DB-YOLOv5: A UAV Object Detection Model Based on Dual Backbone Network for Security Surveillance

Yuzhao Liu 1, Wan Li 1,*, Li Tan 1,2, Xiaokai Huang 1, Hongtao Zhang 1 and Xujie Jiang 1

1 School of Computer Science and Engineering, Beijing Technology and Business University, Beijing 100048, China; jsjlyz21@st.btbu.edu.cn (Y.L.); tanli@th.btbu.edu.cn (L.T.); huangxk@st.btbu.edu.cn (X.H.); zhanghongtao@st.btbu.edu.cn (H.Z.); myishailong@gmail.com (X.J.)
2 Chongqing Institute of Microelectronics Industry Technology, University of Electronic Science and Technology of China, Chongqing 400031, China
* Correspondence: wanli@btbu.edu.cn

Abstract: Unmanned aerial vehicle (UAV) object detection technology is widely used in security surveillance applications, allowing for real-time collection and analysis of image data from camera equipment carried by a UAV to determine the category and location of all targets in the collected images. However, small-scale targets can be difficult to detect and can compromise the effectiveness of security surveillance. In this work, we propose a novel dual-backbone network detection method (DB-YOLOv5) that uses multiple composite backbone networks to enhance the extraction capability of small-scale targets’ features and improve the accuracy of the object detection model. We introduce a bi-directional feature pyramid network for multi-scale feature learning and a spatial pyramidal attention mechanism to enhance the network’s ability to detect small-scale targets during the object detection process. Experimental results on the challenging UAV aerial photography dataset VisDrone-DET demonstrate the effectiveness of our proposed method, with a 3% improvement over the benchmark model. Our approach can enhance security surveillance in UAV object detection, providing a valuable tool for monitoring and protecting critical infrastructure.

Keywords: object detection; UAV; security surveillance; feature pyramid network; attention mechanism

1. Introduction

The use of drones for remotely detecting and tracking persons or vehicles has become increasingly prevalent in the field of security surveillance, particularly in urban areas. However, the widespread use of drones also raises concerns about network security and privacy. Due to their small size and limited power supply performance, UAVs have low computing power, which poses significant challenges for accurate object detection tasks [1]. In 2012, the introduction of the deep convolutional neural network (CNN) by Krizhevsky et al. [2] revolutionized the field of computer vision, leading to the development of more efficient and accurate object detection models, such as the RCNN model proposed by Girshick et al. [3] in 2015. Since then, deep learning-based object detection technology has undergone rapid development, providing significant potential for enhancing security surveillance capabilities while also requiring careful consideration of network security and privacy concerns. Recent studies such as [4] have focused on developing AI-driven solutions to address network security and privacy challenges in the context of UAV object detection.

Currently, object detection methods based on deep learning can be mainly divided into two-stage and one-stage categories. Among them, the two-stage method is based on target candidate regions for detection. This method extracts candidate regions and performs deep learning on the corresponding regions, which has high classification accuracy of detection results. This type of algorithm mainly includes a series of methods such as RCNN [3], Fast R-CNN [5], Faster R-CNN [6], R-FCN [7], CoupleNet [8], Mask RCNN [9], Cascade...
RCNN [10], Libra R-CNN [11], Reasoning-RCNN [12], EfficientDet [13], D2Det [14], DeFeat [15], MViTv2 [16], and AdaMixer [17]. Otherwise, the one-stage method can directly calculate the category probability and position coordinate value of the object through regression, including SSD [18], DSSD [19], FSSD [20], ScratchDet [21], ExtremeNet [22], MimicDet [23], E3Net [24], SaFT [25], CapDet [26], YOLO [27], YOLOv2 [28], YOLOv3 [29], YOLOv4 [30], YOLOv5, YOLOx [31], YOLOF [32], YOLOv6 [33], YOLOv7 [34], YOLOv8, and other methods. The detection result is directly obtained after single detection, which greatly improves the speed of the model. For example, the YOLOv5 model improves the accuracy and speed of YOLOv4. The training phase of the model includes improvements such as mosaic data enhancement, adaptive image scaling, and adaptive anchor box calculation.

With the successive birth of various object detection models with superior performance and improved experimental results on the MS COCO dataset, object detection for UAV has also attracted increasing attention as an important computer vision task. UAV object detection plays an important role in understanding long-range images. Unlike general object detection, the aerial photography angles of UAVs all look down, which leads to a small scale and many targets in the image. Therefore, object detection models that are suitable for general data sets cannot achieve high robustness in UAV aerial photography. As shown in Figure 1, when the YOLOv5 model for object detection is employed in the VisDrone UAV dataset, pedestrian targets with small scales are more likely to be missed compared to vehicle targets with larger scales, which leads to weak generalization ability of the UAV object detection model.

