*remote sensing*

MDPI

*Article*

# FiFoNet: Fine-Grained Target Focusing Network for Object Detection in UAV Images

**Yue Xi [1], Wenjing Jia [2], Qiguang Miao [3,*], Xiangzeng Liu [3], Xiaochen Fan [4] and Hanhui Li [5]**

[1] Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China
[2] Global Big Data Technologies Centre, University of Technology Sydney, Ultimo, NSW 2007, Australia
[3] School of Computer Science and Technology, Xidian University, Xi'an 710071, China
[4] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[5] School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen 518107, China
* Correspondence: qgmiao@xidian.edu.cn

**Abstract:** Detecting objects from images captured by Unmanned Aerial Vehicles (UAVs) is a highly demanding task. It is also considered a very challenging task due to the typically cluttered background and diverse dimensions of the foreground targets, especially small object areas that contain only very limited information. Multi-scale representation learning presents a remarkable approach to recognizing small objects. However, this strategy ignores the combination of the sub-parts in an object and also suffers from the background interference in the feature fusion process. To this end, we propose a **Fi**ne-grained Target **Fo**cusing **Net**work (FiFoNet) which can effectively select a combination of multi-scale features for an object and block background interference, which further revitalizes the differentiability of the multi-scale feature representation. Furthermore, we propose a Global–Local Context Collector (GLCC) to extract global and local contextual information and enhance low-quality representations of small objects. We evaluate the performance of the proposed FiFoNet on the challenging task of object detection in UAV images. A comparison of the experiment results on three datasets, namely VisDrone2019, UAVDT, and our VisDrone_Foggy, demonstrates the effectiveness of FiFoNet, which outperforms the ten baseline and state-of-the-art models with remarkable performance improvements. When deployed on an edge device NVIDIA JETSON XAVIER NX, our FiFoNet only takes about 80 milliseconds to process an drone-captured image.

**Keywords:** object detection; Unmanned Aerial Vehicles; deep learning



**Citation:** Xi, Y.; Jia, W.; Miao, Q.; Liu, X.; Fan, X.; Li, H. FiFoNet: Fine-Grained Target Focusing Network for Object Detection in UAV Images. *Remote Sens.* **2022**, *14*, 3919. https://doi.org/10.3390/rs14163919

Academic Editor: Eufemia Tarantino

Received: 21 June 2022
Accepted: 8 August 2022
Published: 12 August 2022



## 1. Introduction

Unmanned Aerial Vehicles (UAVs) equipped with cameras have received a lot of attention in recent years [1–3]. UAVs can be deployed rapidly with a wide range of new applications, including aerial photography and video surveillance, at a relatively low cost. Therefore, automatic understanding of visual data captured by UAVs is highly demanding, bringing computer vision and UAVs together more and more closely. In the field of computer vision, significant progress has been achieved in object detection. Existing detectors such as the YOLO family [4,5] and Faster RCNN [6] can achieve satisfying performance on natural images. The targets to be recognized in natural scenes generally consist of a large number of pixels. These detectors usually demand a huge amount of computing resources to ensure their capability and performance. However, the existing detectors perform poorly in situ because UAV images contain quite small objects with very limited numbers of pixels, and the UAVs' airborne computational resources are very limited [7–9].

The difficulty of UAV object detection lies in building robust features to distinguish foreground targets with limited pixels from background clutter [10–12]. Existing methods can be roughly grouped into three major streams, i.e., super-resolution-

based (SR-based) methods, context-based methods and multi-scale representation-based (MR-based) methods.

**(1) The SR-based methods** attempt to super-resolve the whole image or the Regions of Interest (RoIs) [9,13,14], and then, they perform object detection with a general-purpose object detector [6,15]. The newly generated details of RoIs can boost the detector's performance to a higher degree. Efficiently selecting regions to be reconstructed is critical to the efficiency of the algorithm. While foreground reconstruction can effectively improve the detection accuracy, background reconstruction just increases the calculation burden of the algorithm [13]. However, it is difficult to locate foreground regions containing objects of interest while excluding any background.

**(2) The context-based methods** leverage the relationship between the object and its surrounding environment to infer the original region of the small object [16–18]. However, due to the complexity and diversity of UAV background scenes, it is often difficult to build such contextual relationships. SR-based and context-based methods generally explicitly design a module specifically used for super-resolving RoIs or encoding context information, respectively, which can significantly increase the computation cost. Therefore, these algorithms are difficult to be deployed on UAVs.

**(3) The MR-based methods** first use different level features to represent objects [19–21] and then recognize these objects in separate feature levels. Specifically, a high-level feature with a low resolution treats an object as a whole, and a low-level feature with a high resolution focuses on the object's parts, such as its boundaries, as shown in Figure 1a. These methods build the *coarse-grained* features, which treat each object as a whole region and process them separately. However, this strategy neglects the *fine-grained* features in an object (as shown in Figure 1b), which has been demonstrated to improve object detection performance [22,23]. Furthermore, there is severe background interference in the combination of low-level and high-level features. In Figure 2a, for example, the red area covers parts of the background in the original image but misses the small targets in the far distance. In Figure 2b, the feature map focuses more accurately on the small objects. The background interference, in turn, adversely affects the learning in the subsequent layers, resulting in misclassifications in the final predictions, especially for recognizing small objects.

In this paper, we propose a **Fi**ne-grained Target **Fo**cusing **Net**work (FiFoNet) to improve the performance of object detection in UAV images through aggregating fine-grained objects' sub-parts with a special focus on foreground target areas. Compared with existing detectors, FiFoNet is distinctive in two significant aspects: (1) FiFoNet aggregates sub-part features in an object from different levels of features to provide a finer-grained object representation. (2) The fine-grained object representation can focus more on the foreground targets by blocking background interference with an object mask under the guidance of the object position label. We design a **G**lobal-**L**ocal **C**ontext **C**ollector (GLCC) to further improve the accuracy of small object detection. Our GLCC module utilizes several convolution filters with different kernel sizes to collect both global and local context information surrounding the objects.

In summary, our main contributions are as follows:

First, we propose a novel conceptual feature representation, called Fine-grained Target Focusing (FiFo) Representation, which aggregates sub-regions from multi-scale features and blocks background interference.

Second, we propose FiFoNet, which effectively detects objects in UAVs' images with our proposed FiFo representation.

Third, we propose GLCC, which utilizes a Global Average Pooling operation to encode global context information and dilated convolutional layers to extract local context information to enhance the low-quality representations of small objects.

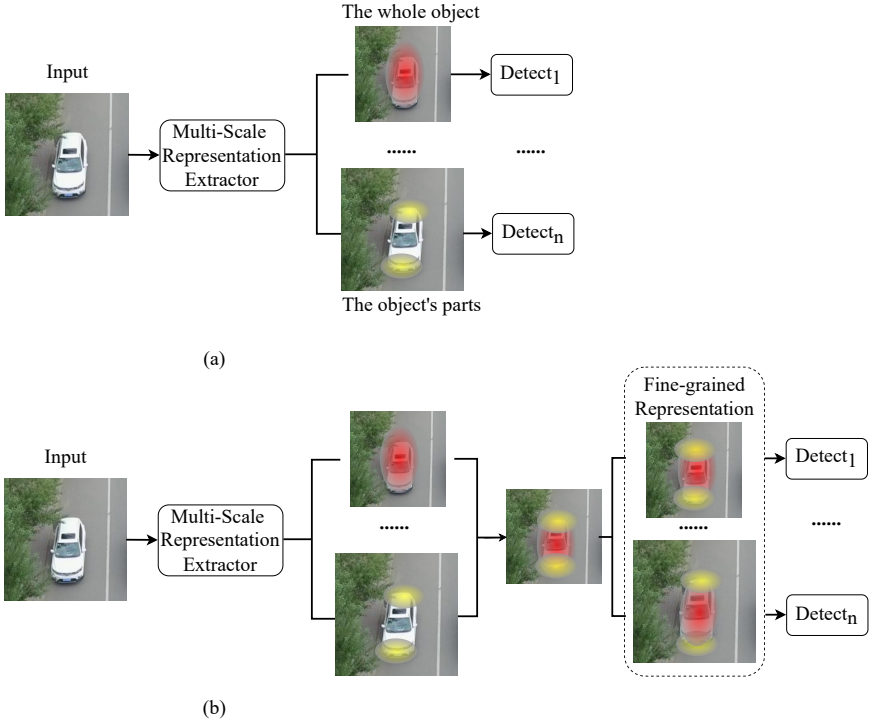


**Figure 1.** Comparisons between the conventional coarse-grained representation (**a**) and our fine-grained representation (**b**).

