaptive average pooling layer and a fully connected layer were additionally added after the backbone of the model for image classification. The dataset used was CIFAR-10. The input image size was set to 32 × 32 pixels, and the batch size was set to 64. We divided the ablation experiments into the following five steps: In Step 1, the residual blocks in Layers 1 to 4 of the backbone were replaced with the ASResNet module. In Step 2, we used the ASRes2Net module to replace the fourth layer. In Step 3, we added the IECA module after the Focus. In Step 4, we used the AEFE module to replace the third layer. In Step 5, we used GSPP to replace the original SPP module. The results of the ablation experiments on the classification task of the backbone are shown in Table 5. In addition, the top-one accuracy of the backbone of YOLOv4 and YOLOv5x is also shown.

Table 5. Results of ablation experiments on classification tasks.

Model	Step 1	Step 2	Step 3	Step 4	Step 5	Top-1 Accuracy	An Improvement over YOLOv5l
YOLOv4						81.18%	
YOLOv5l						78.49%	
YOLOv5x						79.77%	
	√					80.93%	+2.44%
	√	√				82.00%	+3.51%
	√	√	√			83.04%	+4.55%
	√	√	√	√		85.56%	+7.07%
YOLO-UAV	√	√	√	√	√	85.69%	+7.20%

The experimental results lead to three conclusions: (1) as compared with YOLOv5, the backbone of YOLOv4 achieves higher top-one accuracy on classification tasks, indicating that the backbone of YOLOv4 is better than YOLOv5 in its ability to extract features in image classification; (2) due to the increased width and depth of the YOLOv5x, it has a higher top-one accuracy than the YOLOv5l; (3) the top-one accuracy of the YOLOv5l is 78.49%. After improvement, it increased to 85.69%, an increase of 7.20%. This indicates that the proposed improvement strategies are beneficial for enhancing the feature-extraction capability of the backbone.

We compared the total parameters and total FLOPS of the backbones of YOLOv4, YOLOv5l, YOLOv5x, and YOLO-UAV, which measure the size and computational complexity of the backbone. Table 6 shows the comparison results.

Table 6. The complexity of different model backbones.

Model	Total Parameters	Total FLOPS
YOLOv4	26,617,184	17.34 GFlops
YOLOv5l	27,075,968	16.03 GFlops
YOLOv5x	50,301,600	30.49 GFlops
YOLO-UAV	27,506,691	21.67 GFlops

Table 6 shows that the backbone of YOLOv5x has the highest total parameters and total FLOPS. The complexity of YOLO-UAV is not much different from that of YOLOv5l, with only a slight increase. The total parameters and total FLOPS of the backbone of YOLO-UAV are less than YOLOv5x, but its top-one accuracy on image classification tasks is 5.92% higher. This shows that YOLO-UAV has excellent parameter efficiency and achieves a good balance between speed and accuracy.

(2) Ablation experiments on detection tasks

In this part of the ablation experiments, the dataset used was the VOC2007 dataset. The ablation experiments had a total of seven steps, of which the first five steps were the same as the above. In Step 6, we used the anchor boxes mentioned in the previous section. In Step 7, we used EIOU-NMS instead of greedy NMS for suppressing redundant prediction boxes. The results of the ablation experiments are shown in Table 7.

Table 7. Results of ablation experiments on the detection task.

Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	mAP	An Improvement over YOLOv5l
YOLOv4								80.17%	
YOLOv5l								79.96%	
YOLOv5x								83.92%	
	√							81.20%	+1.24%
	√	√						82.25%	+2.29%
	√	√	√					83.57%	+3.97%
	√	√	√	√				84.87%	+4.91%
	√	√	√	√	√			85.02%	+5.06%
	√	√	√	√	√	√		85.26%	+5.30%
YOLO-UAV	√	√	√	√	√	√	√	85.35%	+5.39%

From the table above, it can be seen that the mAPs of YOLOv4, YOLOv5l, and YOLOv5x are 80.17%, 79.96%, and 83.92%, respectively. After the improvement of the backbone of YOLOv5l, the mAP increased to 85.02%, resulting in increases by 4.85%, 5.06%, and 1.10%, respectively. On this basis, after the remaining two points of improvement, mAP increased to 85.35%, with an increase of 5.18%, 5.39%, and 1.43%, respectively. To show the detection performance improvement more clearly, we visualized the feature map output by the backbone. The visualization results of different kinds of feature maps in the VOC2007 dataset are shown in Figure 10. It is clearly observed in the form of heat maps that the features extracted by YOLO-UAV cover the target more accurately, and it is beneficial to alleviate the interference of complex backgrounds. In summary, the experimental results show that the above improvement strategies work well and are beneficial to an overall improvement in the performance of model detection.