![Figure 1](image_url)

**Figure 1.** After the YOLOv5 model performs object detection on the VisDrone dataset, there is a phenomenon of both missed detection and false positives. The figure shows the detected targets marked with positioning boxes of different colors, but many vehicles and pedestrians are still not detected or are incorrectly identified.

In general, small-scale targets can be defined as the size range of objects in an image that are smaller than the minimum detectable size threshold of the object detection model. One way to define this threshold is to use the concept of object coverage ratio, which refers to the proportion of the image area covered by an object. For example, a commonly used threshold for object detection is 0.005, which means that an object must occupy at least 0.5% of the image area to be detected by the model. In this case, too-small scales would refer to objects that are smaller than the minimum size required to achieve an object coverage ratio of 0.005.
Based on the above discussion, our paper proposes a DB-YOLOv5 object detection model that is suitable for UAV, aiming at the problem of too-small targets in UAV. The model adopts a composite backbone network, which connects multiple identical backbone networks in a composite manner and fuses high-level and low-level features of multiple backbone, which expands the receptive field of the network. The bi-directional feature pyramid network structure is also introduced in the feature extraction stage, which can fuse multi-scale features conveniently, quickly, and effectively to improve the detection accuracy of small-scale targets. The spatial pyramid attention mechanism is used in the output stage, which can maintain the feature representation and spatial location information of the target and further strengthen the ability to identify and locate small targets. Finally, ElIoU_loss is used to further optimize the bounding box of the small-scale target to improve the bounding box problem in small target detection.

The main contributions of this paper are as follows:

1. The dual-backbone network model DB-YOLOv5 is proposed. Aiming at the problem of missed and false detection of small targets in UAV images, the model integrates the high-level and low-level features of multiple identical backbone networks to expand the receptive field of the network for small target features.
2. The model uses a bi-directional feature pyramid network, which fuses multi-scale features through the bottom-up and top-down approaches to strengthen the network’s multi-scale feature fusion for small targets.
3. The spatial pyramid attention mechanism enables the model to maintain both the feature information and the location information of small targets, which strengthens their identification and positioning.

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 details the proposed method. The experiments and results analyses are provided in Section 4 while introducing the selected dataset and evaluation indicators. Finally, conclusions are drawn in Section 5.

2. Related Work

With the increasing demand for security surveillance, object detection models are being used more frequently in aerial remote sensing to detect potential security threats. However, one of the major challenges in this field is detecting small targets in aerial images. To address this problem, researchers have proposed various multi-angle improvements to different general object detection models with the aim of enhancing the robustness and generalization ability of the detection models for aerial photography datasets in the context of network security and privacy. For example, Cheng et al. [35] proposed a concept of coarse-grained density maps for the problem of dense small objects and uneven distribution in aerial remote sensing images and designed a density map-based clustering region generation algorithm. They improved the Mosaic data augmentation method to divide the image into multiple sub-regions so that dense small objects could be adjusted to a reasonable scale. This method improved the detection performance of rare objects and difficult samples and alleviated the foreground–background and class imbalances. Huang et al. [36] designed a unified foreground assembly and multi-agent detection network for the problem of dense small targets and target shape similarity in aerial images. The method combines the subregions provided by the coarse detector to suppress the background by clustering, and this method then assembles the resulting subregions into mosaics for a single inference. The method models the object distribution in a fine-grained manner by using multi-agent learning, thereby significantly reducing the overall time cost and improving the efficiency and accuracy of detection. Wang et al. [37] proposed a model based on multiple center points to solve the problem of small target detection in images. The method first located multiple center points and then estimated the offset and scale of multiple corresponding targets, which can improve the detection performance of small targets. Xu et al. [38] aimed at the detection of small targets in aerial images and believed that IoU, as the most commonly used indicator in object detection
tasks, was not suitable for small targets. They proposed a simple and effective dot distance method, which was defined as the normalized Euclidean distance between the center points of the two bounding boxes. This method was suitable for small target detection and achieved better detection performance. Tan et al. [39] proposed the YOLOv4_Drone method based on YOLOv4, aiming at the problems of small targets, complex background, and mutual occlusions of targets in UAV images. This method employed the concept of hole convolution. It introduced an ultra-lightweight subspace attention mechanism and soft-nms to resample the same feature map, which implemented multi-scale feature representation, to solve the problem of missed detection caused by adjacent or even occluded targets captured by drones. In order to improve the precision of UAV object detection while satisfying the lightweight feature, Yang et al. [40] modified the YOLOv5s model. To address the small object detection problem, a prediction head is added to better retain small object feature information. The CBAM attention module is also integrated to better find attention regions in dense scenes. The original IOU-NMS is replaced by NWD-NMS in post-processing to alleviate the sensitivity of IOU to small objects.

