remote sensing

MDPI

Article

Small Object Detection Method Based on Adaptive Spatial Parallel Convolution and Fast Multi-Scale Fusion

Guanqiu Qi [1], Yuanchuan Zhang [2], Kunpeng Wang [3,*], Neal Mazur [1], Yang Liu [4] and Devanshi Malaviya [1]

[1] Computer Information Systems Department, State University of New York at Buffalo State, Buffalo, NY 14222, USA; qig@buffalostate.edu (G.Q.); mazurnm@buffalostate.edu (N.M.); malavidd01@mail.buffalostate.edu (D.M.)

[2] College of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; s190301049@stu.cqupt.edu.cn

[3] School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China

[4] BOE Technology Group Co., Ltd., Chongqing 400799, China; yangliu_cq@boe.com.cn

* Correspondence: kwang@swust.edu.cn

Abstract: As one type of object detection, small object detection has been widely used in daily-life-related applications with many real-time requirements, such as autopilot and navigation. Although deep-learning-based object detection methods have achieved great success in recent years, they are not effective in small object detection and most of them cannot achieve real-time processing. Therefore, this paper proposes a single-stage small object detection network (SODNet) that integrates the specialized feature extraction and information fusion techniques. An adaptively spatial parallel convolution module (ASPConv) is proposed to alleviate the lack of spatial information for target objects and adaptively obtain the corresponding spatial information through multi-scale receptive fields, thereby improving the feature extraction ability. Additionally, a split-fusion sub-module (SF) is proposed to effectively reduce the time complexity of ASPConv. A fast multi-scale fusion module (FMF) is proposed to alleviate the insufficient fusion of both semantic and spatial information. FMF uses two fast upsampling operators to first unify the resolution of the multi-scale feature maps extracted by the network and then fuse them, thereby effectively improving the small object detection ability. Comparative experimental results prove that the proposed method considerably improves the accuracy of small object detection on multiple benchmark datasets and achieves a high real-time performance.

Keywords: small object detection; adaptive spatial parallel convolution; multi-scale fusion

check for updates

Citation: Qi, G.; Zhang, Y.; Wang, K.; Mazur, N.; Liu, Y.; Malaviya, D. Small Object Detection Method Based on Adaptive Spatial Parallel Convolution and Fast Multi-Scale Fusion. *Remote Sens.* **2022**, *14*, 420. https://doi.org/10.3390/rs14020420

Academic Editors: Józef Lisowski, Kouzou Abdellah, Haitham Abu-Rub and Piotr Szymak

Received: 3 December 2021
Accepted: 12 January 2022
Published: 17 January 2022

1. Introduction

Since the advent of deep convolutional neural networks, the performance of object detection methods has been rapidly improving. At present, the representative object detectors, as the core components of various object detection methods, are mainly divided into two categories: (1) two-stage proposal-based detectors with the advantage of accuracy [1,2]; (2) single-stage proposal-free detectors with the advantage of speed [3,4]. Many recently proposed two-stage detectors [5–7] focus on improving the accuracy of object detection. Some single-stage detection frameworks, such as YOLO [8,9] and those using improved YOLO, are applied to different datasets such as MS COCO [10] and PASCAL VOC [11], and their performance is better than some two-stage detectors. Additionally, the real-time performance of these single-stage detectors shows an improvement over two-stage detectors. As an important objective evaluation indicator, the frames per second (FPS) of the real-time performance are generally greater than or equal to 30 [2,12]. Therefore, single-stage detectors [4,9,13] have been widely used in scenes with high real-time requirements.

Most of the current mainstream object detection frameworks have not made special improvements for small objects. However, a large number of cases involve small objects in actual scenes, such as recognizing a disaster victim in an unmanned aerial vehicle (UAV) search-and-rescue, and recognizing distant traffic signs and vehicles using autopilot. In this paper, both the training and testing stages of small object detection are implemented on images with a resolution range 51 × 72 $\leq$ *size* $\leq$ 4064 × 6354. According to the image resolution range used in this paper, the objects with a resolution of 32 × 32 or lower are generally called small objects. When the objects' resolution is 20 × 20 or lower, the corresponding objects are specifically called tiny objects. As shown in Figure 1, the absolute size represents the actual pixel size of the object in the image, and the relative size represents the ratio of the pixel size of the object in the image to the entire image. As shown in Figure 1a–c, TinyPerson [14], Tsinghua–Tencent 100K [15], and unmanned aerial vehicles' detection and tracking (UAVDT) [16] are three small object datasets, which contain a high number of UAV and autopilot object detection scenes, respectively. The resolution range of all the images in TinyPerson is 497 × 700 $\leq$ *size* $\leq$ 4064 × 6354. The resolution of all the images in both Tsinghua–Tencent 100K and UAVDT is 2048 × 2048 and 1024 × 540, respectively. The resolution range of all the images in MS COCO is 51 × 72 $\leq$ *size* $\leq$ 640 × 640. As shown in Figure 1d,e, when objects have a small absolute or relative size, the object detection performance of existing detectors decreases to a certain extent. Many small object detection methods have been proposed to meet the needs of practical applications. Most of them were developed based on the improvement of existing object detection methods. Additionally, these developed methods mainly focus on improving the accuracy of small object detection. However, a high real-time performance of detectors is usually necessary in small object detection scenes.

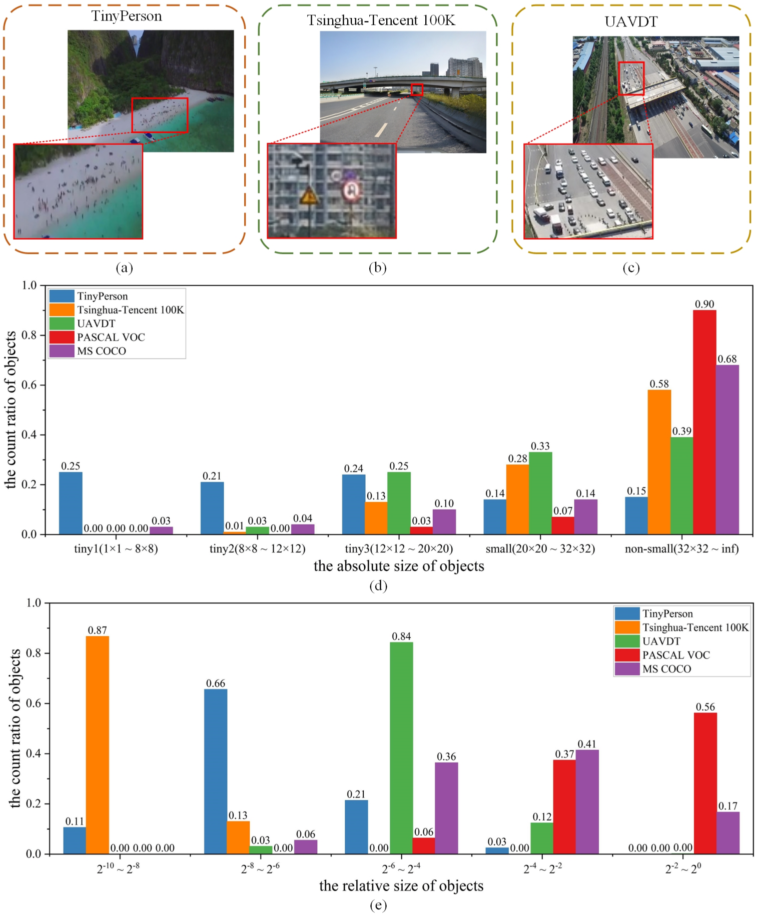


Figure 1. Dataset visualization analysis. (**a**–**c**) are sample images of TinyPerson, Tsinghua–Tencent 100K, and UAVDT benchmark datasets, respectively. (**d**,**e**) are the statistical histograms of the absolute and relative size of objects in the datasets.