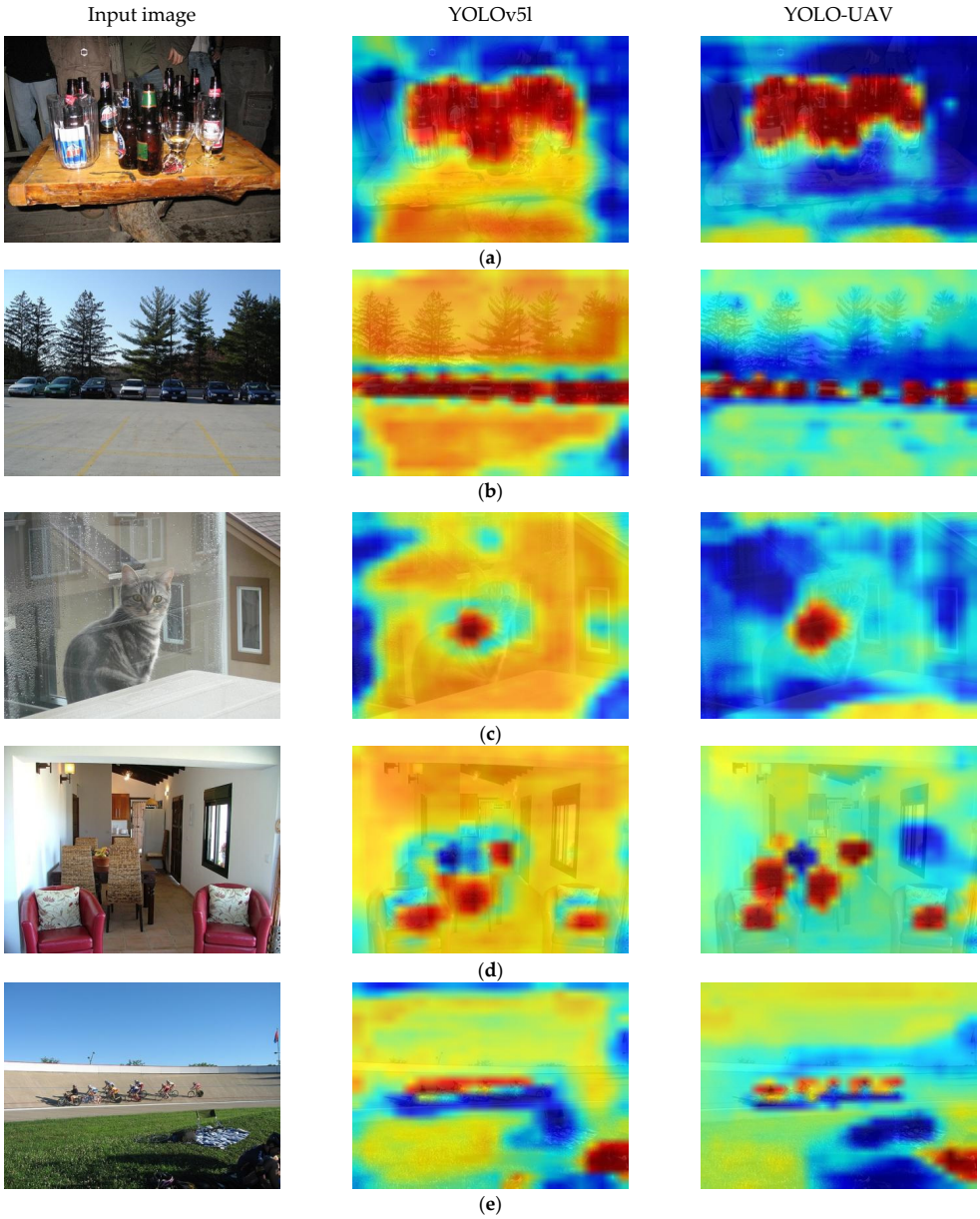

Figure 10. Visualization of different types of feature-extraction results in the VOC2007 dataset. (**a**–**e**) The corresponding types are bottle, car, cat, chair, and person.

3.4.3. Comparison with Other Object Detection Methods

The detection methods used for comparison include Faster R-CNN, SSD, YOLOv3, EfficientDet [60], YOLOv4-Tiny, YOLOv4, and YOLOv5. The dataset used is VOC2007. Table 8 shows the experimental results of the comparative experiments.

Table 8. Experimental results of comparative experiments.

Model	The Backbone of the Model	mAP
Faster R-CNN	ResNet50	70.53%
SSD	VGG16	71.47%
YOLOv3	Darknet53	68.44%
EfficientDet-D0	EfficientNet	71.67%
EfficientDet-D1	EfficientNet	76.34%
YOLOv4-Tiny	Tiny CSPDarknet53	69.97%
YOLOv4	CSPDarknet53	80.17%
YOLOv5l	CSPDarknet_l	79.96%
YOLOv5x	CSPDarknet_x	83.92%
YOLO-UAV	Figure 7	85.35%

The experimental results show that YOLO-UAV achieves the highest mAP, which verifies the superiority of the improved model.