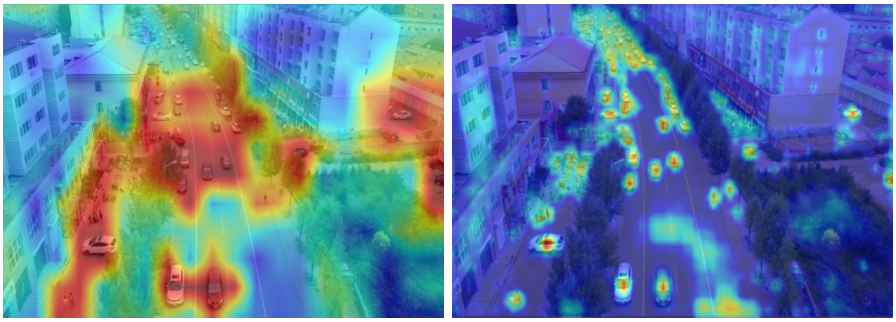


**Figure 2.** The heatmap visualization of the high-level (**a**) and low-level (**b**) features. The features visualized in (**a**) appear to be noisy, covering more background but missing the small targets (the cars) in the far distance region. The features visualized in (**b**) focus more accurately on the objects, especially on small targets.

Extensive experiments, including overall and ablation studies, are conducted on two widely used datasets VisDrone2019 [8] and UAVDT [24], as well as our VisDrone_Foggy dataset. Ablation studies show that with the proposed FiFA, TFB and GLCC modules, our FiFoNet unleashes the ability of multi-scale feature representations and has improved the detection accuracy by 1.1%, 1.0%, and 0.9%, respectively, on the VisDrone2019 dataset. Moreover, we compare our FiFoNet with ten state-of-the-art (SOTA) and baseline detectors to verify its effectiveness. On the VisDrone2019 dataset, the $AP_{50}$ result of FiFoNet is 63.80%, which outperforms the SOTA detector SAIC-FPN [25] by 0.83%. On the UAVDT dataset, our FiFoNet's $AP_{50}$ result is 36.80%, outperforming the SOTA detector GLSAN [9] by 6.30%. Last but not least, we deployed the proposed FiFoNet on an embedded computing board, NVIDIA Jetson Xavier NX, and tested it on the VisDrone2019 dataset. We have achieved an $AP_{50}$ of 57.5% and an average processing speed of 79.5 milliseconds per image on the edge device.

## 2. Related Work

Images captured by UAVs represent a special object detection scene. UAV images usually contain a large number of small targets. Small objects, whose feature representations usually are low quality, are the main reason for the poor detection performance. Some high-performing algorithms for small object detection have expanded the application field of UAVs. We will review three research directions for UAV object detection. The related work of the three research directions is summarized in Table 1.

**Table 1.** Summary of the advantages and drawbacks of object detectors for drone-captured images.

| Methods | Advantages | Drawbacks |
| --- | --- | --- |
| SR-based methods [26–28] | Reconstructed the information of RoIs; can effectively recognize small objects. | Difficult to locate RoIs accurately at places with cluttered backgrounds where it is hard to reconstruct the ROIs. |
| Context-based methods [17,29,30] | Can effectively detect small targets with a fixed background (e.g., cars in roads). | Difficult to build such contextual relationships due to the diversity of UAV background scenes. |
| MR-based methods [15,31,32] | Can effectively detect multi-scale objects, including small, middle and large size objects. | Suffer from the feature-level imbalance issue. |

### 2.1. SR-Based UAV Object Detection

SR-based UAV object detection methods [9,13,26–28,33,34] have attempted to adopt super-resolution techniques to reconstruct the low-quality RoIs or their corresponding features into high-quality counterparts. Hu et al. [34] utilized a simple bilinear interpolation for better target localization. However, super-resolving the whole image is inefficient because the processed background can be irrelevant to the detection task, and this can increase the inference time substantially. Instead of super-resolving whole images, Bai et al. [33] firstly obtained RoIs by using a high-recall detector and then super-resolved those RoIs only. On the other hand, since image features contain rich contextual information, Noh et al. [13] and Li et al. [28] reconstructed the features, instead of the image patches, of RoIs to further improve detection performance. However, it is difficult to accurately estimate the positions of RoIs or the corresponding RoIs' features prior to object detection, which is a chicken-and-egg problem. The above-mentioned methods all ignored the uneven distribution of objects with various dimensions in UAV images. Critical crowded regions should be examined by a detector in fine detail even if it requires a heavy computational burden, whereas sparse regions should be given less attention or even ignored. Following this idea, Deng et al. [9] and Yang et al. [27] firstly zoomed in the crowded regions that contained a large number of objects and then super-resolved the proposed regions for final detection. Aiming at real-world applications, Mukhiddinov [35] developed a smart glass system for blind and visually impaired (BVI) people. The system can recognize objects in low-quality images and help BVI people navigate in dark–light or foggy environments. However, these methods generally consist of three stages, namely crowded region proposal, low-quality image enhancement and object detection, which are inefficient and cannot be trained in an end-to-end fashion.

### 2.2. Context-Based UAV Object Detection

Context-based UAV object detection methods [16,17,29,30,36–38] have attempted to embed the relationship between an object and its surrounding environments into its original RoI's features. Bell et al. [29] adopted a spatial recurrent neural network to capture the contextual information outside the RoIs. Tang et al. [17] proposed the Pyramidbox to learn features from contextual parts around small targets and leveraged the joint fusion of high-level and low-level features. A hierarchical contextual information extracting mod-

ule [30] was proposed to integrate segmentation features into object detection features. Peng et al. [16] provided context information in high-level layers to supply low-level features to improve their semantic discriminativity. However, due to the complexity and diversity of the UAV images' backgrounds, it is difficult to build such contextual relationships. Integrating contextual information can also lead to increase background noise, which may result in degraded performance.

### *2.3. MR-Based UAV Object Detection*

Most of the early object detectors [15,31,32,39–41] failed to detect small objects. We argue the main reason is that features used for recognizing objects are typically extracted in the last layer, and when image feature maps are down-sampled with pooling operations in the feature extraction process, repeated down-sampling operations can degrade the quality of the small objects' features. Specifically, the hierarchical structure of neural networks allows them to extract feature maps with different spatial resolutions. Features extracted by convolutional filers with large kernel sizes in high-level layers contain much semantic information but lose detailed information due to their low resolution. Whereas features extracted in lower-level layers with convolutional filters of smaller kernels are in a higher resolution but lack semantic information. Therefore, MR techniques are introduced to aggregation to improve small object detection accuracy.

MR-based UAV object detection methods [42–46] use a strategy of combining the rich semantic information in high-level features for target classification and the detailed spatial information in low-level features for determining targets' positions. PANet [43] proposed adding a bottom–up path to supply the object spatial information in low-level features to the high-level features, which shortens the information path between the lower-layer and topmost-layer features. A Balanced Feature Pyramid [42] was proposed, which consisted of four stages, i.e., rescaling, integrating, refining and strengthening. The same deeply integrated balanced semantic features were used to enhance the multi-scale features. NAS [44] provided a new exploration direction for vision tasks. NAS-FPN [46] and Bi-FPN [45] utilized neural architecture searches to search Feature Pyramid Networks (FPN) and Path Aggregation Feature Pyramid Networks, respectively, for a better cross-scale feature network topology. However, the search processes required a huge amount of GPU resources and computation time.

In our work, we build a fine-grained object representation from different layers and introduce object position information and context information into the process of feature fusion to obtain multi-scale features with more expressive ability.

## 3. Methodology

### *3.1. Overview*

The overview of the pipeline of FiFoNet is illustrated in Figure 3. It consists of four modules: (1) A CNN-based backbone for feature extraction; (2) A top–down pathway extracting multi-scale feature representations via capturing the detailed position information and semantic information; (3) A fine-grained target-focusing module for further refining the multi-scale feature representations extracted by the second module; (4) A head for estimating the position and classification score of the resultant bounding box.

Our key idea is to build a *fine-grained target focusing representation* to further unleash the ability of the multi-scale feature representations. Toward this end, a fine-grained feature aggregation (FiFA) block is proposed to select a combination of sub-regions from multi-scale features. Moreover, a Target-Focusing Block (TFB) is proposed to focus the attention on the RoIs so as to suppress background noise.