Positioning and classification are two main object detection subtasks [17]. Therefore, object detection should not only accurately locate all objects in an image, but also correctly identify their categories. Object detection tasks usually require the spatial and semantic information extracted from neural networks to assist in object positioning and classification [18,19]. However, due to the inconspicuous/weak features of small objects, small object detection needs to be optimized for the following two aspects. First, the existing research results show that the surrounding environment is essential for humans to recognize small objects [20]. In object detection, local context information represents the visual information of the area around the object to be detected [21]. Additionally, the experimental results of existing computer vision research show that the proper modeling of the spatial background can improve the accuracy of object detection [22]. Therefore, the existing methods capture the local context information of an object through a relatively large receptive field, thereby trying to obtain the abundant fine-grained spatial information of the object [23–26]. However, the excessive use of large-scale convolution kernels with large receptive fields causes an increase in the time and space complexity of detection models, which is not conducive for single-stage detectors to achieve real-time performance. Second, due to their small size, the spatial information of small objects usually disappears in the feature transmission process. In neural networks, image features are gradually transmitted to deep layers. Additionally, the corresponding image size simultaneously decreases. If any relevant processing is not applied to the small objects shown in the image, the related small objects disappear in the feature transmission process. The multi-scale fusion of feature maps between different network levels is an effective way to solve the above issue [21]. For example, some existing solutions, such as [6,27–29], generally adopt a top-down path to construct a feature pyramid [27], thereby alleviating feature disappearance to a certain extent. Additionally, a feature pyramid can be used to fuse both the spatial and semantic feature information, which can optimize small object detection.

This paper proposes a SODNet composed of an adaptively spatial parallel convolution module (ASPConv) and fast multi-scale fusion module (FMF) to optimize both the extraction of spatial information and fusion of spatial and semantic information, thereby achieving real-time processing. ASPConv is used to adaptively extract features by using multi-scale receptive fields. FMF optimizes both the semantic and spatial information of output features to achieve feature map upsampling and multi-scale feature fusion. Additionally, the real-time-related factors are considered in both modules to ensure the high real-time performance of the proposed SODNet.

The proposed SODNet is applied to four public datasets, TinyPerson, Tsinghua-Tencent 100K, UAVDT and MS COCO. According to the comparative experimental results, the proposed SODNet can effectively improve the accuracy of small object detection in real-time. This paper has three main contributions, as follows:

- This paper proposes an adaptive feature extraction method using multi-scale receptive fields. Due to the small proportion in the image and inconspicuous features, the spatial information of small objects is always missing. The proposed method divides the input feature map equally among the channels and performs feature extraction on the separated feature channels in parallel. Additionally, the cascading relationship of multiple convolution kernels is used to achieve the effective extraction of local context information for different channels. Therefore, the features related to small objects with multi-scale spatial environmental information can be obtained by fusing the extracted information.
- This paper proposes a new feature map upsampling and multi-scale feature fusion method. This method uses both nearest-neighbor interpolation and sub-pixel convolution algorithm to map a low-resolution feature map with rich semantic information to a high-resolution space, thereby constructing a high-resolution feature map with rich semantic features. A feature map with sufficient spatial and semantic information is obtained by the fusion of the constructed feature map and a feature map with rich spatial information, thereby improving the detection ability of small objects.

- This paper designs a one-stage, real-time detection framework of small objects. The ASPConv module is proposed to extract image features from multiple channels in parallel, which effectively reduces the time complexity of feature extraction to achieve real-time small object detection. The FMF module is proposed to apply both nearest-neighbor interpolation and sub-pixel convolution to achieve a fast upsampling. The processing time of multi-scale feature map fusion is reduced by improving upsampling efficiency to ensure real-time small object detection.

The rest of this paper is organized as follows: Section 2 discusses the related work; Section 3 describes the details of the proposed method, including the implementation of both ASPConv and FMF modules; Section 4 presents the experimental results, a comparative analysis, and an ablation study on TinyPerson, Tsinghua-Tencent 100K, UAVDT and MS COCO datasets to verify the effectiveness of the proposed method; and Section 5 concludes the paper.

2. Related Work

With the rapid development of deep learning, the performance of object detection methods was accordingly improved. Object detectors are usually classified into two categories: two-stage detectors [2,5–7] and single-stage detectors [4,8,9,13]. Although most of the existing methods achieved a relatively good object detection performance, they do not have any specialized optimization for small objects. However, most existing small object detection methods were proposed based on the optimization of conventional object detection methods. Generally speaking, small object detection methods usually optimize the following aspects:

Feature extraction. Well-designed convolution modules can adaptively extract the rich feature information of small objects in complex scenes.Dilated convolution [24,30] controls the size of receptive fields by changing the sampling center distance. Receptive field block net (RFB) [23] introduces a dilated convolution on the basis of inception [25], and strengthens the network extraction ability by simulating the receptive fields of human vision. Deformable convolution [31] adaptively learns the unique resolution of a single object to adapt it to multi-scale features. Selective kernel networks (SKNet) [32] design a selection module to adaptively adjust the size of receptive fields according to the multi-scale input information. These methods use a specifically designed convolution module to make receptive fields rich enough to adjust for the inconsistency in object size. However, they do not focus on improving the detection performance of small objects and ignore the importance of spatial information.

High-resolution features. Since small objects are difficult to find and locate, spatial information is necessary, which can easily be obtained from high-resolution feature maps. Li et al. [33] proposed a feature-level super-resolution method, specialized for small object detection, which used the features of large objects to enhance the features of small objects through Perceptual GAN. Noh et al. [34] first applied super-resolution techniques to enhance the region of interest (RoI) features of small objects. Then, appropriate high-resolution object features were used as supervision signals in the model training process to enhance the model's small object learning ability. The efficient sub-pixel convolutional neural network (ESPCN) [35] performs super-resolution reconstruction through sub-pixel convolution and uses a series of convolution operations to reconstruct low-resolution features into high-resolution features, to achieve the purpose of upsampling. A reference-based method proposed by Zhang et al. [36] uses the rich texture information of high-resolution reference images to compensate for the missing details in low-resolution images. Although these methods are helpful when obtaining high-resolution feature maps, they do not fully fuse the spatial and semantic information in the features. Therefore, it is still difficult to detect small objects.