3.4.4. Experiments on the UAV Aerial Image Dataset

YOLO-UAV is improved on the basis of YOLOv5l, so in the following experiments, we focused on the difference in mAP between YOLO-UAV and YOLOv5l. The datasets used are the VEDAI, VisDrone 2019, and Forklift datasets. Inspired by transfer learning, when training on UAV aerial images, the pre-training weights used are all from the above-mentioned comparative experiments. At this time, the epoch of training is modified to 500. The experimental results of YOLOv5l and YOLO-UAV on the UAV aerial image datasets are shown in Tables 9–11, respectively.

Table 9. Results on the VEDAI dataset.

Model	YOLOv5l				YOLO-UAV			
	AP	F1	R	P	AP	F1	R	P
boat	45.96%	0.36	22.22%	100.00%	55.21%	0.67	50.00%	100.00%
camping car	74.76%	0.69	63.83%	75.00%	79.00%	0.73	76.60%	69.23%
car	69.32%	0.69	71.43%	66.04%	71.47%	0.73	77.55%	68.67%
pickup	50.34%	0.58	47.27%	74.29%	49.73%	0.53	40.91%	75.00%
plane	94.09%	0.83	70.59%	100.00%	99.35%	0.97	100.100%	94.44%
tractor	41.19%	0.47	33.33%	80.00%	55.44%	0.63	50.00%	85.71%
truck	51.25%	0.48	35.48%	73.33%	60.07%	0.51	35.48%	91.67%
van	15.56%	0.17	9.73%	67.24%	18.52%	0.20	11.11%	100.00%
mAP			55.31%				61.10%	

Table 10. Results on the VisDrone 2019 dataset.

Model	YOLOv5l				YOLO-UAV			
	AP	F1	R	P	AP	F1	R	P
awning-tricycle	4.41%	0.01	0.76%	37.50%	11.93%	0.11	6.09%	55.81%
bicycle	0.07%	0.00	0.00%	0.00%	10.08%	0.04	2.29%	92.86%
bus	50.85%	0.56	41.84%	86.22%	52.29%	0.58	43.70%	87.54%
car	57.27%	0.62	47.92%	88.89%	58.96%	0.63	48.92%	90.32%
motor	22.80%	0.08	4.22%	81.76%	26.03%	0.16	8.84%	80.61%
pedestrian	22.28%	0.21	11.84%	87.29%	24.13%	0.23	13.19%	86.30%
people	6.11%	0.00	0.03%	100.00%	11.24%	0.04	2.03%	83.56%
tricycle	13.86%	0.06	3.00%	60.00%	23.11%	0.17	9.83%	71.95%
truck	48.36%	0.52	37.92%	83.84%	50.66%	0.55	41.24%	84.32%
van	34.38%	0.44	32.07%	71.25%	36.53%	0.45	32.56%	72.64%
mAP			26.04%				30.50%	

Table 11. Results on the Forklift dataset.

Model	YOLOv5l				YOLO-UAV			
	AP	**F1**	**R**	**P**	**AP**	**F1**	**R**	**P**
forklift	61.53%	0.62	47.37%	88.52%	70.43%	0.71	57.96%	91.57%
mAP	61.53%				70.43%			

According to the experimental results, it can be seen that the mAP of YOLOv5l on the VEDAI, VisDrone 2019, and Forklift datasets is 55.31%, 26.04%, and 61.53%, respectively. YOLO-UAV is 61.10%, 30.50%, and 70.43%, respectively. The detection accuracy of YOLO-UAV is better than that of YOLOv5l on all three datasets, and the mAP is improved by 5.79%, 4.46%, and 8.90%, respectively. The experimental results verify the superiority of the improved methods in UAV aerial images. YOLO-UAV handles the challenges brought by factors such as small targets, dense arrangement, sparse distribution, and complex backgrounds very well, and it has a better performance in UAV aerial images. Figures 11–13 show some post-detection results on the VEDAI, VisDrone 2019, and Forklift datasets.