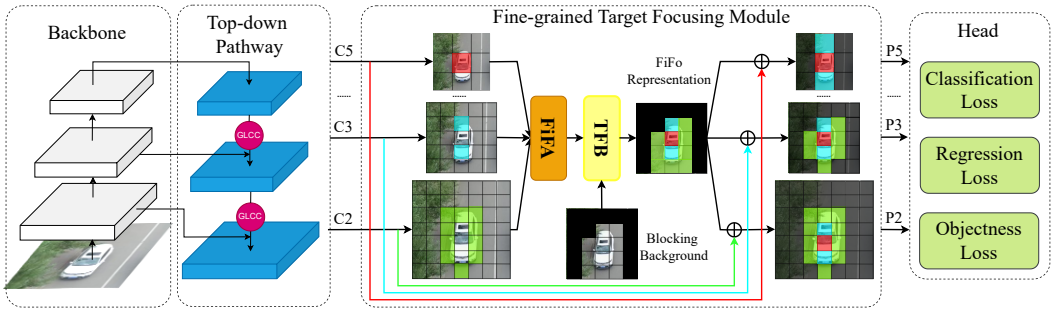


**Figure 3.** The pipeline of the proposed FiFoNet, which consists of four modules. The first module is a backbone for feature extraction. The second module, which includes our proposed GLCC, is a top–down pathway for extracting multi-scale features. The third module is the newly proposed fine-grained target-focusing module for fine-grained feature aggregation and background noise suppression at the feature level. The last module is a head for predicting classification scores and bounding box positions. These four modules are merged into a unified network for an end-to-end training.

### *3.2. Fine-Grained Feature Aggregation (FiFA)*

The MR-based detectors lay more emphasis on object-level coarse-grained approaches and consider each object as a whole or a large part. However, the coarse-grained features tend to miss those fine details critical for detecting small objects and hence leave small objects in a disadvantaged situation. Different from the coarse-grained approaches, our FiFA is performed on the adaptive combination of multi-scale features to obtain fine-grained object representation for both large and small objects without significantly increasing the computational complexity.

Figure 4 illustrates the details of FiFA. We utilize the feature activation output from each stage's last residual block. The outputs of these last residual blocks are defined as $\boldsymbol{C} = \{C_2, C_3, C_4, C_5\}$ for outputs of Conv2, Conv3, Conv4, and Conv5. We do not use Conv1 in the pyramid due to its large memory footprint. The input of FiFA is $\boldsymbol{C}$, and its output is $R$. The detailed fusion process is described as follows.

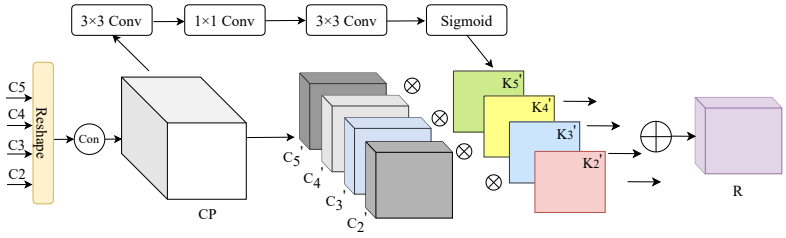


**Figure 4.** The proposed fine-grained feature aggregation (FiFA) Block.

Firstly, we resize the features $\{C_3, C_4, C_5\}$ to the size of $C_2$ to integrate multi-level features. A $1 \times 1$ convolution layer is used to reduce the number of channel dimensions of $\{C_3, C_4, C_5\}$, and we then perform bilinear interpolation on the three feature maps to up-sample them. After the above operations, we obtain the reshaped features with the same size as $C_2$. Secondly, we concatenate the reshaped features to obtain the integrated feature $CP$. A $3 \times 3$ convolution layer is used to extract local context information. Then, a $1 \times 1$ convolution layer is used to transform the channel dimension to the number of elements in $\{C_2, C_3, C_4, C_5\}$. A $3 \times 3$ convolution layer is adopted to further extract local information, and we use a sigmoid activation function to generate a weight map $\boldsymbol{K} = \{K_2, K_3, K_4, K_5\}$, where $K_i$ is the $i$-th weight map for the shape $C_i$, the value of elements in $K_i$ ranges in $[0, 1]$. The $i$-th weight map $K_i^{'}$ is obtained by a broadcasting operation in which the original $i$-th

weight map $K_i$ is stretched to become an array of same shape as $C_i^{'}$. Finally, the weight map $\boldsymbol{K'}$=$\{K_2^{'}, K_3^{'}, K_4^{'}, K_5^{'}\}$ is multiplied with the features $\{C_2^{'}, C_3^{'}, C_4^{'}, C_5^{'}\}$ to obtain $R$ as follows:

$$R = \sum_{i=2}^{5} K_i^{'} \times C_i^{'} \tag{1}$$

The pseudo-code for aggregating the *fine-grained features* with our FiFA module is illustrated in Algorithm 1.

**Algorithm 1** The pseudo-code for aggregating *fine-grained features* with our FiFA module.

**Input:** Features extracted from different convolution layers $\{C_2, C_3, C_4, C_5\}, C_i \in \mathbb{R}^{w_i \times h_i \times c_i}$.
**Output:** Features aggregated by our FiFA module $R \in \mathbb{R}^{w_2 \times h_2 \times c_2}$.
1: **Reshape** $\{C_3, C_4, C_5\}, C_i \in \mathbb{R}^{w_i \times h_i \times c_i} \rightarrow \{C_3^{'}, C_4^{'}, C_5^{'}\}, C_i^{'} \in \mathbb{R}^{w_2 \times h_2 \times c_2}$
2: **Concatenate** $\{C_2, C_3^{'}, C_4^{'}, C_5^{'}\} \rightarrow CP \in \mathbb{R}^{w_2 \times h_2 \times 4c_2}$
3: **Conv**($CP$) $\rightarrow \boldsymbol{K}$=$\{K_2, K_3, K_4, K_5\}, K_i \in \mathbb{R}^{w_2 \times h_2 \times 1}$
4: **Broadcast**($\boldsymbol{K}$) $\rightarrow \boldsymbol{K'} = \{K_2^{'}, K_3^{'}, K_4^{'}, K_5^{'}\}, K_i^{'} \in \mathbb{R}^{w_2 \times h_2 \times c_2}$
5: **Aggregate** $R$ with Equation (1).

### *3.3. Target Focusing Block (TFB)*

As shown in Figure 2, the fine-grained object representation generated by our FiFA block tends to contain some background noise, instead of focusing on the targets. We propose a Target-Focusing Block (TFB) to focus the fine features more on the target area and less on the background area. Then, representation learning is performed under the constraint of the object position.

Figure 5 presents the details of our TFB. The input of TFB is the feature $R$, whose output is denoted as $\{P_2, P_3, P_4, P_5\}$. We design a mask-guided mechanism to refine the fused feature $R$ to suppress background noise and highlight the foreground objects. Specifically, $R$ first passes through a $3 \times 3$ convolutional layer to encode the local context information, which then goes through a $1 \times 1$ convolution layer to learn a one-channel feature map. The value of the one-channel feature map indicates the likelihood of a pixel in the foreground or background. Finally, a new enhanced feature map is obtained by multiplying the fused feature $R$ and the one-channel feature map passing through a sigmoid function.

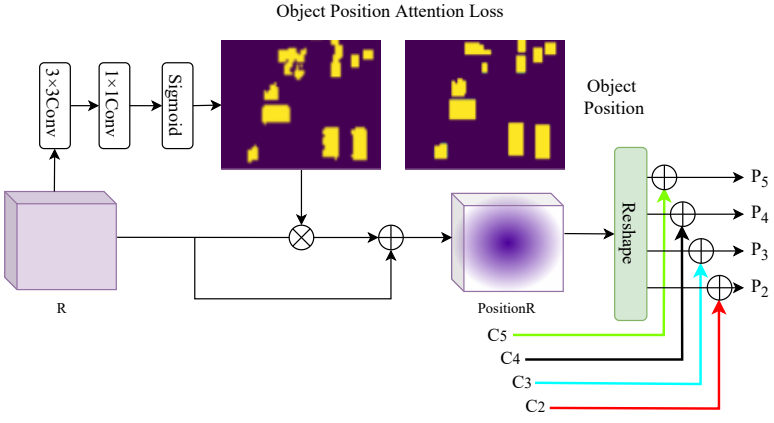


**Figure 5.** The Target-Focusing Block (TFB).

For supervised training, the cross-entropy loss between the one-channel feature map and the binary mask is utilized to compute the object position attention loss. Due to the lack of mask annotation for high-precision object positions in UAV images, we assign the value of 1 to all the pixels inside the ground-truth bounding box and 0 for all other pixels to obtain the object position mask, as shown in Figure 5. Regardless of annotations for objects' categories, all the value of the pixels inside the ground-truth bounding boxes in an image have the value of 1 to highlight the foreground regions.

Thus, the refined feature map $PositionR$ is defined as:

$$M = \sigma(\phi(R)) \tag{2}$$

and

$$PositionR = R + M \times R, \tag{3}$$

where $\phi$ represents convolutional layers, $\sigma$ denotes a sigmoid function, and $M$ is the object mask map.