Multi-scale feature maps. Many studies have proved that the fusion of multi-scale feature maps is also conducive to small object detection. Due to the slow speed of image pyramids and high memory consumption, the current mainstream object detection methods use feature pyramid structures to achieve cross-scale connections and fusion. Feature pyramid networks (FPN) [27,37] use a top-down method to build one feature pyramid and fuse the features of different scales to improve the multi-scale detection performance. The path aggregation network (PANet) [28] adds a bottom-up path based on FPN to transmit detailed spatial information. Gated feedback refinement network (G-FRNet) [38] is a gated feedback optimization network. It adds a gated structure based on FPN. It uses the feature layer with rich semantic information to filter the fuzzy information and ambiguity in the feature layer with rich spatial information. Therefore, the transmission of key information is achieved to block ambiguous information. The bi-directional feature pyramid network (BiFPN) structure used in an efficient detection net [6] overlaps the optimized FPN in a repeated form to improve feature richness. Although these feature pyramid-like structures improve the performance of multi-scale detection, they cannot improve the detection performance on small objects.

Therefore, this paper proposes the ASPConv module to perceive the local context of the target through multi-scale receptive fields and enrich the spatial information. In addition, this paper proposes the FMF module to integrate high-resolution feature map generation and multi-scale feature map fusion. Both spatial and semantic feature information is fused. The suitable high-resolution feature maps are generated for small object detection.

3. The Proposed Method

Figure 2 shows the structure of the proposed SODNet. In the feature extraction stage, the detailed spatial information of small objects is first extracted by the ASPConv module. Then, further feature extraction is performed to obtain the feature maps $C_i, i = 1,2,3,4$ that were downsampled by the sub-backbone in the proposed backbone module. The obtained feature maps contain features from different network levels and can be used in the subsequent feature fusion (shown in Table 1). In the feature fusion stage, the element-by-element addition to the horizontal connection of FPN is replaced by a concatenation operation and the fine-tuned FPN structure is applied to the fusion of feature maps to obtain the new feature maps P_2, P_3, and P_4. Specifically, the feature map P_4 is obtained by the convolution of the feature map C_4, the feature map P_3 is obtained by fusing the feature maps P_4 and C_3, and the feature map P_2 is obtained by fusing the feature maps P_3 and C_2. Additionally, the FMF module maps the low-resolution feature maps (P_2, P_3, and P_4) with rich semantic information to the high-resolution space, and integrates the feature map C_1 with rich spatial information from the ASPConv module to generate the high-resolution feature map P_1 with rich semantic and spatial information. Therefore, the obtained feature maps $P_j, j = 1,2,3,4$ can be used to improve the detection ability of small objects.

As shown at the bottom of Figure 2, the predictor uses four independent convolution units to perform positioning and classification. The feature maps $P_j, j = 1,2,3,4$ are processed to obtain the detection results [9]. The size and step length of the convolution kernel of each convolution unit are 1×1 and 1, respectively. The predictor predicts the categories and bounding boxes of all objects in the corresponding feature maps through these convolution units. In the training stage, the categories and bounding boxes obtained by the predictor and ground-truth are classified and regressed. The loss of each image is calculated, and the network weights are continuously updated through backpropagation until the model converges. In the inference stage, threshold filtering and non-maximum suppression are applied to the classification and positioning results obtained by the predictor to eliminate overlapping or abnormal bounding boxes and obtain the final detection result. This section may be divided by subheadings. It should provide a concise and precise description of the experimental results and their interpretation, as well as the experimental conclusions that can be drawn.

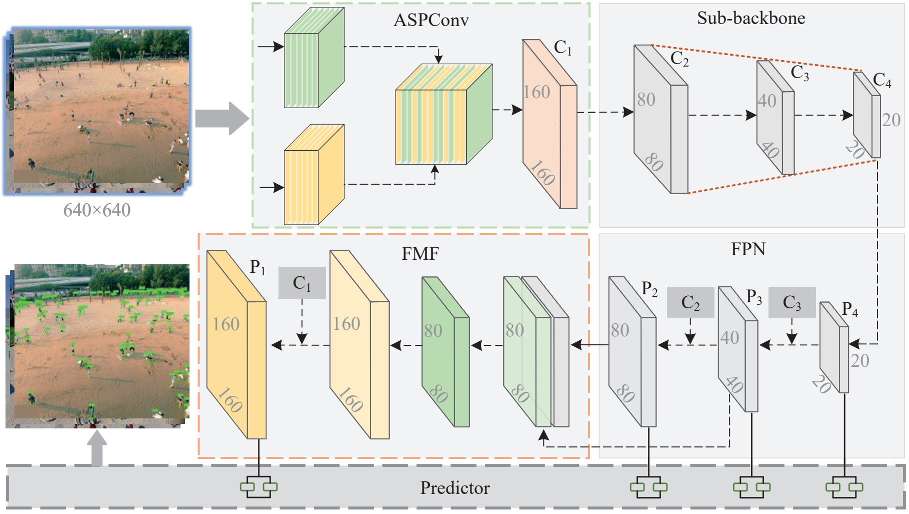


Figure 2. The structure of SODNet. SODNet consists of four components. (1) ASPConv module (shown in Section 3.1) extracts rich spatial information of objects from the input image. (2) The proposed backbone module (shown in Section 3.2) is composed of two components: sub-backbone and FPN. The sub-backbone further extracts the output features of the ASPConv module and generates multi-scale feature maps through FPN. (3) FMF module (shown in Section 3.3) quickly fuses both semantic and spatial information in multi-scale feature maps to generate high-resolution feature maps that are conducive to small object detection. (4) Predictor (shown in Section 3.4) classifies and locates the fused multi-scale feature maps.

3.1. Adaptively Spatial Parallel Convolution Module

As shown in Figure 3, this paper proposes an adaptively spatial parallel convolution module to enable neurons to adaptively learn the spatial information of objects in the original image. This module can associate with local context to obtain rich spatial information. *Conv* module consists of *Conv*2*d*, Bacth Normalization [39] and Hardswish function [40]. *Conv*2*d* represents a standard convolution operation. k represents the size of the convolution kernel. s is the stride (unless specified, the default size of the convolution kernel is 3×3; the default stride is 1). C represents the channel number of a feature map. ASPConv module first downsamples the original image to obtain the feature map X. Then, the obtained feature map X is equally divided on the channel to obtain feature maps X_1 and X_2. Next, the obtained feature maps X_1 and X_2 are convolved in a parallel manner. Additionally, the cascading relationship of multiple convolutions is used to realize the effective extraction of the local context of the objects in the feature maps X_1 and X_2. Subsequently, all the information is fused by a jumping connection [41] to obtain the detailed spatial information that is conducive to the detection of small objects. Finally, downsampling and further feature extraction are performed on the fused feature map to obtain the feature map C_1.

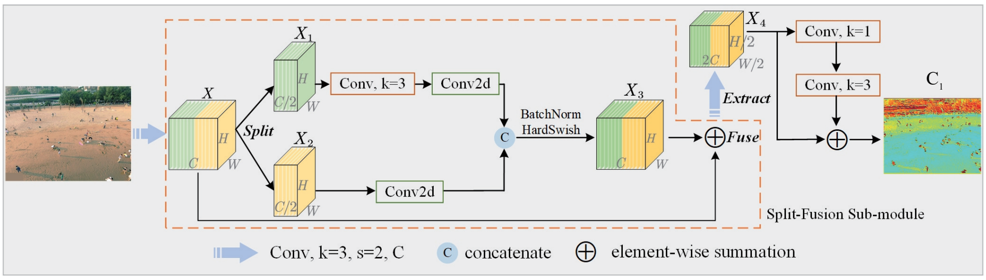


Figure 3. The structure of ASPConv. The multi-scale local context information of small objects is first adaptively learned in a split-fusion sub-module, and then fused to form detailed spatial information.