Although the above algorithms have carried out various studies and explorations on the problem of small targets in UAV images, the proposed models still cannot obtain real-time and efficient detection results of small targets in practical UAV applications. Therefore, based on the YOLOv5 algorithm, this paper introduces a composite backbone network, a bidirectional feature pyramid network structure, and a spatial pyramid attention mechanism and proposes the UAV image object detection model DB-YOLOv5.

The YOLOv5 model has excellent speed and accuracy in the detection algorithm due to using the CSPDarknet53 backbone network to extract features. The CSPDarknet53 is based on the Darknet53 in YOLOv3, which combines the CSPNet [41] to develop the backbone structure. This network contains five CSP modules, which are composed of convolution kernels with a size of $3 \times 3$ and a stride of 2, so it can play a role in downsampling. Thus, the model adopts the CSPDarknet53 backbone network, which can enhance the feature learning ability of network, maintain the model accuracy while remaining lightweight, and reduce the computational cost of the model. In addition, the input of the model adopts the mosaic method for data enhancement, which splices four pictures through random scaling, cropping, and arrangement. This method is highly effective for small target detection. In the Neck stage of YOLOv5, the PANet [42] structure of FPN + PAN [43] is adopted, with which the model can strengthen the multi-scale feature fusion ability, accurately save the location information of small targets, and contribute to locating the target correctly. Although the DB-YOLOv5 model focuses on improving object detection in low-altitude aerial images, there are also several studies addressing other challenges in video surveillance. Sun et al. [44] proposed a dynamic partial-parallel data layout (DPPDL) for green video surveillance storage, which aims to reduce energy consumption and improve storage efficiency. Similarly, Yu et al. [45] introduced an extra-parity energy-saving data layout for video surveillance, which reduces energy consumption and optimizes storage utilization. These studies demonstrate the importance of developing efficient and sustainable solutions for video surveillance, which can have significant implications for various applications, such as security and public safety. Zhang et al. [46] conducted research on backdoor attacks on deep neural network models used in image classification. They analyzed the impact of these attacks on classification accuracy and proposed a defense mechanism to mitigate them. This study emphasizes the importance of developing secure and robust deep neural network models for reliable image classification.

In low-altitude aerial images, the visual information contained in tiny targets is limited by the condition of looking down, which results in significant difficulties in aerial target detection. Therefore, improving the detection performance of small and ambiguous targets and reducing the occurrence of missed and false detections is an urgent problem for UAV image object detection. This paper is dedicated to providing an effective solution to this purpose, namely, the UAV image object detection model DB-YOLOv5.
3. Methods

3.1. Overall Structure

The DB-YOLOv5, which is the UAV object detection model proposed, improves the capabilities of feature extraction and fusion by introducing a composite backbone network, a bidirectional feature pyramid network, and a spatial pyramid attention mechanism. The problem of false and missed detection because the scale of targets in the UAV environment is too small can be solved by this model. Thus, the model can improve the accuracy of small targets. The structure of DB-YOLOv5 is shown in Figure 2.

![Figure 2](image_url)

Figure 2. The overall architecture of the DB-YOLOv5 object detection model. The green module represents the image feature obtained through the model network; the number above the module represents the channel number of the feature.

The model performs data enhancement using the Mosaic operation in the image preprocessing stage, scales the input UAV image to the prescribed input size of 640 × 640 for this network model, and performs data preprocessing operations such as normalization. After that, Focus slicing and convolution are performed on the input image to obtain 320 × 320 × 32 feature maps. In the N-layer assisting backbone network (yellow part in Figure 1), the H × W × C feature map of layer N − 1 can be obtained after 3 × 3 convolution and normalization operations to obtain the H/2 × W/2 × 2C N-layer feature map. To clarify, in the N-layer main backbone network (green part in Figure 1), the process to obtain the Nth-layer main backbone network (green part in Figure 1), the H × W × C feature map involves superimposing and fusing the 2H × 2W × C/2 feature maps of the N − 1st-layer and the Nth-layer feature maps in the assisting backbone network. This process is shown in Equation (1), where $x_{main}^{N}$ and $x_{assist}^{N−1}$ denote the feature maps of the main backbone network at layer $N$ and layer $N−1$, respectively; $x_{assist}^{N}$ denotes the feature map of the assisting backbone network at layer $N$; and $UP(\cdot)$ denotes the UpSampling operation. The value of $N$ used in this paper is 5.