We use $PositionR$ to improve the original multi-scale representations. $PositionR$'s shape is the same as $C_2$, which is different from $C_3, C_4$, and $C_5$. We first reshape it to the size of $C_i$ and then add $PositionR$ to the corresponding $C_i$ to obtain the final enhanced multi-scale representations $\boldsymbol{P}= \{P_2, P_3, P_4, P_5\}$ for final detection, which is expressed as:

$$P_i = \psi(PositionR) + C_i. \tag{4}$$

Here, $\psi$ denotes the operation of convolutional layers and up-sampling layers to reshape $PositionR$ to the size of $C_i$.

The pseudo-code of refining features with our TFB module is summarized in Algorithm 2.

**Algorithm 2** The pseudo-code for refining features with the proposed Target-Focusing Block (TFB).

**Input:** Features aggregated by FiFA $R \in \mathbb{R}^{w_2 \times h_2 \times c_2}$;
the ground truth of objects' position $GT_{ObjectPosi}$.
**Output:** The final enhanced multi-scale features $\boldsymbol{P} = \{P_2, P_3, P_4, P_5\}, P_i \in \mathbb{R}^{w_i \times h_i \times c_i}$,
the object position loss $loss_{AT}$.
**Estimate** an object mask map, $\sigma(\phi(R)) \rightarrow M \in \mathbb{R}^{w_2 \times h_2 \times 1}$
**Compute** object position attention loss, $loss_{AT}(M, GT_{ObjectPosi})$
**Obtain** the refined feature map, $R + M^{'} * R \rightarrow PositionR$
**Resize** $PositionR_i \rightarrow PositionR_i^{'} \in \mathbb{R}^{w_i \times h_i \times c_i}$
**Output** the final $\boldsymbol{P}$:
    $P_2 = PositionR_2^{'} + C_2$
    $P_3 = PositionR_3^{'} + C_3$
    $P_4 = PositionR_4^{'} + C_4$
    $P_5 = PositionR_5^{'} + C_5$

### *3.4. Global-Local Context Collector*

As shown in Figure 3, the lateral connection $F_i(C_i)$ in the original top–down pathway is a $1 \times 1$ convolutional kernel, which is utilized to reduce or increase the number of the feature channel. The features extracted with the $1 \times 1$ convolutional kernel generally lack contextual information due to the small and fixed size of the convolutional kernel. We propose to collect both global and local contextual information in our proposed detector so as to improve small object detection performance.

Our proposed Global–Local Context Collector, denoted as GLCC , consists of three components, i.e., a global average pooling layer followed by a $1 \times 1$ convolutional layer, several dilated convolutions with different $3 \times 3$ kernels and atrous rates and a $1 \times 1$ convolutional layer. The global average pooling layer is used to collect the global context information, and several dilated convolutions are used to collect the local context information. Figure 6 illustrates the details of GLCC. In the top branch, a $1 \times 1$ convolutional layer $\psi_i$ is used to embed the input feature $C_i$. In the bottom branch, a global average pooling layer $\phi_i$ is adopted to collect the global contextual information at the image level. In the middle branch, several $3 \times 3$ convolution filters $v_i^d$ with the atrous rate $d = (1, 2, ..., N)$ are utilized to encode local context information. Finally, these features extracted through the above three branches are concatenated together. In particular, we formulate this procedure as follows:

$$F_i(C_i) = \sum_{k=1,2,...,N} Con(v_i^d(C_i), \psi_i(C_i), \phi_i(C_i)) \tag{5}$$

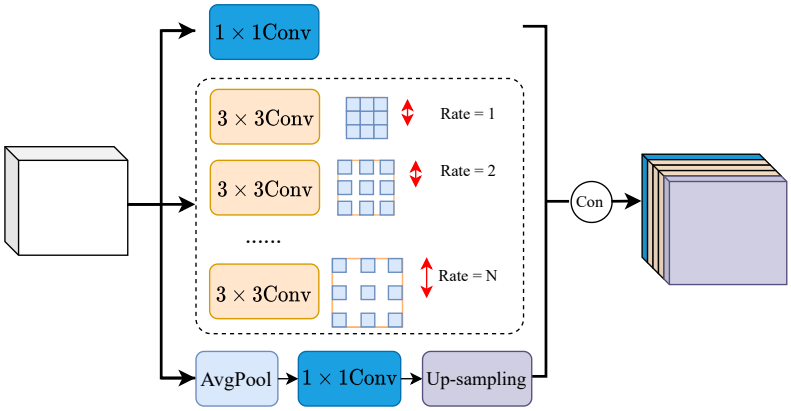


**Figure 6.** The details of the proposed Global–Local Context Collector.

The proposed GLCC mainly consists of several convolution filters with different kernel sizes. Our GLCC captures contextual information surrounding targets to improve the expressive power of target representations. Especially, this strategy is very effective for enhancing small objects' representations.

## 4. Experiments

In order to demonstrate that our FiFoNet method can effectively recognize small objects in UAV images, we conducted experiments on the UAV benchmark datasets VisDrone2019 [7], UAVDT [24], and our synthetic VisDrone_Foggy. In the following subsections, we first describe the three datasets and then use them to verify the effectiveness of our FiFoNet for UAV object detection.

### *4.1. Datasets and Models*

In this work, VisDrone2019 [7] and UAVDT [24] are used to verify the effectiveness of our FiFoNet because these two datasets contain a large number of small size objects. We will give the details of the datasets and models as follows.

*(1) VisDrone2019* [7]: The VisDrone2019 benchmark includes 6471 images for training, 548 images for validation, and 3190 testing images captured by UAVs. The captured images' resolution is about 2000 × 1500 pixels. These images are labeled with bounding boxes and ten categories: bicycle, awning-tricycle, tricycle, van, bus, truck, motor, pedestrian, person and car.

*(2) VisDrone_Foggy*: Our synthetic VisDrone_Foggy dataset is built upon the VisDrone2019 benchmark dataset. According to the atmospheric scattering model [47], we transform images in the VisDrone2019 dataset to foggy images for our VisDrone_Foggy. We generate images with thin, medium thick, and thick fog by setting different parameters of the atmospheric scattering model, as shown in Figure 7. Our VisDrone_Foggy adopts the same annotations in the original VisDrone2019.

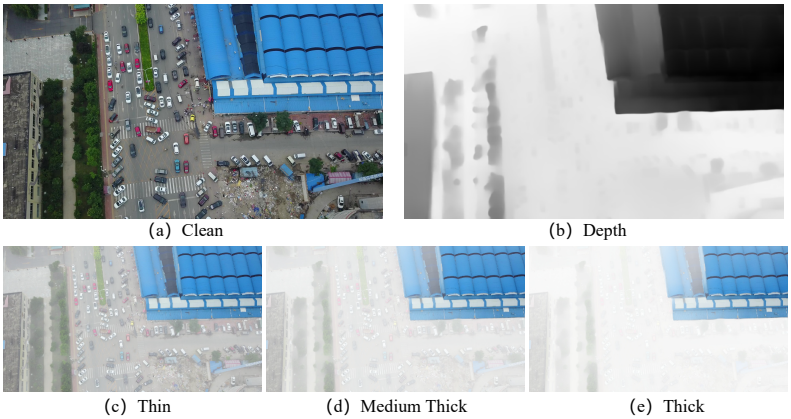


**Figure 7.** Examples of images from our Visdrone_Foggy dataset. (**a**) The original image; (**b**) The estimated depth map; (**c**) The image with thin fog; (**d**) The image with medium thick fog; (**e**) The image with thick fog.

Specifically, the formation of foggy images can be formulated as follows:

$$I(x) = J(x)t(x) + A(1 - t(x)) \tag{6}$$

where $I(x)$ is the observed foggy image, $J(x)$ is the corresponding clean images, $A$ is the global atmospheric light, and $t$ is the transmission describing the portion of the light.

The transmission $t$ can be described as follows:

$$t(x) = e^{-\beta d(x)} \tag{7}$$

where $\beta$ is the scattering coefficient of the atmosphere, and $d(x)$ is the depth map. We use ViTDepthNet [48] for image depth estimation.

*(3) UAVDT* [24]: The UAVDT benchmark includes 23,258 images for training and 15,069 images for testing. The captured images' resolution is about 1080 × 540 pixels. These images are labeled with bounding boxes and three predefined categories: bus, truck and car.

*(4) Models*: We have implemented several object detection models as the baselines, including SSD [49], FPN [50], YOLO [5], and FRCNN [6]. We also compare our method with the state-of-the-art (SOTA) methods, such as mSODANet [51], DSHNet [52], CRENET [53], GLSAN [9], ClustDet [27], SAIC-FPN [25], and HRDNet [54], which are designed specifically for UAV object detection.