Local context information. The size of receptive fields (RF) is important for the recognition of small objects. The size of the area mapped on the original image is defined through the pixels on the feature map output by each convolutional neural network layer [42] as follows:

$$RF_{i+1} = RF_i + (k-1) \times \prod_{i=1}^{n} Stride_i \tag{1}$$

where RF_{i+1} represents the receptive field of the (i+1)-th layer (the current layer), RF_i represents the receptive field of the i-th layer (the previous layer), $Stride_i$ represents the stride in the i-th layer of convolution or pooling operation, and k represents the convolution kernel size of the (i+1)-th layer (the current layer). According to the calculation of Equation (1), the pixel points on the feature map have a mapping area size of 3, 7, and 11 on the original image, respectively, after the convolution of X, X_1, and X_2. Therefore, the parallel convolution operations in the ASPConv module can capture the multi-scale local context information. In addition, unlike some existing structures [24,25], the ASPConv module captures rich local context information through the multi-convolution cascading relationship of the SF sub-module. Additionally, the reduction in feature maps in the SF sub-module also effectively reduces the time complexity of ASPConv.

Split. Given any feature map $X \in \mathbb{R}^{C \times H \times W}$, two transformations are first performed on the feature map, and then the feature map is divided into two components, $\mathcal{F}_1 : X \to X_1 \in \mathbb{R}^{\frac{C}{2} \times H \times W}$ and $\mathcal{F}_2 : X \to X_2 \in \mathbb{R}^{\frac{C}{2} \times H \times W}$. The transformation $\mathcal{F}_1$ consists of $Conv2d$, Batch Normalization [39], and Hardswish function [40]. The transformation $\mathcal{F}_2$ is composed of $Conv2d$. To further improve the efficiency, two convolutions with 3 × 3 kernel size are used to replace the convolution with 5 × 5 kernel size on the channel of the feature map X_1, so the amount of both the calculation and parameters can be reduced under the same receptive field conditions [43].

Fuse. The multi-scale local context information of objects is adaptively learned and fused in a different receptive field size to construct the spatial position relationship between objects and environment. Therefore, the spatial information that is conducive to the detection of small objects can be obtained. The calculation process of the fused feature map X_3' is given as follows:

$$X_3' = \delta(\mathcal{B}(X_3)) + X \tag{2}$$

where $\delta(\cdot)$ and $\mathcal{B}(\cdot)$ represent Hardswish function and Batch Normalization, respectively, and $X_3 \in \mathbb{R}^{C \times H \times W}$. X_3 is the intermediate result of the feature maps X_1 and X_2 after parallel convolution and concatenation operation, which is defined as follows.

$$X_3 = Cat[\mathcal{F}_1, \mathcal{F}_2] = Cat[\mathcal{C}(\mathcal{G}(X_1)), \mathcal{C}(X_2)] \tag{3}$$

where $Cat[\cdot]$ represents a concatenation operation, $\mathcal{C}(\cdot)$ represents a standard convolution operation with a 3 × 3 kernel size and a stride of 1, and $\mathcal{G}(\cdot) = \delta(\mathcal{B}(\mathcal{C}(\cdot)))$.

Extract. After the $Split$ and $Fuse$ operations, the ASPConv module effectively fuses the multi-scale local context information from the same layer, enriches the spatial information and obtains the fused feature map X_3'. Subsequently, the feature map X_3' is downsampled to reduce its resolution and a further feature extraction is performed to obtain the output feature map X_{out} of the ASPConv module, which is defined as follows:

$$X_4 = \mathcal{G}(X_3') \tag{4}$$

$$X_{out} = X_4 + \mathcal{G}(\mathcal{G}(X_4)) \tag{5}$$

Finally, the ASPConv module effectively extracts and fuses the multi-scale local context information through operations $Split$, $Fuse$, and $Extract$, and obtains a feature map X_{out}

with rich spatial information, which corresponds to the feature map C_1 in Figure 2. In addition, the feature map C_1 is applied to the multi-scale feature map fusion used in the FMF module (Section 3.3) to improve the small object detection ability of the proposed model.

3.2. Proposed Backbone Module

As shown in Figure 4, the proposed backbone module consists of a sub-backbone and FPN. The sub-backbone is mainly improved by the cross-stage partial darknet (CSP-DarkNet) [9] to balance the speed and accuracy. The original structure of CSPDarkNet is $C_i, i = 1,2,3,4$ shown in Table 1. In the sub-backbone, the structure of the feature map C_1 is changed to the proposed ASPConv module. Therefore, feature maps with rich spatial information can be obtained, which are conducive to small object detection. In addition, as shown in Table 1, the original PANet [28] is replaced by FPN [27] in the multi-scale feature map ($P_j, j = 2,3,4$) generation stage. The specific implementation details of FPN are shown in Figure 4.

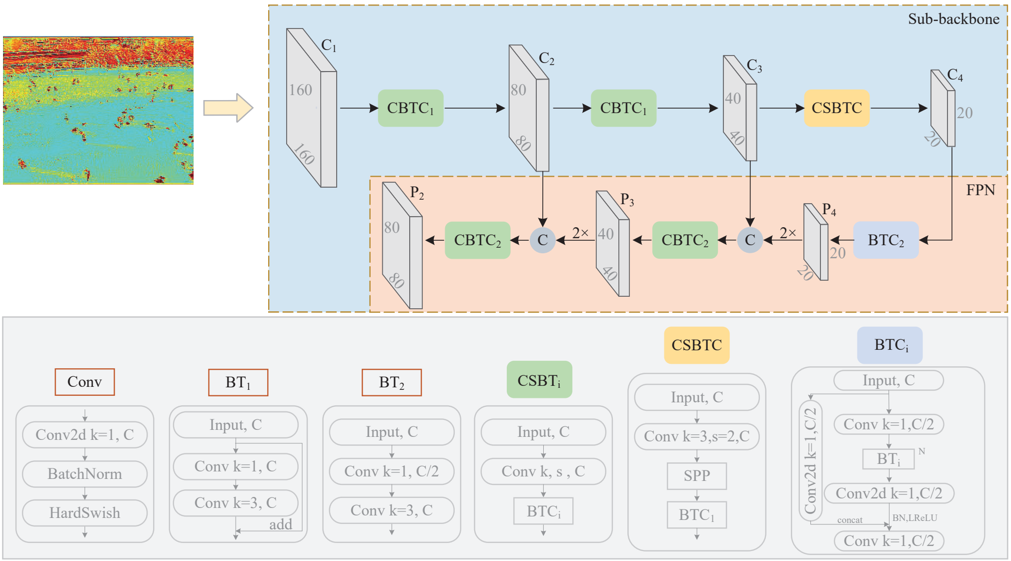


Figure 4. The structure of the proposed backbone module.