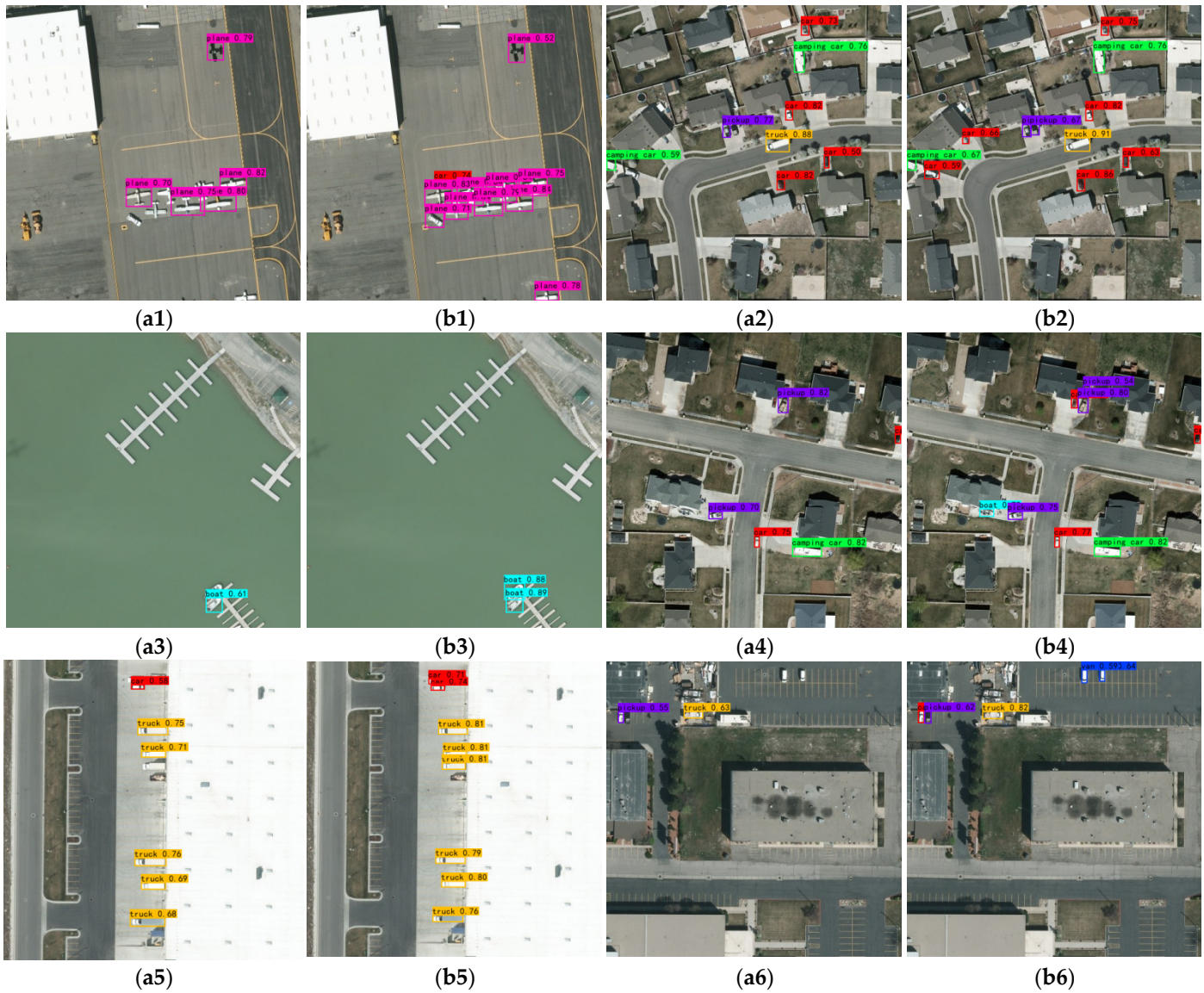

Figure 11. Detection results on the VEDAI dataset: (**a1**–**a6**) and (**b1**–**b6**) are the detection results of YOLOv5l and YOLO-UAV, respectively.