### *4.2. Implementation and Evaluation Metrics*

*(1) Implementation*: We implement our FiFoNet with PyTorch 1.8.1. The proposed model is run on a server with an NVIDIA RTX3090 GPU and an edge device with a NVIDIA JETSON XAVIER NX. During the training stage, we use part of the pre-trained model YOLOv5 (https://github.com/ultralytics/yolov5, accessed on 1 August 2022), which saves a lot of training time. We use the Adam optimizer for training and use $3 \times 10^{-4}$ as the initial learning rate with the cosine learning rate schedule. The learning rate of the last epoch decays to 0.12 of the initial learning rate. The size of the input image of our model is very large, with the long side of the image being 1536 pixels, which is the same configuration as in TPH-YOLOv5 [55].

*(2) Evaluation Metrics*: The same evaluation metrics as in PASCAL VOC [56] are adopted to evaluate the detection performance of our FiFoNet. The metrics are defined as follows, including mean Average Precision (mAP) and Average Precision (*AP*):

$$AP = \int_0^1 P(R)dR. \tag{8}$$

Here, $P$ stands for Precision, measuring how accurate the prediction is, i.e., the fraction of correct positive instances among all the positive instances. $R$ represents Recall, measuring how good the classifier estimates the positives, i.e., the fraction of true positive instances among all the positive instances. $P(R)$ is the curve composed of $P$ and $R$. Then, $mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i$, where $N$ is the number of categories. $AP$ is averaged on ten Intersection over Union (IoU) values of $[0.50 : 0.05 : 0.95]$, $AP_{50}$ and $AP_{75}$ are computed at the single IoU of 0.5 and 0.75, respectively. $P$ and $R$ are defined as:

$$P = \frac{TP}{TP + FP} \tag{9}$$

and

$$R = \frac{TP}{TP + FN}, \tag{10}$$

where $FN$, $FP$ and $TP$ indicate the number of false negative predictions, false positive predictions and true positive predictions.

### *4.3. Ablation Studies*

In order to analyze the impact of our method and validate the contributions of each component of our approach, we conducted seven experiments on the VisDrone2019 dataset. YOLOv5 [57] is used as the baseline for the ablation studies.

*(1) The Effectiveness of FiFA*: We evaluate three methods to fuse the information of low-level features with high-level features. The three methods are Element-wise Sum and Average, Concatenation, and our FiFA. To compare their effectiveness, we apply different strategies on the same baseline. Table 2 presents that the detection accuracy of FiFA outperforms that of the Sum-and-Average approach by 1.1%. The result matches the intuition that the fine-grain feature fusion strategy of our FiFA module can release the advantages of the multi-scale feature representations more effectively. Therefore, we performed the remaining experiments with our FiFA module.

**Table 2.** Comparison of Average Precision obtained with different fusion strategies on VisDrone2019.

| Method | Sum and Average | Concatenation | FiFA |
|---|---|---|---|
| $AP_{50}$ | 33.5 | 33.8 | 34.6 |

*(2) The effectiveness of TFB*: The TFB is used to make the network suppress background noise and focus on foreground objects. Table 3 shows that the TFB obtain a 1.00% improvement compared with the baseline model.

We verify the effectiveness of the generated objects' mask for future refinement with the supervised information of object location. Specifically, the input feature map (Figure 8a) passes through a $3 \times 3$ convolutional layer and then a $1 \times 1$ convolution operation to estimate a one-channel mask (as shown in Figure 8d). The mask indicates the likelihood of the background and foreground. Finally, a new refined feature map is obtained, as shown in Figure 8c.

The comparison between the estimated attention mask (Figure 8d) and object position generated by ground truth without categories (Figure 8e) indicates that our TFB module can effectively estimate the target objects' positions and scales. We compare the original feature maps (Figure 8b) and feature maps generated by TFB (Figure 8c). The comparison suggests that the proposed TFB module can suppress the interference of background noise effectively.

**Table 3.** Comparison of Average Precision obtained with our method with/without each module for small object detection on VisDrone2019-Val.

| Method | Train Imgz | Test Imgz | mAP @.5 | Speed (ms) | Ped | People | Bicycle | Car | Van | Truck | Tricycle | Awning-Tricycle | Bus | Motor |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Baseline | 640 | 640 | 33.5 | 7.0 | 39.3 | 32.0 | 11.9 | 73.8 | 36.2 | 31.2 | 20.3 | 12.2 | 39.7 | 38.0 |
| Baseline + FiFA | 640 | 640 | 34.6 | 7.0 | 40.9 | 32.9 | 11.4 | 74.5 | 37.5 | 31.3 | 20.5 | 10.9 | 46.8 | 38.8 |
| Baseline + TFB | 640 | 640 | 34.5 | 7.0 | 39.8 | 33.1 | 11.0 | 74.5 | 37 | 31.7 | 18.6 | 11.2 | 48.7 | 39.0 |
| Baseline + tinyHead | 640 | 640 | 37.2 | 7.0 | 45.1 | 35.4 | 12.8 | 79.1 | 40.0 | 34.3 | 22.0 | 12.1 | 48.8 | 42.6 |
| Baseline | 640 | 1996 | 35.8 | 7.0 | 52.6 | 34.8 | 14.5 | 81.0 | 39.2 | 21.2 | 22.2 | 10.6 | 40.8 | 42.7 |
| Baseline | 1536 | 640 | 34.0 | 7.0 | 34.8 | 30.3 | 14.4 | 73.4 | 36.3 | 32.5 | 22.0 | 11.7 | 48.0 | 36.8 |
| Baseline | 1536 | 1996 | 55.6 | 15.6 | 69.2 | 54.9 | 36.2 | 89.1 | 55.6 | 49.4 | 44.7 | 24.8 | 69.6 | 62.6 |
| Baseline + tinyHead | 1536 | 1996 | 56.1 | 15.6 | 70.4 | 55.4 | 37.6 | 89.7 | 57.7 | 48.9 | 42.8 | 24.2 | 68.7 | 65.3 |
| Baseline + largeModel | 1536 | 1996 | 61.4 | 15.6 | 74.6 | 60.8 | 46.4 | 90.8 | 60.5 | 55.0 | 50.2 | 30.2 | 76.6 | 69.0 |

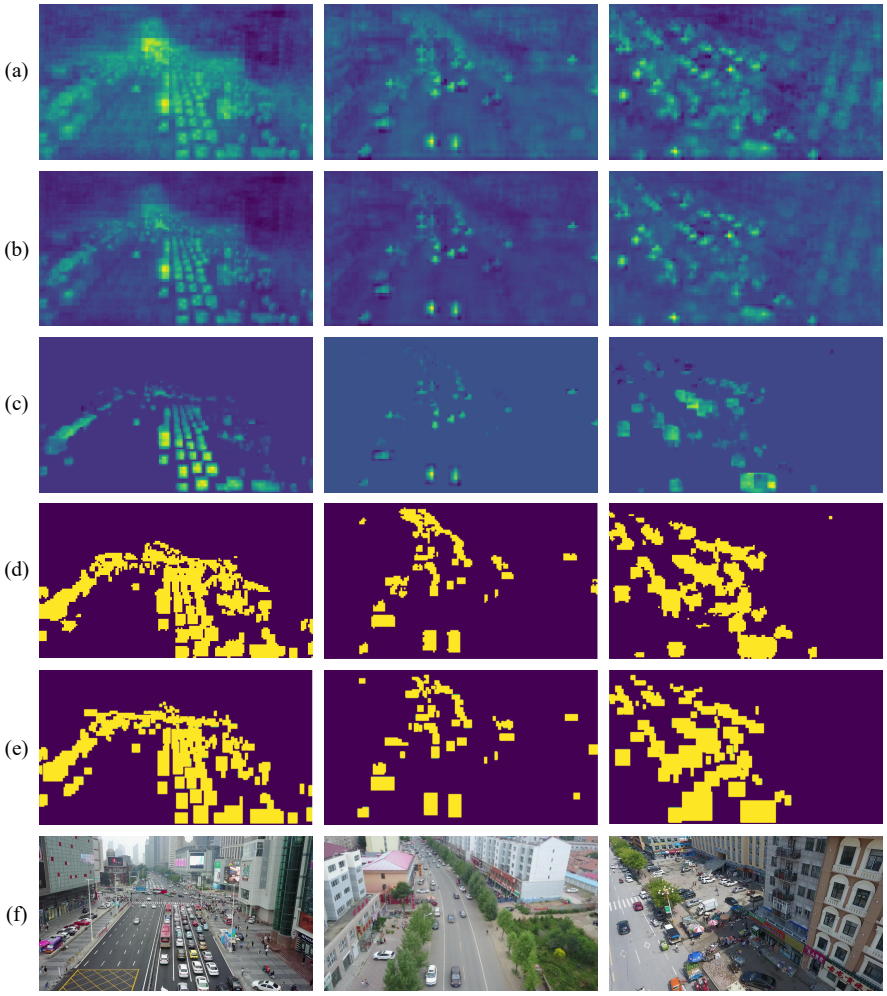


**Figure 8.** Visualization of the results obtained with our TFB module. (**a**) The feature map input to TFB. (**b**) The output feature map without TFB. (**c**) The output features to TFB. (**d**) The object mask generated by BFR. (**e**) The ground-truth object position without their category. (**f**) The original images.