$$x_{main}^{N} = x_{main}^{N−1} \oplus UP(x_{assist}^{N})$$  \hspace{1cm} (1)$$

where n is the feature fusion stage with $N'$ output scales, and the dimension $H \times W \times C$ of the $N'$th output feature map is obtained by fusing the $N'$ − 1st output feature map with a scale of $2H \times 2W \times C/2$ and the feature maps of the same $H \times W \times C$ dimensions in the shallow network processed by the bidirectional feature pyramid network, as shown in Equation (2), where $x_{fpm}^{N'}$ and $x_{fpm}^{N'−1}$ denote the $N'$th and $N'−1$st feature fusion output, respectively; $x_{backward}^{N'}$ denotes the feature map of the same size as the $N'$th output in the shallow network; and $Bi(\cdot)$ denotes the bidirectional feature pyramid network. In this paper, the value of $N'$ is taken as 3.

$$x_{fpm}^{N'} = x_{fpm}^{N'−1} \oplus Bi(x_{backward}^{N'})$$  \hspace{1cm} (2)$$
The feature maps after feature extraction and multiscale fusion are adaptively averaged pooled at three scales of $80 \times 80$, $40 \times 40$, and $20 \times 20$ through a spatial pyramid structure to generate an attention map. The generated attention maps are weighted by a combination of a fully connected layer and a sigmoid activation layer to generate attention weights in the corresponding feature maps, which label the small targets in the original images more accurately.

3.2. Composite Backbone Network Based on CSPDarknet53

The backbone of the YOLOv5 model adopts the CSPDarknet53 network combined with the CSP structure. However, the feature extraction ability of this network for small-scale targets cannot achieve satisfactory results. Most of the current research on the backbone network focuses on deepening or widening the backbone. To deepen the network of the model without introducing additional pre-training overhead, we introduce the structure of the composite backbone network (CBNet) in the backbone stage of DB-YOLOv5. The structure aims to superimpose multiple layers of the same type of backbone to expand the feature receptive field of the network, thereby enhancing the capability of the backbone’s feature extraction for the small target in UAV.

CBNet is divided into two types: the main backbone and the assisting backbone. The purpose of using the assisting backbone is to complement the features extracted by the main backbone. Each backbone has $L$ stages, which contain a series of layers of convolution and have the same size of feature maps. The nonlinear transformation implemented in the $l$th stage is defined as $F^l$. The output of the $l$th stage of the assisting backbone (denoted as $x^l_{\text{assist}}$) is fused with the output of the $(l-1)$st stage of the main backbone ($x^{l-1}_{\text{main}}$), which is the input in the parallel stage ($l$) of the main backbone, as shown in Equation (3):

$$x^l_{\text{main}} = F^l_{\text{main}}\left(x^{l-1}_{\text{main}} + g\left(x^l_{\text{assist}}\right)\right), \quad l \geq 2$$

(3)

where $g(\cdot)$ represents a layer of $1 \times 1$ convolution and a layer of batch normalization, whose purpose is to concatenate the features of the main and assisting backbones.

As shown in Figure 3, after the $80 \times 80$ feature image obtained by convolution in the assisting CSPDarknet53 backbone is input to the main backbone, it is superimposed and fused with the feature image obtained after the $640 \times 640$ original image is processed by Focus slicing, and the obtained result is input as the input content to the starting position of the main backbone. The $40 \times 40$ feature image obtained by convolution processing in the assisting backbone model and the $80 \times 80$ feature image in the main backbone model are then superimposed and fused, and the result is used as the input to continue the next convolution process. Finally, the $20 \times 20$ feature image output by the assisting backbone is superimposed and fused with the $40 \times 40$ feature image in the main backbone model, and the result is input to the main backbone model to continue the convolution operation. Thus, the obtained feature image is passed to the next module. After that, the feature information extracted from the backbone network is used as the input in Section 3.3 to perform multi-scale stacking and processing of features through the bidirectional feature pyramid network. Using the features learned by the network, combined with the spatial pyramid attention mechanism in Section 3.4, the objects in the input images of the model are classified and localized.

Meanwhile, to further enhance the operating efficiency and cut down on time cost, we no longer connect the low-level features of the first two layers in the composite backbone network module and only connect and stack the features of the last two layers of backbone networks. The high-level semantic feature information is further saved and learned while retaining lower-level location information, thereby easing the contradiction between time and accuracy to a certain extent. We named this module CBNet-tiny, as shown in Figure 4.
Meanwhile, to further enhance the operating efficiency and cut down on time cost, we no longer connect the low-level features of the first two layers in the composite backbone network module and only connect and stack the features of the last two layers of backbone networks. The high-level semantic feature information is further saved and learned while retaining lower-level location information, thereby easing the contradiction between time and accuracy to a certain extent. We named this module CBNet-tiny, as shown in Figure 4.