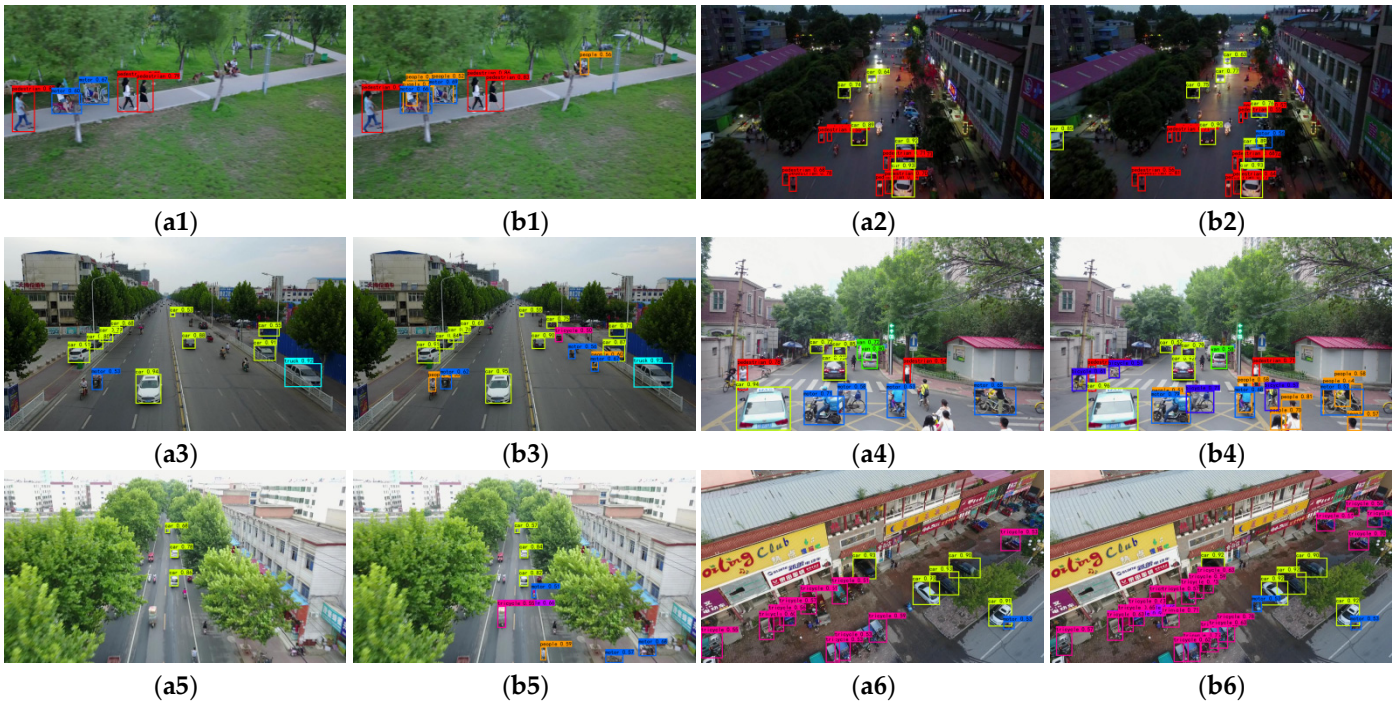

Figure 12. Detection results on the VisDrone 2019 dataset: (**a1**–**a6**) and (**b1**–**b6**) are the detection results of YOLOv5l and YOLO-UAV, respectively.

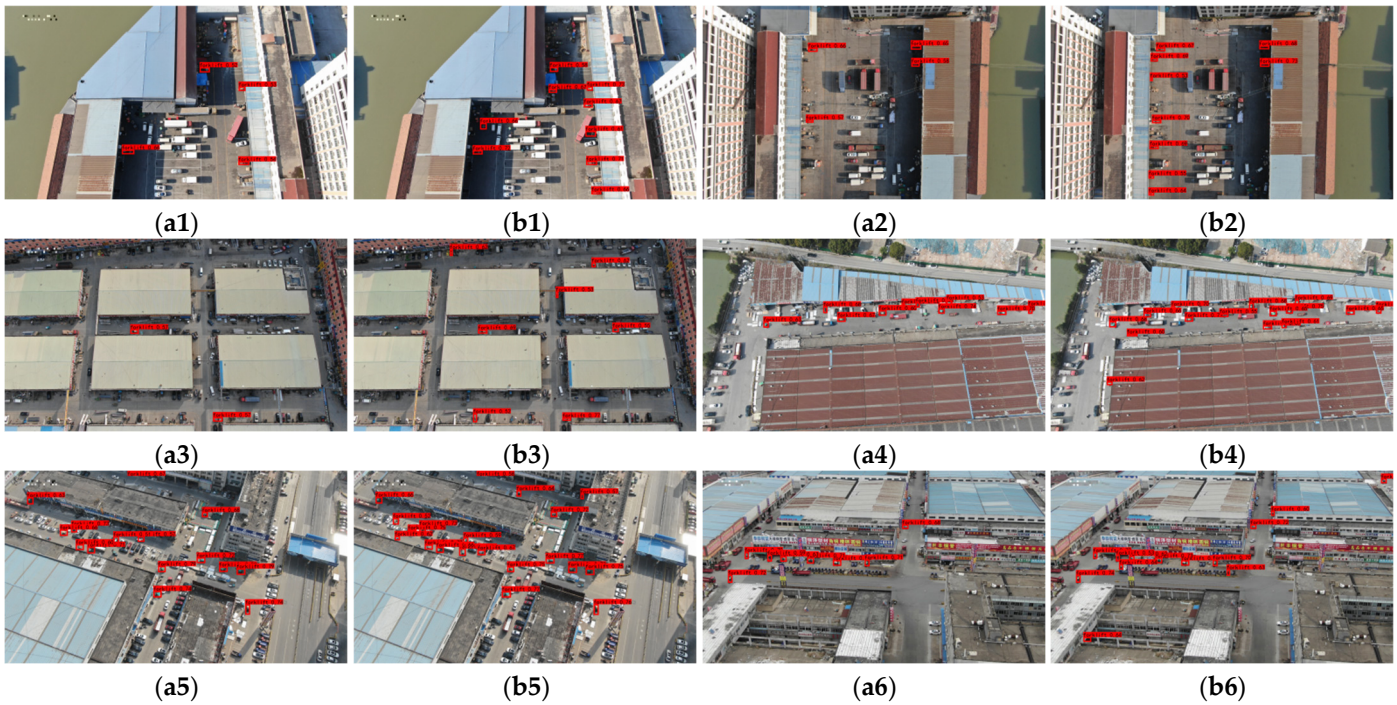

Figure 13. Detection results on the Forklift dataset: (**a1**–**a6**) and (**b1**–**b6**) are the detection results of YOLOv5l and YOLO-UAV, respectively.