*(3) The Effectiveness of GLCC*: Table 4 presents that GLCC can significantly improve the detection performance by collecting contextual information. As we gradually aggregate more features from different convolutional layers, the detection performance of the algorithm continues to improve. Firstly, we follow the architecture of FPN and demonstrate its performance in the setting of $k = 1, d = 1$, where $k$ is the kernel size and $d$ denotes the dilation rate. We add a $3 \times 3$ convolution in each lateral connection, which leads to 0.2, 0.4, and 0.3 gain in $AP$, $AP_{50}$, and $AP_{75}$.

Furthermore, we add more convolutions with different dilation rates to collect more contextual information. It can be observed that the addition of convolutions with a dilation rate of 2, 3, 4 and 5 improves the detection accuracy. The method in the last row benefits from all kinds of convolutions and achieves the best results, which leads to a 0.6, 0.9, and 0.8 gain in $AP$, $AP_{50}$, and $AP_{75}$. To this end, the convolutional layers in our GLCC module finally adopt the following configuration: kernel size = $[1, 3, 3, 3, 3, 3]$, dilation rate = $[1, 1, 2, 3, 4, 5]$. We denote GLCC with this configuration as $\text{GLCC}_{k6d6}$.

We further visualize some features generated by $\text{GLCC}_{k6d6}$ as shown in Figure 9. The first and second columns are the input images and their feature maps are generated by the feature extractor of the baseline. The last column is the feature maps enhanced by our $\text{GLCC}_{k6d6}$. It can be observed that the enhanced feature maps are better than the originally extracted feature maps, especially in the red-circled area. The visualization result demonstrates that our GLCC can effectively collect contextual information by convolutional

filters with different dilation rates and thus generate high-quality representations for final object detection.

**Table 4.** Ablation study of GLCC.

| k = 1 d = 1 | k = 3 d = 1 | k = 3 d = 2 | k = 3 d = 3 | k = 3 d = 4 | k = 3 d = 5 | $AP$[%] | $AP_{50}$[%] | $AP_{75}$[%] |
|---|---|---|---|---|---|---|---|---|
| ✓ | | | | | | 21.4 | 33.5 | 20.2 |
| ✓ | ✓ | | | | | 21.6 | 33.9 | 20.5 |
| ✓ | ✓ | ✓ | | | | 21.8 | 34.2 | 20.7 |
| ✓ | ✓ | ✓ | ✓ | | | 21.9 | 34.3 | 20.9 |
| ✓ | ✓ | ✓ | ✓ | ✓ | | 22.0 | 34.3 | 20.9 |
| ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 22.0 | 34.4 | 21.0 |

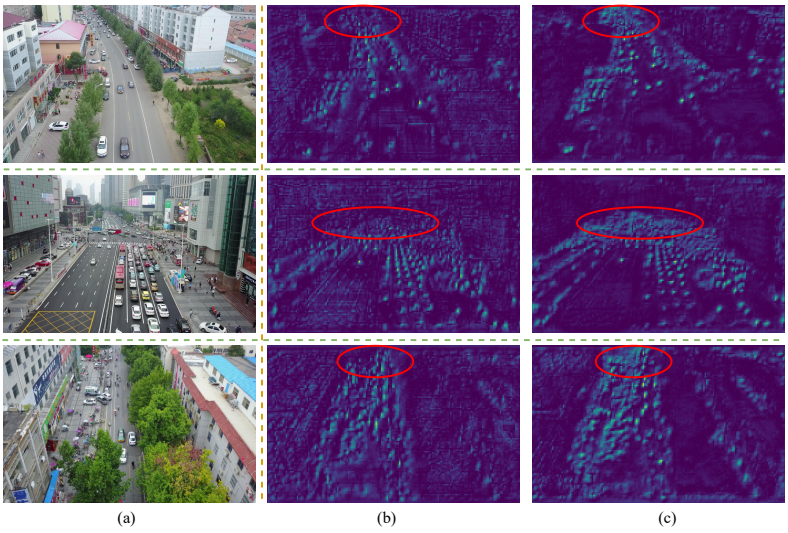


**Figure 9.** Visualization of the feature maps obtained with GLCC. (**a**) The input image. (**b**) Features obtained without GLCC. (**c**) Features obtained with GLCC.

*(4) Complexity Comparison*: Table 5 compares the model complexity of our FiFoNet with that of the baseline. The evaluation metrics of the experiment include FLOPs (Floating Point Operations, the calculation amount of a model), Parameters (the number of model parameters), and Times (in milliseconds) on a server and edge device. The evaluation metric Times is calculated by the average of processing all of the images from the validation set. The total time includes the image pre-processing time, inference time and post-processing time. Table 5 shows that the $AP_{50}$ of the model with our modules has increased by 1∼2%, while the number of model parameters and processing time only increase slightly.

*(5) Visual Comparison of the Detection Results*: We also visually compare the detection results obtained with and without the proposed components in Figure 10. Comparing Figure 10b with Figure 10d, it can be seen that smaller objects can be well detected by our method. This is because our method converts the position-semantic inconsistency issue into one that makes the RoIs' (located by tiny objects) multi-scale feature simultaneously contain detailed spatial information and strong semantic information.

*(6) Effect of the low-level representation*: We also conducted experiments to verify the effectiveness of low-level feature maps for UAV object detection. Table 3 shows that the tinyHead modification has boosted the performance by 3.7%. The tinyHead represents the baseline model with a lower-level representation extracted from the Backbone module. Figure 11 shows the comparison between the heatmaps of the last and second last layers. The red areas in the right column are darker and smaller than those in the middle column. The results show that the feature maps in the right column focus more accurately on the object spatial locations, indicating that our tinyHead can help make the network focus

more precisely on objects. This is because the lower-level features are generated by the convolutional filter with a smaller kernel size, which is beneficial for extracting small objects' features.

**Table 5.** Comparison of model complexity between our model and the baseline.

| Model | Input Size | $AP_{0.5}$ | FLOPs | Params (M) | Time (ms) | |
|---|---|---|---|---|---|---|
| | | | | | Server | Edge Device |
| Baseline | Small | 55.6 | 15.9 | 7.0 | 18.4 | 69.8 |
| | Large | 61.4 | 204.2 | 86.2 | 79.7 | 480.9 |
| Baseline + FiFA | Small | 56.7 | 16.6 | 7.2 | 19.1 | 74.5 |
| | Large | 62.9 | 209.0 | 87.7 | 82.2 | 497.8 |
| Baseline + TFB | Small | 56.8 | 18.5 | 7.4 | 19.5 | 75.3 |
| | Large | 63.1 | 220.8 | 88.6 | 82.4 | 501.2 |
| Baseline + FiFA + TFB | Small | 57.5 | 18.9 | 7.5 | 19.7 | 79.5 |
| | Large | 63.8 | 225.9 | 89.8 | 82.8 | 509.8 |


**Figure 10.** Visualization of the detection performance of our method. (**a**) Baseline detection results. (**b**) Zoomed-in baseline detection results on the sub-area. (**c**) Our detection results. (**d**) Our detection results on the corresponding sub-area.

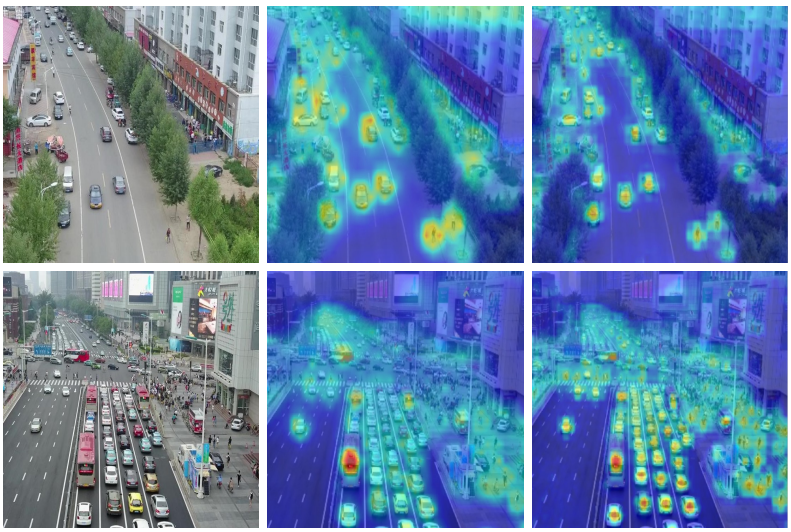

**Figure 11.** Heatmap comparison between our newly added head for lower-level features. The raw image (**left**), heatmap from the last second layer (**middle**), heatmap from the last layer (**right**).