Table 1. Structure comparison between Yolov5s [9] and the proposed SODNet. C_i and P_j are consistent with the definitions in Figure 2. $CBTC_1$ and $CSBTC$ are consistent with the definitions in Figure 4.

Layer Name	**Layer Components**	
	Yolov5s [9]	**SODNet (Proposed)**
C_1	Focus [9] CBTC1 ($BT_i \times 1$)	ASPConv (Proposed)
C_2	$CBTC_1$ ($BT_i \times 3$)	$CBTC_1$ ($BT_i \times 3$)
C_3	$CBTC_1$ ($BT_i \times 3$)	$CBTC_1$ ($BT_i \times 3$)
C_4	$CBTC_1$ ($BT_i \times 1$)	$CBTC_1$ ($BT_i \times 2$)
$P_2/P_3/P_4$	PANet [28]	FPN [27]
P_1	-	FMF (Proposed)

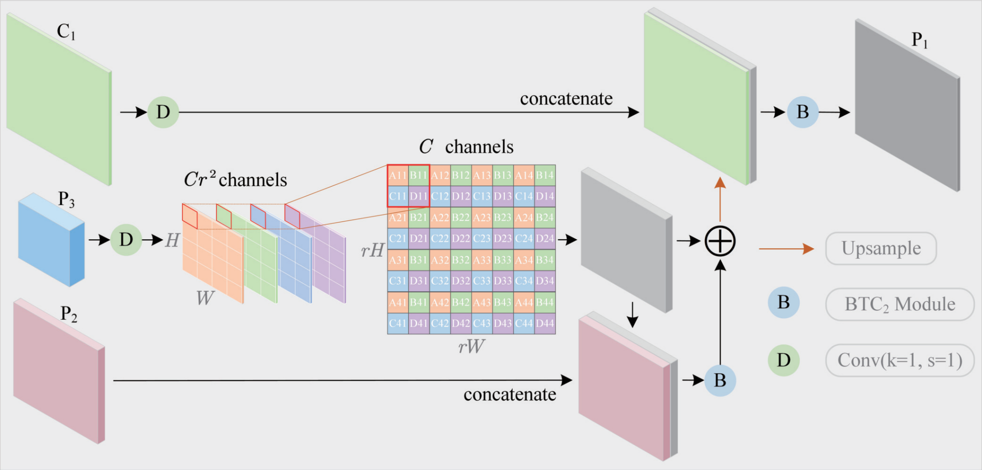
In Figure 4, the sub-backbone is composed of the $CBTC_i$ and $CSBTC$ modules, respectively. They all consist of some convolution units and feature extraction units, in which the BTC_i module obtained after CSP-related operations is the basic component of $CBTC_i$ and

$CSBTC$ modules. N feature extraction units of BT_i integrated into the BTC_i module. The number of BT_i owned by C_i (C_2, C_3, and C_4) is adjusted to 3, 3, and 2, respectively. For the middle layer of $CBTC_1$ module, $Conv$ module, as its component, is set to $k = 3, s = 2$ to achieve the downsampling according to the parameter settings in [9]. The $Conv$ module of $CBTC_2$ in FPN is set to $k = 3, s = 1$ to further extract features and eliminate the aliasing effects that may be caused by feature fusion. In addition, the module $CSBTC$ consists of convolution units and SPP [44]. SPP is applied to pool and cascade multi-scale local area features. Meanwhile, global and local multi-scale features are used to improve the detection accuracy. The parameter settings of max pooling in SPP are the same as the corresponding ones used in [9]. The core size of max pooling is set to 1×1, 5×5, 9×9, and 13×13, respectively, in the following experiments in this paper. Finally, the proposed backbone module improves the sub-backbone with a low computational cost to increase the inference speed of the overall model and enhance the feature extraction ability. Meanwhile, the feature maps P_2, P_3, and P_4 required for fusion and detection are generated through FPN.

3.3. Fast Multi-Scale Fusion Module

The fusion of multi-scale feature maps is conducive to the detection of small objects. In addition, the effective spatial information of small objects usually exists in the feature map C_1 [45]. A fast multi-scale fusion module is proposed to fuse multi-scale feature maps and generate high-resolution feature maps with rich semantic and spatial information, thereby improving the detection ability of small objects.

As shown in Figure 5, the FMF module uses the sub-pixel convolutional layer [35] to learn an ascending filter array, upscales the feature map P_3 to a high-resolution space, and automatically learns the interpolation function in the transformation process from low-resolution to high-resolution through the previous convolutional layers. The sub-pixel convolution is highly efficient, dispersing pixels in the channel dimension and adding pixels in the width and height dimensions. The feature dimension of the input sub-pixel convolutional layer is $\mathcal{L} \in \mathbb{R}^{Cr^2 \times H \times W}$. Therefore, the feature dimension is $\mathcal{H} \in \mathbb{R}^{C \times rH \times rW}$ after rearrangement. The corresponding operation can be formalized as follows.



A11	B11	A12	B12	A13	B13	A14	B14
C11	D11	C12	D12	C13	D13	C14	D14
A21	B21	A22	B22	A23	B23	A24	B24
C21	D21	C22	D22	C23	D23	C24	D24
A31	B31	A32	B32	A33	B33	A34	B34
C31	D31	C32	D32	C33	D33	C34	D34
A41	B41	A42	B42	A43	B43	A44	B44
C41	D41	C42	D42	C43	D43	C44	D44

Figure 5. The structure of FMF. The low-resolution feature maps P_2 and P_3 with rich semantic information are mapped to the high-resolution space and fused with the feature map C_1 with rich spatial information to generate a high-resolution feature map P_1 with rich semantic and spatial information. P_2 and P_3 are generated by FPN shown in Figure 4. C_1 is the output feature map in the APSConv module.

$$\mathcal{SPC}(\mathcal{T})_{x,y,c} = \mathcal{T}_{\lfloor \frac{x}{r} \rfloor, \lfloor \frac{y}{r} \rfloor, C\cdot r\cdot mod(y,r)+C\cdot mod(x,r)+c} \tag{6}$$

where $\mathcal{SPC}(\mathcal{T})_{x,y,c}$ is the output pixel value on the spatial coordinates (x, y, c) after the pixel scatter operation $\mathcal{SPC}(\cdot)$, and r is the upscaling ratio. The proposed method uses $r = 2$ to double the spatial scale of feature map P_3 for fusion with the feature map P_2.