3.4.5. Comparison of Different NMS Methods in Multiple Datasets

We conducted comparative experiments on multiple datasets to verify the superiority of the EIOU-NMS method. The datasets include VOC2007, VEDAI, VisDrone 2019, and Forklift datasets. The NMS methods used for comparison included EIOU-NMS, DIOU-NMS, and greedy NMS. In this part of the experiments, the detection method used was

YOLO-UAV, and only different NMS methods were replaced on its basis. We set the threshold for non-maximal suppression to 0.30. The precision of mAP was increased to five decimal places to better show the difference in mAP. The comparison results are shown in Table 12.

Table 12. The effects of different NMS methods on mAP.

Dataset	EIOU-NMS	DIOU-NMS	Greedy NMS
VOC2007	85.34573%	85.27970%	85.26226%
VEDAI	61.09853%	61.07132%	61.0585%
VisDrone 2019	30.49512%	30.45333%	30.41479%
Forklift	70.43432%	70.38989%	70.34265%

From the data in the above table, it can be seen that mAP is the highest when using EIOU-NMS. We verified that the proposed EIOU-NMS method can more effectively suppress redundant prediction boxes, assisting in improving the model's detection accuracy. The performance improvement benefits from the EIOU indicator, which makes the suppression criteria not only limited to the overlapping area of the two prediction boxes and the distance between the center points, but also pays attention to the difference in width and height between boxes. In addition, the EIOU-NMS method can be easily added to different models, without additional training.

4. Discussion

From the above experimental results, it can be seen that YOLO-UAV gives a better detection performance than YOLOv5l. The proposed improvement strategies include modifications to the backbone of the model and optimization of other parts. Specifically, they can be divided into the following five parts: (1) Inspired by asymmetric convolution, we modified ResNet, DPN, and Res2Net and proposed three feature-extraction modules, named ASResNet module, AEFE module, and ASRes2Net module, respectively. According to the respective characteristics of the above three modules, the residual blocks in different positions in the backbone of YOLOv5 were replaced accordingly. The improved modules explicitly enhance square convolutions with horizontal and vertical asymmetric convolutions. The addition of the multilayer convolution outputs together also make the extracted features more robust. (2) Since the number of channels of the input image will be expanded multiple times after passing through the Focus module, the interdependence between channels is more complicated at this time. Hence, the IECA channel attention module was added. It helps the detection model focus more on the target's position, suppress irrelevant details, and extract more discriminative features. (3) The SPP module was replaced with GSPP. The GSPP module uses grouped convolutions to reduce the number of parameters, increasing model efficiency and reducing the risk of overfitting. (4) Use the K-Means++ algorithm to get more accurate anchor boxes. This algorithm effectively alleviates the problem of influence on convergence caused by the random selection of initial points. This helps to choose better initial cluster centers. (5) Use EIOU as the judgment basis for NMS. It not only considers the coincidence of the two prediction boxes and the distance between the center points, but also the difference in width and height. These features help improve the postprocessing capabilities of the model.

Compared with YOLOD [48], another recently proposed target-detection method suitable for UAV aerial images, YOLO-UAV performs better in regard to detection accuracy and running speed. YOLOD adds a total of four IECA modules at various positions in the backbone and three ASFF modules at the end of the neck. Although the detection accuracy is improved, it undoubtedly increases the computational cost and slows down the operation speed. The location added by the attention mechanism in YOLO-UAV is more targeted. The asymmetric convolution it uses only slightly increases the number of parameters, but it significantly improves the feature extraction capability. The multiple convolutional structures used in the backbone enrich the extracted features and expand the

receptive field of the model. The number of parameters for YOLO-UAV remains in a good range in the end.

5. Conclusions

This research analyzed the shortcomings of the detection method for UAV aerial images based on YOLO. According to the characteristics of UAV aerial images, we made some improvements on the basis of YOLOv5. The detection performance of the model is improved by the modification of the backbone and optimization of other parts. In the production of the UAV aerial image dataset, the previous Forklift dataset was expanded, and some images that were similar to natural images were replaced. We ran a series of experiments on five datasets, namely CIFAR-10, PASCAL VOC, VEDAI, VisDrone 2019, and Forklift. To verify the effectiveness of the improved strategies, ablation experiments were performed on image classification and detection tasks, respectively. The experimental results show that the improved model not only increases detection accuracy but also keeps total parameters and computational complexity at a reasonable level. The superiority of the proposed method is verified by comparison with other advanced detection methods. The experimental results from the tests on the UAV aerial images show that the proposed method still gives a better detection performance despite the challenges of small targets, dense arrangements, sparse distributions, and complex backgrounds. It is suitable for target detection in UAV aerial images. In the final experiment, different NMS methods were compared. The experimental results from the tests on the multiple datasets demonstrate that the proposed EIOU-NMS method is more effective in suppressing redundant prediction boxes.