3.3. Bidirectional Feature Pyramid Network

The feature information extracted by the composite backbone network in Section 3.2 is also deviated in sensitivity to small targets according to the different feature extraction scales. In order to coordinate the feature information extracted from different scales, the model needs to combine the extracted features for multi-scale fusion and learning—that is, to take methods to further represent and process multi-scale features effectively, which is also one of the difficulties in target detection. Early detection models usually make predictions through a pyramid structure directly, which is based on features extracted from the backbone. In this process, the feature pyramid network plays an important role, proposing the idea of combining multi-scale features in a top-down manner. Inspired by this idea, PANet, based on FPN, adds a bottom-to-top path to further aggregate the feature

Figure 4. Diagram of the CBNet-tiny module’s structure.
information. However, it also consumes a lot of time, especially in the training phase, but achieves good performance. Since the contribution of nodes—only one input edge—to the fusion feature network is small, the Bi-FPN removes the intermediate nodes of P3 and P7 in PANet to form a simplified bidirectional network to reduce model overhead. Additionally, this module adds a skip connection between input nodes to output nodes at the same scale, incorporating more features without increasing excessive computational overhead. At the same time, the model regards the Bi-FPN module, which achieves feature fusion through a bidirectional path, as a network layer and reuses it many times to achieve better feature fusion, as shown in Figure 5.

![Figure 5. Comparison among different feature extraction modules. (a) Top-down FPN structure; (b) top-down and bottom-up PANET structure; (c) BiFPN structure with bidirectional path.](image)

In the bidirectional feature pyramid network, the output results of each node are shown in Equations (4)–(6):

$$x_{f_{pn}}^{l} = F_{conv}(x_{1}^{l} + F_{conv}(x_{1}^{l+1})).$$  \hspace{1cm} (4)

$$x_{f_{pn}}^{l} = F_{conv}(x_{1}^{l} + F_{conv}(x_{1}^{l+1}) + x_{f_{pn}}^{l-1}).$$ \hspace{1cm} (5)

$$x_{f_{pn}}^{l} = F_{conv}(x_{1}^{l} + x_{f_{pn}}^{l-1}).$$ \hspace{1cm} (6)

where $F_{conv}(\cdot)$ represents the convolutional layer, $x_{1}^{l}$ represents the input features of the bidirectional pyramid network, and $x_{f_{pn}}^{l}$ represents the output features. Equation (4) represents the output result of node P3 in Figure 5, Equation (5) represents the output result of node P7 in Figure 5, and Equation (6) represents the output result of the intermediate node in Figure 5. Although the PANet structure of YOLOv5 can fuse multi-scale target feature information, it can easily cause missing features in the process of feature fusion between small and large targets, thus affecting the ability of the model to detect small-scale targets. To address this, we use the Bi-FPN module four times in DB-YOLOv5 to bi-directionally fuse multi-scale feature information multiple times on the three output branches to further strengthen the model’s feature fusion and extraction capabilities for small targets and ensure the accuracy of positioning and classification of small targets in the model.

3.4. Spatial Pyramid Attention Mechanism

Although the model can strengthen the feature fusion and extraction capabilities of small objects through the bidirectional feature pyramid network described in Section 3.3, it may still be challenging to fully grasp the mechanism involved. However, in practical applications, the model also needs to be able to ignore the complex background and other interfering information features in the image and identify the desired feature information.
of the small target. To achieve accurate classification and positioning to avoid the problem of missed detection and false detection in the detection of small targets in the drone environment, we use the spatial pyramid attention network (SPANet) in DB-YOLOv5. The introduction of an attention mechanism makes our model focus on the small target part in the image and selectively extract key information from images while ignoring the interference of irrelevant information, such as the background. This improves the localization and classification performance of the entire model for small-scale targets and the accuracy of the detection model.

The feature processing process of the spatial pyramid attention mechanism is shown in Equation (7):

$$x_{weight} = \text{sigmoid} \left( F_{fc} \left( P_{1\times1}(x) + P_{2\times2}(x) + P_{4\times4}(x) \right) \right)$$

Among them, $F_{fc}(\cdot)$ represents the fully connected layer; $\text{sigmoid}(\cdot)$ represents the activation function layer; $P_{1\times1}(\cdot)$, $P_{2\times2}(\cdot)$, and $P_{4\times4}(\cdot)$ represent the $1 \times 1$, $2 \times 2$, and $4 \times 4$ adaptive average pooling layers, respectively; and $x_{weight}$ represents the output weight. The spatial pyramid attention mechanism locates the information of interest by using the structure of the spatial pyramid instead of global average pooling, which consists of two parts. As shown in Figure 6, the input feature maps are passed through a spatial pyramid structure, which is adaptively average pooled at three scales, to generate attention maps. Among them, the purpose of the $1 \times 1$ adaptive average pooling layer is to obtain the key information of the category in the feature map, the $2 \times 2$ pooling layer is used to save the less important key feature information in the image, and the $4 \times 4$ average pooling can effectively obtain the key position information in the feature map. Afterwards, the generated attention map is passed through a weight module, which is composed of a fully connected layer and a sigmoid activation layer, to generate the attention weights in the corresponding feature map. Thus, through the attention weight output by the attention module, the small objects in the original image are more accurately marked.