The FMF module first concatenates the feature map P_2 with $\mathcal{H}$ on the channels, then passes through the BTC_2 module to eliminate the aliasing effects that may be caused by concatenation, and further extracts the fused feature information. At the same time, an element-wise addition method is adopted to ensure that the output $\hat{\mathcal{H}}$ fuses the semantic and regional information from feature maps P_2 and P_3 as follows.

$$\hat{\mathcal{H}} = f_{BTC_2}(Cat[\mathcal{H}, \mathcal{P}3]) + \mathcal{H} \tag{7}$$

$$\mathcal{H} = \mathcal{SPC}(f_{Conv}(\mathcal{P}_4)) \tag{8}$$

where f_{Conv} and f_{BTC_2} represent the $Conv$ module in Figure 3 and BTC_2 module in Figure 4 respectively, and $\hat{\mathcal{H}} \in \mathbb{R}^{C \times rH \times rW}$. $\mathcal{SPC}$ is first used to perform the channel-to-space transformation, and then f_{BTC_2} is used to enhance the transformation in the spatial range. Finally, the high-resolution feature map P_1 is generated by fusing semantic and spatial information, as follows:

$$P_1 = f_{BTC_2}(Cat[\mathcal{NNI}(\hat{\mathcal{H}}), f_{Conv}(\mathcal{C}_2)]) \tag{9}$$

where $\mathcal{NNI}(\cdot)$ represents the nearest neighborhood interpolation operation, and $P_1 \in \mathbb{R}^{\frac{C}{2} \times 2rH \times 2rW}$. Similarly, $\mathcal{NNI}$ is first used to transform the input features in space, and then f_{BTC_2} is used to further spread their spatial influence.

In the FMF module, $\mathcal{SPC}$ and $\mathcal{NNI}$ are alternately used in upsampling to achieve the fusion of semantic and spatial information. Therefore, the loss of detailed information on small objects caused by the too-high upsampling rate can effectively be avoided. The convolution kernel of 1×1 or 3×3 size is used in the FMF module. Compared with large-size convolution kernels, the time complexity of the FMF module is lower.

3.4. Predictor

SODNet finally inputs four feature maps P_1, P_2, P_3, and P_4 to the predictor to detect the classification and positioning. According to the existing research [3,9,14], the loss function specialized for classification and positioning in SODNet mainly includes three components: location loss, confidence loss, and classification loss.

The position loss is the error between the predicted bounding boxes and the ground-truth, which is calculated using the generalized intersection over union (GIoU) [46] loss function. Assuming that the coordinates of bounding boxes and ground-truth are $B^p = \left(x_1^p, y_1^p, x_2^p, y_2^p\right)$ and $B^g = \left(x_1^g, y_1^g, x_2^g, y_2^g\right)$, respectively, the area of B^p and B^g can be calculated as follows.

$$A^p = \left(\hat{x}_2^p - \hat{x}_1^p\right) \times \left(\hat{y}_2^p - \hat{y}_1^p\right) \tag{10}$$

$$A^g = \left(x_2^g - x_1^g\right) \times \left(y_2^g - y_1^g\right) \tag{11}$$

where A^p and A^g represent the area of B^p and B^g, respectively, $\hat{x}_1^p = \min\left(x_1^p, x_2^p\right)$, $\hat{x}_2^p = \max\left(x_1^p, x_2^p\right)$, $\hat{y}_1^p = \min\left(y_1^p, y_2^p\right)$, and $\hat{y}_2^p = \max\left(y_1^p, y_2^p\right)$. The overlapping area I of B^p and B^g is obtained as follows:

$$I = \begin{cases} \left(x_2^I - x_1^I\right) \times \left(y_2^I - y_1^I\right) & x_2^I > x_1^I, y_2^I > y_1^I \\ 0 & otherwise \end{cases} \tag{12}$$

where $x_1^I = \max\left(\hat{x}_1^p, x_1^g\right)$, $x_2^I = \min\left(\hat{x}_2^p, x_2^g\right)$, $y_1^I = \max\left(\hat{y}_1^p, y_1^g\right)$, and $y_2^I = \min\left(\hat{y}_2^p, y_2^g\right)$. Additionally, the minimum area A^c of the bounding box containing B^p and B^g can be calculated as follows:

$$A^c = (x_2^c - x_1^c) \times (y_2^c - y_1^c) \tag{13}$$

where $x_1^c = \min\left(\hat{x}_1^p, x_1^g\right)$, $x_2^c = \max\left(\hat{x}_2^p, x_2^g\right)$, $y_1^c = \min\left(\hat{y}_1^p, y_1^g\right)$, and $y_2^c = \max\left(\hat{y}_2^p, y_2^g\right)$. The Intersection over Union (IoU) of B^p and B^g is shown as follows:

$$IoU = \frac{I}{U} \tag{14}$$

where I as the area intersection of the coordinates B^p and B^g is obtained by Equation (12), U is the area union of the coordinates B^p and B^g, and $U = A^p + A^g - I$. So, $GIoU$ can be calculated as follows:

$$GIoU = IoU - \frac{A^c - U}{A^c} \tag{15}$$

The final position loss L_{GIoU} can be obtained by $GIoU$ as follows:

$$L_{GIoU} = 1 - GIoU \tag{16}$$

Confidence loss L_{conf} is the relative error of the object confidence score prediction [3], which can be calculated as follows:

$$L_{conf} = -\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\log(C_{ij}) - \lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{noobj}\log(1 - C_{ij}) \tag{17}$$

where S is the side length of the feature map that is input into the predictor; B is the number of anchors in each cell of the feature map; λ_{noobj} as the balance coefficient is set to 0.5 during training; I_{ij}^{obj} indicates whether the j-th anchor in the i-th cell is responsible for the probability of falling into the area's bounding box. If it is responsible, the value of I_{ij}^{obj} is 1, otherwise the value of I_{ij}^{obj} is 0. The definition and value of I_{ij}^{noobj} are opposite to that of I_{ij}^{obj}; C_{ij} is the confidence score of the j-th anchor in the i-th cell predicted by the proposed SODNet, and $C_{ij} \in [0, 1]$.

The classification loss L_{cls} is the error between the predicted category of the object and the corresponding true category [3], which is calculated as follows.

$$L_{cls} = -\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\sum_{c\in cls}\left[\hat{p}_{ij}^c\log\left(p_{ij}^c\right) + \left(1 - \hat{p}_{ij}^c\right)\log\left(1 - p_{ij}^c\right)\right] \tag{18}$$

where p_{ij}^c is the predicted category of the j-th anchor in the i-th cell, and $\hat{p}_{ij}^c$ is the true category. Therefore, the total loss L of SODNet is obtained as follows.

$$L = L_{GIoU} + L_{conf} + L_{cls} \tag{19}$$

In the training stage, SODNet continuously optimizes the loss L and updates the network weights through backpropagation until the model converges. In the testing stage, SODNet does not perform backpropagation. It directly performs post-processing operations, such as confidence threshold screening and non-maximum suppression processing, on the classification and positioning results obtained by the predictor to obtain the final detection results.