We will continue to focus on the characteristics of targets in UAV aerial images in the future and propose more targeted optimization strategies. In terms of image collection and dataset annotation, more new target types will be involved.

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

Funding: This research was funded by the National Nature Science Founding of China grant number 61573183.

Acknowledgments: This research was funded by Wuxi Gewu Intelligent Technology Co., Ltd. Thanks to the equipment and personnel support provided by Wuxi Gewu Intelligent Technology Co., Ltd.

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

References

1. Osco, L.P.; de Arruda, M.D.; Goncalves, D.N.; Dias, A.; Batistoti, J.; de Souza, M.; Gomes, F.D.G.; Ramos, A.P.M.; Jorge, L.A.D.; Liesenberg, V.; et al. A CNN approach to simultaneously count plants and detect plantation-rows from UAV imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *174*, 1–17. [CrossRef]
2. Sivakumar, A.N.V.; Li, J.T.; Scott, S.; Psota, E.; Jhala, A.J.; Luck, J.D.; Shi, Y.Y. Comparison of Object Detection and Patch-Based Classification Deep Learning Models on Mid- to Late-Season Weed Detection in UAV Imagery. *Remote Sens.* **2020**, *12*, 2136. [CrossRef]
3. Wang, L.; Xiang, L.R.; Tang, L.; Jiang, H.Y. A Convolutional Neural Network-Based Method for Corn Stand Counting in the Field. *Sensors* **2021**, *21*, 507. [CrossRef] [PubMed]
4. Wu, J.T.; Yang, G.J.; Yang, H.; Zhu, Y.H.; Li, Z.H.; Lei, L.; Zhao, C.J. Extracting apple tree crown information from remote imagery using deep learning. *Comput. Electron. Agric.* **2020**, *174*, 105504. [CrossRef]
5. Ammour, N.; Alhichri, H.; Bazi, Y.; Benjdira, B.; Alajlan, N.; Zuair, M. Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sens.* **2017**, *9*, 312. [CrossRef]
6. Liu, Y.; Shi, G.; Li, Y.; Zhao, Z. M-YOLO based Detection and Recognition of Highway Surface Oil Filling with Unmanned aerial vehicle. In Proceedings of the 7th International Conference on Intelligent Computing and Signal Processing, ICSP 2022, Xi'an, China, 15–17 April 2022; pp. 1884–1887.

7. Ding, W.; Zhang, L. Building Detection in Remote Sensing Image Based on Improved YOLOV5. In Proceedings of the 17th International Conference on Computational Intelligence and Security, CIS 2021, Chengdu, China, 19–22 November 2021; pp. 133–136.
8. Zhang, R.; Wen, C.B. SOD-YOLO: A Small Target Defect Detection Algorithm for Wind Turbine Blades Based on Improved YOLOv5. *Adv. Theory Simul.* **2022**, *5*, 2100631. [CrossRef]
9. Guo, J.; Xie, J.; Yuan, J.; Jiang, Y.; Lu, S. Fault Identification of Transmission Line Shockproof Hammer Based on Improved YOLO V4. In Proceedings of the 2021 International Conference on Intelligent Computing, Automation and Applications (ICAA), Nanjing, China, 25–27 June 2021; pp. 826–833.
10. Liu, C.Y.; Wu, Y.Q.; Liu, J.J.; Han, J.M. MTI-YOLO: A Light-Weight and Real-Time Deep Neural Network for Insulator Detection in Complex Aerial Images. *Energies* **2021**, *14*, 1426. [CrossRef]
11. Sambolek, S.; Ivasic-Kos, M. Automatic Person Detection in Search and Rescue Operations Using Deep CNN Detectors. *IEEE Access* **2021**, *9*, 37905–37922. [CrossRef]
12. Bozic-Stulic, D.; Marusic, Z.; Gotovac, S. Deep Learning Approach in Aerial Imagery for Supporting Land Search and Rescue Missions. *Int. J. Comput. Vis.* **2019**, *127*, 1256–1278. [CrossRef]
13. de Oliveira, D.C.; Wehrmeister, M.A. Using Deep Learning and Low-Cost RGB and Thermal Cameras to Detect Pedestrians in Aerial Images Captured by Multirotor UAV. *Sensors* **2018**, *18*, 2244. [CrossRef]
14. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
15. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
16. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1627–1645. [CrossRef] [PubMed]
17. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [CrossRef]
18. Papageorgiou, C.; Poggio, T. A trainable system for object detection. *Int. J. Comput. Vis.* **2000**, *38*, 15–33. [CrossRef]
19. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
20. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1440–1448.
21. Ren, S.Q.; He, K.M.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015.
22. He, K.M.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
23. Cai, Z.W.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
24. Dai, J.F.; Li, Y.; He, K.M.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016.
25. Huang, L.; Yang, Y.; Deng, Y.; Yu, Y.J. DenseBox: Unifying Landmark Localization with End to End Object Detection. *arXiv* **2015**, arXiv:1509.04874.
26. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.M.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007.
27. 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 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
28. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
29. Redmon, J.; Farhadi, A.J. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M.J. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
31. 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. [CrossRef]
32. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J.J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
33. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 779–788.
34. Sahin, O.; Ozer, S. YOLODrone: Improved YOLO Architecture for Object Detection in Drone Images. In Proceedings of the 44th International Conference on Telecommunications and Signal Processing (TSP), Virtual, 26–28 July 2021; pp. 361–365.
35. Junos, M.H.; Khairuddin, A.S.M.; Thannirmalai, S.; Dahari, M. Automatic detection of oil palm fruits from UAV images using an improved YOLO model. *Vis. Comput.* **2022**, *38*, 2341–2355. [CrossRef]
36. Cheng, Y. Detection of Power Line Insulator Based on Enhanced YOLO Model. In Proceedings of the 2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers, IPEC 2022, Dalian, China, 14–16 April 2022; pp. 626–632.