![Figure 6. The structure of the spatial pyramid attention mechanism.](image-url)

4. Experiments

4.1. Datasets

The data selected for the experiments were from the VisDrone-DET dataset. This dataset was collected by the AISKYEYE team at the Machine Learning and Data Mining Laboratory of Tianjin University, China. The dataset consisted of 10,209 still images captured by various drone-mounted cameras, including different locations (taken from 14 different cities that were thousands of kilometers apart in China), different environments (urban and rural), different objects (pedestrians, vehicles, bicycles, etc.), and different densities (sparse and crowded scenes). There were mainly 10 categories of objects. The number of samples for each category is shown in Figure 7.
The data selected for the experiments were from the VisDrone-DET dataset. This data was collected by the AISKYEYE team at the Machine Learning and Data Mining Laboratory of Tianjin University, China. The dataset consisted of 10,209 still images captured from different locations (taken from 14 different environments (urban and rural), different densities (urban and rural), different dates), and different objects (pedestrians, vehicles, bicycles, etc.), and different locations (taken from 14 different environments (urban and rural)).

We conducted our experiment using 6471 images from VisDrone2019-DET-train as the training set, 548 images from VisDrone2019-DET-val as the validation set, and 1610 images from VisDrone2019-DET-test-dev as the test set.

For DB-YOLOv5, we set the model input image size to $640 \times 640$, the batch size was 16, the confidence threshold was 0.25, and the Intersection over Union threshold was 0.45. The learning rate was initialized to 10$^{-4}$ and halved after every 50% training batch. We implemented our model on the Torch 1.8.0 platform and conducted 300 epochs of training experiments on the training and validation sets on a single NVIDIA GeForce RTX 3070.

### 4.3. Quantitative Experiments

To verify the detection effect of the model proposed in the paper, we compared our model with other models in the field of object detection. The detection results are shown in Table 1.

Table 1. Experimental results of different object detection models on the VisDrone-DET dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
<th>Ped</th>
<th>People</th>
<th>Bicycle</th>
<th>Car</th>
<th>Van</th>
<th>Trunk</th>
<th>Tricycle</th>
<th>Awn.</th>
<th>*</th>
<th>Bus</th>
<th>Motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light-R CNN [48]</td>
<td>16.53</td>
<td>17.02</td>
<td>4.83</td>
<td>5.73</td>
<td>32.39</td>
<td>22.12</td>
<td>18.39</td>
<td>16.63</td>
<td>11.91</td>
<td>29.02</td>
<td>11.93</td>
<td></td>
</tr>
<tr>
<td>FPN [42]</td>
<td>16.51</td>
<td>15.69</td>
<td>5.02</td>
<td>4.93</td>
<td>38.47</td>
<td>20.82</td>
<td>18.82</td>
<td>15.03</td>
<td>15.04</td>
<td>26.72</td>
<td>12.83</td>
<td></td>
</tr>
<tr>
<td>Sparse R-CNN [48]</td>
<td>36.70</td>
<td>26.50</td>
<td>18.40</td>
<td>56.00</td>
<td>35.80</td>
<td>25.40</td>
<td>19.50</td>
<td>12.20</td>
<td>12.20</td>
<td>43.30</td>
<td>26.10</td>
<td></td>
</tr>
<tr>
<td>YOLOv4 [30]</td>
<td>42.44</td>
<td>49.20</td>
<td>24.80</td>
<td>25.90</td>
<td>74.30</td>
<td>62.20</td>
<td>69.60</td>
<td>31.40</td>
<td>30.50</td>
<td>73.40</td>
<td>43.10</td>
<td></td>
</tr>
<tr>
<td>YOLOv5</td>
<td>48.44</td>
<td>49.20</td>
<td>24.80</td>
<td>25.90</td>
<td>74.30</td>
<td>62.20</td>
<td>69.60</td>
<td>31.40</td>
<td>30.50</td>
<td>73.40</td>
<td>43.10</td>
<td></td>
</tr>
<tr>
<td>DB-YOLOv5</td>
<td>51.53</td>
<td>52.70</td>
<td>28.10</td>
<td>28.40</td>
<td>77.10</td>
<td>72.30</td>
<td>65.40</td>
<td>33.80</td>
<td>33.90</td>
<td>76.60</td>
<td>47.00</td>
<td></td>
</tr>
</tbody>
</table>

* The meaning of awn. is awning-tricycle.