*(7) Detection Results of Our FiFoNet on VisDrone_Foggy*: We evaluate our FiFoNet on the VisDrone_Foggy dataset. We synthesize thin, medium thick and thick fog with the parameters $\beta = 1.0, 2.0, 3.0$ in Equation (7). Table 6 shows the $AP_{50}$ results. FiFoNet improves the baseline by 3.90%, 2.59%, and 1.84% on the thin foggy, medium thick foggy, and thick foggy images, respectively. Figure 12 shows FiFoNet's detection results on the VisDrone_Foggy dataset. It can be observed that most of the objects are correctly recognized, in spite of a small number of false positive and false negative samples as the fog grows. The detection results demonstrates its effectiveness in object detection in foggy scene.

**Table 6.** Detection results of our FiFoNet with synthetic images on our VisDrone_Foggy dataset.

| Method | $AP_{50}$(%) | | |
|---|---|---|---|
| | **Thin Fog** | **Medium Thick Fog** | **Thick Fog** |
| Baseline | 54.25 | 53.72 | 51.01 |
| FiFoNet | 58.15 | 56.31 | 52.85 |

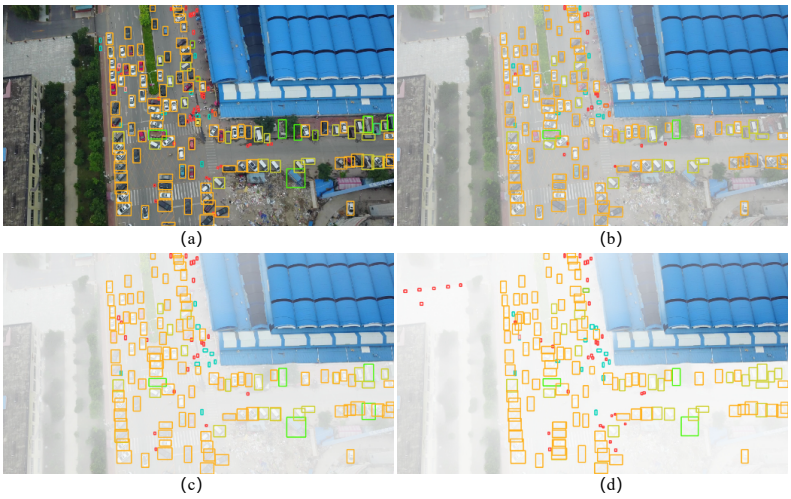


**Figure 12.** Visualization of the results obtained with our FiFoNet on the clean image (**a**), the thin (**b**), medium thick (**c**), and thick (**d**) fog scenes.

### *4.4. Comparison with State-of-the-Art Methods*

We compare our proposed FiFoNet with the SOTA algorithms on two datasets.

*(1) Detection Results on VisDrone2019 Dataset*: We compare the detection results with representative detectors' results on VisDrone2019 in Table 7, including two-stage detectors, i.e., FRCNN [6], FPN [50], and one-stage detectors, i.e., SSD [49] and YOLOv5 [57]. We achieve an $AP$ of 36.5%, $AP_{50}$ of 63.8% and $AP_{75}$ of 36.1%. The performance comparison with the SOTA methods, namely mSODANet [51], DSHNet [52], CRENet [53], GLSAN [9], ClustDet [27] SAIC-FPN [25], and HRDNet [54], is also summarized in Table 7. Compared to the SOTA drone-view detector (SAIC-FPN), the $AP$ is increased by 1.22% and $AP_{50}$ is increased by 0.83%, suggesting that our FiFoNet outperforms these SOTA methods. To make a fair comparison, we do not use overlays of various tricks, oversized backbones, or model ensembles, which are often used in existing methods dealing with UAV data. Figure 13 shows the object detection results on aerial images of large or low-light scenes. It is worthy mentioning that FiFoNet can detect people in night-time images.

*(2) Detection Results on UAVDT Dataset*: The performance comparison of our FiFoNet and SOTA detectors on the VisDrone dataset including FRCNN [6], ClusDet [27], GLSAN [9], and YOLOv5 [5] is presented in Table 7. It can be seen from the table that the proposed approach achieves an $AP$ of 21.3%, an $AP_{50}$ of 36.8% and an $AP_{75}$ of 22.5%, outperforming the SOTA methods.

**Figure 13.** Visualization of the results obtained with our FiFoNet on large or low-light scenes. Note the zoomed-in views of the crowded areas highlighted in green boxes.

**Table 7.** Comparison of our method with the baseline and SOTAs on VisDrone2019-Val and UAVDT. '-' means that the statistics are not available. The top two results are highlighted in red and green fonts.

| Method | VisDrone2019 | | | UAVDT | | |
|---|---|---|---|---|---|---|
| | $AP(\%)$ | $AP_{50}(\%)$ | $AP_{75}(\%)$ | $AP(\%)$ | $AP_{50}(\%)$ | $AP_{75}(\%)$ |
| SSD [49] (ECCV 16) | - | 15.20 | - | 9.30 | 21.40 | 6.70 |
| FRCNN [6] + FPN [50] | 21.80 | 41.80 | 20.10 | 11.00 | 23.40 | 8.40 |
| YOLOv5 [57] (Github 21) | 24.90 | 42.40 | 25.10 | 19.10 | 33.90 | 19.60 |
| DSHNet [52] (WACV 21) | 30.30 | 51.80 | 30.90 | 17.80 | 30.40 | 19.70 |
| CRENet [53] (ECCV 20) | 33.70 | 54.30 | 33.50 | - | - | - |
| GLSAN [9] (TIP 20) | 30.70 | 55.60 | 29.90 | 19.00 | 30.50 | 21.70 |
| ClustDet [27] (ICCV 19) | 32.40 | 56.20 | 31.60 | 13.70 | 26.50 | 12.50 |
| SAIC-FPN [25] (Nerocomputing 19) | 35.69 | 62.97 | 35.08 | - | - | - |
| HRDNet [54] (ICME 21) | 35.50 | 62.00 | 35.10 | - | - | - |
| mSODANet [51] (PR 22) | 36.89 | 55.92 | 37.41 | - | - | - |
| **FiFoNet (Ours)** | 36.91 | 63.80 | 36.11 | 21.30 | 36.80 | 22.50 |

## 5. Limitation and Discussion

In addition to the above success, our FiFoNet has certain limitations. FiFoNet trained on high-quality drone-captured images would fail to obtain satisfactory detection performance under adverse weather conditions, including foggy or raining scenarios. The main reason for the poor detection performance of FiFoNet is considerable inconsistency in data distribution between high-quality images under sunny weather and low-quality images under adverse weather. Therefore, one of the main limitations of FiFoNet is poor detection performance under adverse weather.

Our VisDrone_Foggy dataset is synthetic and not collected from the real world, while we have achieved preliminary detection results under foggy weather conditions. There is still a significant difference between synthetic and real-world data. The degradation process of the real-world dataset is very complicated. We need to comprehensively collect real-world fog drone-captured images to verify the effectiveness of FiFoNet in adverse weather.

## 6. Conclusions

In this paper, we have proposed our FiFoNet to effectively detect objects in UAV images. The proposed FiFoNet first builds a FiFo representation, which contains strong semantic information and detailed spatial positions. Then, the FiFoNet refines the multi-scale features to focus them on the foreground against the noisy background. Finally, the GLCC collects the global and local context information surrounding small objects to further improve object detection accuracy. Extensive experiments on benchmark datasets have shown the effectiveness of our proposed method in terms of both quantitative and visual results. Our core components, FiFA, TFB and GLCC, can be easily plugged into existing MR-based detectors. Our FiFoNet has been deployed on an embedded computing board running on a real drone.

**Author Contributions:** Conceptualization, Y.X. and W.J.; methodology, Y.X. and W.J.; validation, Y.X.; formal analysis, Y.X.; writing—original draft preparation, Y.X.; writing—review and editing, W.J., X.L., X.F. and H.L.; visualization, Y.X.; supervision, Q.M.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Fundamental Research Funds for the Central Universities (No. 20101216855) and the Key R&D Projects of Qingdao Science and Technology Plan (No. 21-1-2-18-xx).