37. Wang, X.W.; Zhao, Q.Z.; Jiang, P.; Zheng, Y.C.; Yuan, L.M.Z.; Yuan, P.L. LDS-YOLO: A lightweight small object detection method for dead trees from shelter forest. *Comput. Electron. Agric.* **2022**, *198*, 107035. [CrossRef]
38. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef] [PubMed]
39. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 770–778.
40. Liu, S.; Qi, L.; Qin, H.F.; Shi, J.P.; Jia, J.Y. Path Aggregation Network for Instance Segmentation. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
41. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.M.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
42. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
43. Chen, Y.P.; Li, J.N.; Xiao, H.X.; Jin, X.J.; Yan, S.C.; Feng, J.S. Dual Path Networks. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017.
44. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2Net: A New Multi-Scale Backbone Architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 652–662. [CrossRef] [PubMed]
45. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 2818–2826.
46. Ding, X.H.; Guo, Y.C.; Ding, G.G.; Han, J.G. ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 1911–1920.
47. Shen, J.; Jiang, L.P. Correlation Analysis Between Japanese Literature and Psychotherapy Based on Diagnostic Equation Algorithm. *Front. Psychol.* **2022**, *13*, 906952. [CrossRef] [PubMed]
48. Luo, X.D.; Wu, Y.Q.; Zhao, L.Y. YOLOD: A Target Detection Method for UAV Aerial Imagery. *Remote Sens.* **2022**, *14*, 3240. [CrossRef]
49. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
50. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In Proceedings of the 13th European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 346–361.
51. Gao, G.H.; Wang, S.Y.; Shuai, C.Y.; Zhang, Z.H.; Zhang, S.; Feng, Y.B. Recognition and Detection of Greenhouse Tomatoes in Complex Environment. *Traitement Du Signal* **2022**, *39*, 291–298. [CrossRef]
52. Wu, Z.S.; Xue, R.; Li, H. Real-Time Video Fire Detection via Modified YOLOv5 Network Model. *Fire Technol.* **2022**, *58*, 2377–2403. [CrossRef]
53. Arthur, D.; Vassilvitskii, S. K-means++: The Advantages of Careful Seeding. In Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
54. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S.J. Soft-NMS—Improving Object Detection With One Line of Code. *arXiv* **2017**, arXiv:1704.04503.
55. Zheng, Z.H.; Wang, P.; Liu, W.; Li, J.Z.; Ye, R.G.; Ren, D.W. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 12993–13000.
56. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T.J. Focal and Efficient IOU Loss for Accurate Bounding Box Regression. *arXiv* **2021**, arXiv:2101.08158. [CrossRef]
57. Krizhevsky, A.; Hinton, G. Learning multiple layers of features from tiny images. Master's Thesis, University of Toronto, Toronto, ON, Canada, 2009.
58. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent.* **2016**, *34*, 187–203. [CrossRef]
59. Du, D.W.; Zhu, P.F.; Wen, L.Y.; Bian, X.; Ling, H.B.; Hu, Q.H.; Peng, T.; Zheng, J.Y.; Wang, X.Y.; Zhang, Y.; et al. VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 213–226.
60. 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, 13–19 June 2020; pp. 10781–10790.