It can be seen from the mean average precision (mAP) results in Table 1 that, compared with the anchor-based method adopted by DB-YOLOv5, the anchor-free detection model CornerNet [47] was not suitable for UAV detection. At the same time, the comparison of our model with FPN [42] also verifies that the BiFPN structure of the bidirectional path combined with the composite backbone network method could achieve more excellent results than basic FPN in UAV object detection. The comparison of Cascade RCNN [10] and Sparse R-CNN [48] shows that the one-stage method outperformed the two-stage method for the detection of small targets of UAVs. Obviously, compared with YOLO series models such as YOLOv4 [30] and YOLOv4_Drone [39], our model improved the overall detection performance and the performance of various categories. For example, in the three categories of “pedestrian”, “people”, and “motor”, our model obtained an improvement of...
346% and 292%, 244% and 216%, and 213% and 180%, respectively, which verifies the high accuracy of our model for the detection of small objects like pedestrians and motorcycles. However, it can also be observed that our model had lower accuracy on the three target categories of “trunk”, “tricycle”, and “awning-tricycle”. After analysis, it is believed that the detection ability of small targets with similar semantics will be enhanced after the detection ability of our model for small targets is further improved. Due to the semantic similarity between tricycles and bicycles, trunks and vans, and awning-tricycles and cars, it is a significant challenge to distinguish them through the model during the learning process. Through the above experiments, the effectiveness of the improvement idea of our proposed target detection method applicable to small targets of UAVs can be seen, so we also hope that the improvement method proposed in this thesis can be applied to the same kind of YOLO algorithm and make performance breakthroughs on these versions as well, which is the work we will continue to study in depth in the future.

4.4. Qualitative Experiments

The detection results of the DB-YOLOv5 model are visualized in Figure 8. The figure includes the detection results under various conditions of insufficient light, sufficient light, dark, blurred image, and top-down angle. We can see that our method could better detect small and dense objects, especially in the central region. The target in the image is marked by a bounding box, whose color was randomly generated, and the same category in an image is marked with the same color, whereas different categories are marked with different colors.

Figure 8. The detection effect of the DB-YOLOv5 model in different scenarios. The images on the left are the original images in the dataset, and the images on the right are the output images of the model. (a) The detection effect under insufficient light; (b,c) the detection results when it is dark; (d,g) the detection results when the light is sufficient; (e) the detection effect when the light is sufficient but the image is blurred; (f,h) the detection effect diagram from the top-down angle when the light is sufficient; (i,j) the detection results from the top-down angle when it is dark.
4.5. Ablation Experiment

4.5.1. Influence of the Number of Backbone Networks

In the composite backbone network module of DB-YOLOv5, the visualization of the experimental results shows that the effect of using two backbone networks was better than three backbone networks in this model, as shown in Figure 9. Therefore, in this paper, we connected the two CSPDarknet53 backbone networks through the connection module to strengthen the main backbone with the assisting backbone, thereby improving the capability of feature extraction in the backbone.

Figure 9. Schematic diagram of the detection results of two backbone networks (a) and three backbone networks (b) in the composite backbone network. From the comparison of the two figures, the number of targets detected in (b) is significantly less than in (a), which proves that the effect of the two backbone networks in the composite backbone network is better than that of the three backbone networks.

4.5.2. Influence of Composite Backbone Network on Model Parameters

In the module of CBNet, we improved the ability of the backbone network to extract image features by adding two identical CSPDarknet53 backbone networks. However, the introduction of two backbone networks into the model caused an exponential increase in the parameters of the entire network model. Our experiment demonstrated that our proposed improvements can meet the requirements of high precision and short time consumption for UAV object detection. The parameter comparison after the improved model is shown in Table 2. Based on the above results, although the introduction of a composite backbone network structure in the model led to a significant improvement in parameters, it did not increase by multiples. At the same time, we can see that the floating point operations (FLOPs) computing power of the model was doubled. Therefore, after experimental verification, it can be concluded that the method of adding a composite backbone network structure in the model has feasibility and practical application prospects.