4. Experiments

This section evaluates the performance of the proposed SODNet on the benchmark datasets of TinyPerson [14], Tsinghua–Tencent 100K [15], UAVDT [16] and MS COCO [10]. In the following experiments, Yolov5s [9] is used as the baseline. Both the training and testing of comparative experiments were conducted on a server equipped with 4 Nvidia Tesla P100 16 G, 256 G memory, and an Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20 GHz. First, this section introduces the basic information of two datasets and the corresponding evaluation indicators, and specifies the experimental details. Then, the proposed method is compared with the two-stage detectors, FRCNN-FPN [27], Libra RCNN [5], Grid RCNN [7], FRCNN-FPN-SM [14], FRCNN-FPN-SM $S-\delta$ [47], and single-stage detectors, FCOS [48], SSD512 [4], FS-SSD512 [49], RetinaNet [13], RetinaNet-MSM [14], RetinaNet-SM $S-\delta$ [47], Scaled-YOLOv4-CSP [8] on the TinyPerson dataset. RCNN-FPN-SM [14], FRCNN-FPN-SM $S-\delta$ [47], FS-SSD512 [49], RetinaNet-MSM [14], RetinaNet-SM $S-\delta$ [47] are specifically optimized methods for small objects. The processing efficiency of the proposed method is compared with the processing efficiency of part of the above detectors [4,8,9,14,27,49] with published source codes. On the Tsinghua–Tencent 100K dataset, the proposed method is compared with the methods that were especially optimized for small objects, such as FRCNN [2] +ResNet101 [41], Zhu et al. [15], Perceptual GAN [33], EFPN [45], SOS-CNN [50], and Noh et al. [34]. On the UAVDT dataset, the proposed method is compared with R-FCN [1], SSD [4], RON [51], FRCNN [2], FRCNN-FPN [27], and ClusDet [52]. On the MS COCO dataset, the proposed method is compared with R-FCN [1], SSD [4], YOLOv3 [3], FRCNN [2], FRCNN-FPN [27], and Noh et al. [34]. In addition, an ablation study is carried out for both of the proposed modules, ASPConv and FMF and baseline. Finally, a qualitative analysis of the proposed SODNet and baseline is given.

4.1. Experiment Preparation

4.1.1. Datasets and Evaluation Metrics

TinyPerson. TinyPerson [14] is a small object benchmark dataset containing a high number of small objects. All the images were collected from real-world scenes by unmanned aerial vehicles (UAVs). TinyPerson contains 1610 images with 72,651 labeled frames, of which 794 and 816 are used as training and testing images, respectively. The objects in TinyPerson are very small. According to the area occupied by each object, the objects are divided into tiny1 ($area \leq 8 \times 8$), tiny2 ($8 \times 8 < area \leq 12 \times 12$), tiny3 ($12 \times 12 < area \leq 20 \times 20$), tiny ($area \leq 20 \times 20$) consisting of tiny1, tiny2 , and tiny3, small ($20 \times 20 < area \leq 32 \times 32$), and non-small ($area > 32 \times 32$) objects. As shown in Figure 1, their corresponding proportions are 25.2%, 21.4%, 24.4%, 71%, 14.0%, and 15.0%, respectively. The object proportion of the interval $area \leq 32 \times 32$ in the TinyPerson dataset is 85%. This means that the TinyPerson dataset can be used to evaluate the small object detection performance of the proposed model. Therefore, both real-time testing experiments and an ablation study were carried out on this dataset.

According to Tiny Benchmark [14], average precision (AP) and miss rate (MR) are used to evaluate the performance of object detection. AP, as a widely used evaluation indicator in object detection, reflects the precision and recall of detection results. When the value of AP increases, the detector performance improves. MR is usually used in pedestrian datasets. It reflects the object loss rate. When the value of MR decreases, the detector performance improves. The comparative experiments were implemented over five intervals of small and tiny objects respectively, including tiny1, tiny2, tiny3, tiny, and small. A detailed analysis is provided. The threshold of intersection over union (IoU) was set to 0.25, 0.5, and 0.75, and both MR^{tiny} and AP^{tiny} at $IoU = 0.5$ were used as the main indicators to evaluate the small object detection performance in the TinyPerson dataset [14]. $IoU = 0.5$ means that when IoU ratio between bounding boxes and ground-truth in the detection result was greater than or equal to 0.5, the detection was correct [34]. The objects in the TinyPerson dataset are quite small. When the value of IoU exceeds 0.5, the detector performance dropped considerably. Therefore, only three IoU values, 0.25, 0.5, and 0.75, were selected to evaluate the detection performance of small objects on the TinyPerson dataset, instead of an IoU interval.

Tsinghua-Tencent 100K. Tsinghua–Tencent 100K [15] is a large-scale traffic sign benchmark dataset, which contains 100,000 high-resolution (2048 × 2048) images and 30,000 traffic sign instances. According to the area occupied by each object, Tsinghua–Tencent 100K divides objects into smaller objects ($area \leq 32 \times 32$ pixels), medium objects ($32 \times 32 < area \leq 96 \times 96$ pixels), and large objects ($area > 96 \times 96$ pixels). The original division of objects is applied to the following experiments on the Tsinghua–Tencent 100K dataset [15]. The proportions of small, medium, and large objects are 42%, 50%, and 8%, respectively. Since small and medium objects are dominant in this dataset, it is also a good benchmark to evaluate the performance of small object detection.

According to the protocols in Tsinghua–Tencent 100K [15], classes with fewer than 100 instances are ignored. A total of 45 classes were finally selected for evaluation, and accuracy and recall were used as evaluation indicators. Additionally, F_1 score was also used as an evaluation indicator. In the Tsinghua–Tencent 100K experiments, when IoU (the ratio of bounding boxes to ground-truth) was greater than or equal to 0.5, the corresponding detection was considered successful [34].

UAVDT. UAVDT [16] is a large-scale challenging benchmark dataset. It contains about 80,000 frames of images with annotated information. It is used to achieve three basic computer vision tasks (object detection, single-object tracking, and multiple-object tracking). For the object detection, the UAVDT dataset has three categories of objects (car, truck, bus), and contains 23,829 training images and 16,580 testing images, with 1024 × 540 resolution. The object classification standard of UAVDT also uses the classification standard in MS COCO [10], which is the same as that of Tsinghua–Tencent 100K. According to Figure 1, the proportion of objects in the small interval $area \leq 32 \times 32$ of the UAVDT dataset is 61.5%. Therefore, this dataset is also a good benchmark to evaluate the performance of small object detection.

The indicators used in MS COCO [10], including $AP_{[0.5]}$, $AP_{[0.75]}$, and $AP_{[0.5,0.95]}$, are applied to evaluate the performance of the experimental results on the UAVDT dataset. $AP_{[0.5]}$ and $AP_{[0.75]}$ represent the average accuracy when the IoU ratios of bounding boxes and ground-truth in the detection result are at least 0.5 and 0.75, respectively. As the main indicator used in MS COCO [10], $AP_{[0.5,0.95]}$ represents the average accuracy when the value range of IoU was $[0.5, 0.95]$ and the growth rate was 0.05. The above three indicators were used to evaluate the experimental results of all types of targets in the dataset. The indicator $AP^{small}_{[0.5,0.95]}$ is used to further evaluate the detection performance of small objects (the object size in the interval $area \leq 32 \times 32$) in the UAVDT dataset.

MS COCO. MS COCO [10] is a widely used benchmark dataset for object detection. It consists of 115K train, 5K val and 20K test-dev images in 80 object categories. The division of the instance size of MS COCO is consistent with Tsinghua–Tencent 100K, including small ($area \leq 32 \times 32$ pixels), medium ($32 \times 32 < area \leq 96 \times 96$ pixels), and large ($area > 96 \times 96$ pixels) objects. The proportions of small, medium, and large objects are 41.43%, 34.32%, and 24.24%, respectively.