**Data Availability Statement:** Not applicable.

**Acknowledgments:** We sincerely thank the authors of Yolov5 and Faster RCNN for providing their algorithm codes to facilitate the comparative experiments.

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

## References


1. Avola, D.; Cinque, L.; Diko, A.; Fagioli, A.; Foresti, G.L.; Mecca, A.; Pannone, D.; Piciarelli, C. MS-Faster R-CNN: Multi-stream backbone for improved Faster R-CNN object detection and aerial tracking from UAV images. *Remote Sens.* **2021**, *13*, 1670. [CrossRef]
2. Stojnić, V.; Risojević, V.; Muštra, M.; Jovanović, V.; Filipi, J.; Kezić, N.; Babić, Z. A method for detection of small moving objects in UAV videos. *Remote Sens.* **2021**, *13*, 653. [CrossRef]
3. Ma, Y.; Li, Q.; Chu, L.; Zhou, Y.; Xu, C. Real-time detection and spatial localization of insulators for UAV inspection based on binocular stereo vision. *Remote Sens.* **2021**, *13*, 230. [CrossRef]
4. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Paradise, NV, USA, 26 June–1 July 2016; pp. 779–788.
5. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. Scaled-yolov4: Scaling cross stage partial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 13029–13038.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, *1137–1149*. [CrossRef] [PubMed]
7. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. . [CrossRef] [PubMed]
8. Wen, L.; Du, D.; Zhu, P.; Hu, Q.; Wang, Q.; Bo, L.; Lyu, S. Detection, tracking, and counting meets drones in crowds: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 7812–7821.
9. Deng, S.; Li, S.; Xie, K.; Song, W.; Liao, X.; Hao, A.; Qin, H. A global-local self-adaptive network for drone-view object detection. *IEEE Trans. Image Process.* **2020**, *30*, 1556–1569. [CrossRef] [PubMed]
10. Yang, X.; Yan, J.; Liao, W.; Yang, X.; Tang, J.; He, T. Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**. [CrossRef]
11. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Sun, X.; Fu, K. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8232–8241.
12. Deng, C.; Wang, M.; Liu, L.; Liu, Y.; Jiang, Y. Extended feature pyramid network for small object detection. *IEEE Trans. Multimed.* **2021**, *24*, 1968–1979. [CrossRef]
13. Noh, J.; Bae, W.; Lee, W.; Seo, J.; Kim, G. Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In Proceedings of the the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9725–9734.
14. Bashir, S.M.A.; Wang, Y. Small object detection in remote sensing images with residual feature aggregation-based super-resolution and object detector network. *Remote Sens.* **2021**, *13*, 1854. [CrossRef]
15. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
16. Peng, J.; Wang, H.; Yue, S.; Zhang, Z. Context-aware co-supervision for accurate object detection. *Pattern Recognit.* **2022**, *121*, 108199. [CrossRef]
17. Tang, X.; Du, D.K.; He, Z.; Liu, J. Pyramidbox: A context-assisted single shot face detector. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 797–813.
18. Kong, Y.; Feng, M.; Li, X.; Lu, H.; Liu, X.; Yin, B. Spatial context-aware network for salient object detection. *Pattern Recognit.* **2021**, *114*, 107867. [CrossRef]
19. Jiao, L.; Gao, J.; Liu, X.; Liu, F.; Yang, S.; Hou, B. Multi-Scale Representation Learning for Image Classification: A Survey. *IEEE Trans. Artif. Intell.* **2021**. [CrossRef]
20. Qiao, S.; Chen, L.C.; Yuille, A. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 10213–10224.
21. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic head: Unifying object detection heads with attentions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 7373–7382.
22. Han, J.; Yao, X.; Cheng, G.; Feng, X.; Xu, D. P-CNN: Part-Based Convolutional Neural Networks for Fine-Grained Visual Categorization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 579–590. [CrossRef]
23. Song, L.; Li, Y.; Jiang, Z.; Li, Z.; Sun, H.; Sun, J.; Zheng, N. Fine-grained dynamic head for object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 11131–11141.
24. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 370–386.
25. Zhou, J.; Vong, C.M.; Liu, Q.; Wang, Z. Scale adaptive image cropping for UAV object detection. *Neurocomputing* **2019**, *366*, 305–313. [CrossRef]

26. Xi, Y.; Jia, W.; Zheng, J.; Fan, X.; Xie, Y.; Ren, J.; He, X. DRL-GAN: Dual-stream representation learning GAN for low-resolution image classification in UAV applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *14*, 1705–1716. [CrossRef]
27. Yang, F.; Fan, H.; Chu, P.; Blasch, E.; Ling, H. Clustered object detection in aerial images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 8311–8320.
28. Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; Yan, S. Perceptual generative adversarial networks for small object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1222–1230.
29. Bell, S.; Zitnick, C.L.; Bala, K.; Girshick, R. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Paradise, NV, USA, 26 June–1 July 2016; pp. 2874–2883.
30. Qiu, H.; Li, H.; Wu, Q.; Meng, F.; Xu, L.; Ngan, K.N.; Shi, H. Hierarchical context features embedding for object detection. *IEEE Trans. Multimed.* **2020**, *22*, 3039–3050. [CrossRef]
31. Li, Y.; Chen, Y.; Wang, N.; Zhang, Z. Scale-aware trident networks for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6054–6063.
32. Zou, Z.; Shi, Z. Random access memories: A new paradigm for target detection in high resolution aerial remote sensing images. *IEEE Trans. Image Process.* **2017**, *27*, 1100–1111. [CrossRef]
33. Bai, Y.; Zhang, Y.; Ding, M.; Ghanem, B. Sod-mtgan: Small object detection via multi-task generative adversarial network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 206–221.
34. Hu, P.; Ramanan, D. Finding tiny faces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 951–959.
35. Mukhiddinov, M.; Cho, J. Smart glass system using deep learning for the blind and visually impaired. *Electronics* **2021**, *10*, 2756. [CrossRef]
36. Yuan, Y.; Xiong, Z.; Wang, Q. VSSA-NET: Vertical spatial sequence attention network for traffic sign detection. *IEEE Trans. Image Process.* **2019**, *28*, 3423–3434. [CrossRef]
37. Liu, Y.; Cao, S.; Lasang, P.; Shen, S. Modular lightweight network for road object detection using a feature fusion approach. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *51*, 4716–4728. [CrossRef]
38. Xiang, W.; Zhang, D.Q.; Yu, H.; Athitsos, V. Context-aware single-shot detector. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1784–1793.
39. Ouyang, W.; Wang, K.; Zhu, X.; Wang, X. Chained cascade network for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1938–1946.
40. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3578–3587.
41. Lyu, P.; Yao, C.; Wu, W.; Yan, S.; Bai, X. Multi-oriented scene text detection via corner localization and region segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7553–7563.
42. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 821–830.
43. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
44. Zoph, B.; Le, Q.V. Neural architecture search with reinforcement learning. *Int. Conf. Learn. Represent.* **2017**, 1–16 .
45. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 14–19 June 2020; pp. 10781–10790.
46. Ghiasi, G.; Lin, T.Y.; Le, Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7036–7045.
47. Narasimhan, S.G.; Nayar, S.K. Vision and the atmosphere. *Int. J. Comput. Vis.* **2002**, *48*, 233–254. [CrossRef]
48. Ranftl, R.; Bochkovskiy, A.; Koltun, V. Vision transformers for dense prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 12179–12188.
49. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
50. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
51. Chalavadi, V.; Jeripothula, P.; Datla, R.; Ch, S.B. mSODANet: A Network for Multi-Scale Object Detection in Aerial Images using Hierarchical Dilated Convolutions. *Pattern Recognit.* **2022**, *126*, 108548. [CrossRef]
52. Yu, W.; Yang, T.; Chen, C. Towards resolving the challenge of long-tail distribution in UAV images for object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021; pp. 3258–3267.
53. Wang, Y.; Yang, Y.; Zhao, X. Object detection using clustering algorithm adaptive searching regions in aerial images. In Proceedings of the ECCV, Glasgow, UK, 23–28 August 2020; pp. 651–664.

54. Liu, Z.; Gao, G.; Sun, L.; Fang, Z. HRDNet: High-resolution detection network for small objects. In Proceedings of the ICME, Shenzhen, China, 5–9 July 2021; pp. 1–6.
55. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios. In Proceedings of the ICCVW, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
56. Everingham, M.; Eslami, S.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
57. Jocher, G. YOLOv5. 2021. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 August 2022).