Table 2. Comparison of parameters of different detection models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Size (Pixels)</th>
<th>Params (M)</th>
<th>FLOPs@640 (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv5s</td>
<td>640</td>
<td>7.2</td>
<td>12.6</td>
</tr>
<tr>
<td>DB-YOLOv5s</td>
<td>640</td>
<td>12.9</td>
<td>41.6</td>
</tr>
<tr>
<td>YOLOv5m</td>
<td>640</td>
<td>21.2</td>
<td>49.0</td>
</tr>
<tr>
<td>DB-YOLOv5m</td>
<td>640</td>
<td>32.8</td>
<td>102.6</td>
</tr>
<tr>
<td>YOLOv5l</td>
<td>640</td>
<td>46.5</td>
<td>109.1</td>
</tr>
<tr>
<td>DB-YOLOv5l</td>
<td>640</td>
<td>88.4</td>
<td>220.3</td>
</tr>
<tr>
<td>YOLOv5x</td>
<td>640</td>
<td>86.7</td>
<td>205.7</td>
</tr>
<tr>
<td>DB-YOLOv5x</td>
<td>640</td>
<td>176.5</td>
<td>440.6</td>
</tr>
</tbody>
</table>

4.5.3. The Influence of Each Module on the Model

To verify the performance of the improved detection model, we compared the proposed model with the original YOLOv5 model and calculated the mean average precision (mAP) index for evaluation. As discussed in Section 3, we refer to the model with the CBNet module as YOLOv5_cb, the model with the CBNet-tiny module is YOLOv5_cbty,
the YOLOv5_cb model with the BiFPN module is YOLOv5_bi, and the model proposed in this paper is DB-YOLOv5. The experimental results are shown in Table 3.

**Table 3. Ablation experiment results of the model on the VisDrone-DET dataset.**

<table>
<thead>
<tr>
<th>Model</th>
<th>CBNet-tiny</th>
<th>CBNet</th>
<th>Bi-FPN</th>
<th>SPANet</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48.44%</td>
</tr>
<tr>
<td>YOLOv5_cbty</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>49.26%</td>
</tr>
<tr>
<td>YOLOv5_cb</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>49.65%</td>
</tr>
<tr>
<td>YOLOv5_bi</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>50.74%</td>
</tr>
<tr>
<td>DB-YOLOv5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>51.53%</td>
</tr>
</tbody>
</table>

It can be seen from the results in Table 3 that the three improved methods proposed in this paper significantly increased the detection accuracy of the categories in the UAV dataset VisDrone-DET. Compared with the baseline model, after adding the faster CBNet-tiny module, the YOLOv5_cbty model achieved an 0.82% improvement on the mAP indicator. At the same time, the mAP index of the model with a complete CBNet module had a 1.21% improvement compared to the benchmark model and a 0.4% performance improvement compared to the YOLOv5_cbty model. On this basis, the model obtained by adding the BiFPN module also had a 2.3% improvement in performance indicators compared to the benchmark model and a 1.1% performance improvement based on the YOLOv5_cb model. Moreover, our proposed model DB-YOLOv5 had a performance improvement of nearly 3.1% compared to the benchmark model and a 0.8% improvement compared to YOLOv5_bi. The contribution of the three modules to our model from high to low were CBNet, BiFPN, and SPA.

5. Conclusions

In this paper, we proposed a DB-YOLOv5 UAV object detection algorithm to address the issue of detecting small targets in UAV images. We built the model on top of YOLOv5 by incorporating a composite backbone network, bidirectional feature pyramid, and pyramid attention mechanism, which improved the network’s capability for multi-scale feature fusion and small target detection. Our experiments on the VisDrone-DET dataset demonstrated that the proposed model achieved better performance in terms of objective detection metrics, making it suitable for small target detection tasks in UAV images. The proposed method has significant implications for security surveillance, particularly in the field of network security and privacy. By identifying small targets in UAV images, our approach can aid in detecting potential security threats, such as identifying security vulnerabilities in critical infrastructure and monitoring public events for potential security risks. Overall, this research provides a valuable contribution to the field of security surveillance by enhancing the capabilities of object detection algorithms for small target detection in UAV images, ultimately improving network security and privacy. Our next step is to develop a practical platform based on the simulation experiments. However, since we need to collect image data before learning, this process may be relatively slow. Our ultimate goal is to create a real-time platform and conduct experiments in a real environment. These are the directions for our future work.

**Author Contributions:** Conceptualization: L.T. and Y.L.; methodology: Y.L. and X.H.; formal analysis and investigation: Y.L.; writing—original draft preparation: Y.L.; writing—review and editing: Y.L., W.L., L.T., H.Z., X.H. and X.J.; supervision: L.T. and Y.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Natural Science Foundation of Chongqing (CSTB2022NSCQ-MSX1415).

**Data Availability Statement:** The dataset used during the current study is available at: [http://aiskyeye.com/download/object-detection-2/] (accessed on 27 June 2023).
Conflicts of Interest: All authors have read the final manuscript, have approved the submission to the journal, and have accepted full responsibilities pertaining to the manuscript’s delivery and contents. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References


46. Zhang, Q.; Ma, W.; Wang, Y. Backdoor Attacks on Image Classification Models in Deep Neural Networks. Chin. J. Electron. 2022, 31, 199–212. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.