For the MS COCO dataset, this paper used six evaluation indicators $AP_{[0.5]}$, $AP_{[0.75]}$, $AP_{[0.5,0.95]}$, $AP^{small}_{[0.5,0.95]}$, $AP^{medium}_{[0.5,0.95]}$, and $AP^{large}_{[0.5,0.95]}$ to evaluate the experimental results. The definitions of indicators $AP_{[0.5]}$, $AP_{[0.75]}$, $AP_{[0.5,0.95]}$, and $AP^{small}_{[0.5,0.95]}$ are consistent with the corresponding evaluation indicators used on the UAVDT dataset. $AP^{medium}_{[0.5,0.95]}$ and $AP^{large}_{[0.5,0.95]}$ represent the experimental results in ($32 \times 32 < area \leq 96 \times 96$ pixels) and ($area > 96 \times 96$ pixels), respectively.

4.1.2. Implementation Details

The aspect ratio of people in most of the TinyPerson images varies considerably. Therefore, according to the approach used in [14], the original images were segmented into overlapping sub-images during training and inference. In comparative experiments, the original images in TinyPerson were adjusted to 640 × 640 size for training and testing. Kaiming normal [53] was used to initialize the network. The experiments in all four datasets use the default parameters in [9] for training. The number of training rounds was 300 epochs. The initial learning rate was 0.01. The warm-up strategy [41] was used to adjust the learning rate. A stochastic gradient descent (SGD) with a weight decay of 0.0005 and a momentum of 0.937 was used to train the entire network.

According to the settings used in [15,33,34], the image size was adjusted to 1600 × 1600 for training and testing in Tsinghua–Tencent-100K-related experiments. In UAVDT-related experiments, the original image resolution 1024 × 540 was used for training and testing. Both Tsinghua–Tencent-100K- and UAVDT-related experiments used models that were pre-trained on the MS COCO dataset to initialize the network. In MS-COCO-related experiments, the original image resolution 640 × 640 was used for training and testing.

For both TinyPerson and UAVDT datasets, one GPU was used for training, and the batch size was 32. For the Tsinghua–Tencent 100K dataset, due to the large image resolution, four GPUs were used for training, and the batch size was 20. For the MS COCO dataset, due to the high amount of data, four GPUs were used for training, and the batch size was 64. For all four datasets, only one GPU was used in testing.

4.2. Experiment Preparation

TinyPerson. The proposed method was compared with the state-of-the-art single-stage and two-stage object detection methods. Tables 2 and 3 show the detailed experimental results of the TinyPerson test dataset.

Table 2. Comparisons of *MRs* on TinyPerson test dataset. The best results are marked in bold.

Methods	$MR^{tiny}_{[0.5]}$	$MR^{tiny1}_{[0.5]}$	$MR^{tiny2}_{[0.5]}$	$MR^{tiny3}_{[0.5]}$	$MR^{small}_{[0.5]}$	$MR^{tiny}_{[0.25]}$	$MR^{tiny}_{[0.75]}$
Libra RCNN [5]	89.22	90.93	84.64	81.62	74.86	82.44	98.39
Grid RCNN [7]	87.96	88.31	82.79	79.55	73.16	78.27	98.21
FRCNN-FPN [27]	87.57	87.86	82.02	78.78	72.56	76.59	98.39
FRCNN-FPN-SM [14]	86.22	87.14	79.60	76.14	68.59	74.16	98.28
FRCNN-FPN-SM $S-\delta$ [47]	85.96	86.57	79.14	77.22	69.35	73.92	98.30
FCOS [48]	96.28	99.23	96.56	91.67	84.16	90.34	99.56
SSD512 [4]	93.56	94.55	90.42	85.54	76.79	82.80	99.23
FS-SSD512 [49]	94.01	93.98	91.18	86.01	78.10	83.78	99.35
RetinaNet [13]	92.66	94.52	88.24	86.52	82.84	81.95	99.13
RetinaNet-MSM [14]	88.39	87.80	79.23	79.77	72.18	76.25	98.57
RetinaNet-SM $S-\delta$ [47]	87.00	87.62	79.47	77.39	69.25	74.72	98.41
Scaled-YOLOv4-CSP [8]	86.77	87.36	79.76	76.04	**67.69**	73.03	98.25
YOLOv5s [9]	85.98	87.73	80.09	75.26	68.77	72.32	98.23
SODNet (Proposed)	**83.30**	**82.99**	**76.30**	**72.29**	68.05	**67.52**	**98.04**

Table 3. Comparisons of APs on TinyPerson test dataset. The best results are marked in bold.

Methods	$AP^{tiny}_{[0.5]}$	$AP^{tiny1}_{[0.5]}$	$AP^{tiny2}_{[0.5]}$	$AP^{tiny3}_{[0.5]}$	$AP^{small}_{[0.5]}$	$AP^{tiny}_{[0.25]}$	$AP^{tiny}_{[0.75]}$
Libra RCNN [5]	44.68	27.08	49.27	55.21	62.65	64.77	6.26
Grid RCNN [7]	47.14	30.65	52.21	57.21	62.48	68.89	6.38
FRCNN-FPN [27]	47.35	30.25	51.58	58.95	63.18	68.43	5.83
FRCNN-FPN-SM [14]	51.33	33.91	55.16	62.58	**66.96**	71.55	6.46
FRCNN-FPN-SM $S-\delta$ [47]	51.76	34.58	55.93	62.31	66.81	72.19	6.81
FCOS [48]	17.90	2.88	12.95	31.15	40.54	41.95	1.50
SSD512 [4]	34.00	13.54	35.16	48.73	57.14	61.21	2.52
FS-SSD512 [49]	34.10	14.11	36.17	49.50	56.37	61.58	2.13
RetinaNet [13]	33.53	12.24	38.79	47.38	48.26	61.51	2.28
RetinaNet-MSM [14]	49.59	31.63	56.01	60.78	63.38	71.24	6.16
RetinaNet-SM $S-\delta$ [47]	52.56	33.90	58.00	63.72	65.69	73.09	6.64
Scaled-YOLOv4-CSP [8]	51.25	33.07	56.04	61.94	65.39	73.31	7.04
YOLOv5s [9]	49.61	32.21	52.11	60.95	64.23	71.51	6.63
SODNet (Proposed)	**55.55**	**40.53**	**59.52**	**64.62**	66.22	**75.98**	**7.61**

Although some SOTA detectors (such as Libra RCNN [5], Grid RCNN [7], etc.) performed well on MS COCO [10] or PASCAL VOC [11], they did not achieve good results for small object datasets. A potential reason for this is that the target size in the TinyPerson dataset is too small, which causes the performance of these detectors to considerably decrease. The proposed method uses YOLOv5s [9] as the baseline. Although YOLOv5s achieved good results, the proposed method still improves the core indicators $MR^{tiny}_{[0.5]}$ and $AP^{tiny}_{[0.5]}$ by 2.68% and 5.94%, respectively. Compared with some methods, which are specialized for small object detection, such as FS-SSD512 [49], FRCNN-FPN-SM [14], FRCNN-FPN-SM $S-\delta$ [47], etc., the proposed method performed better than the best one, RetinaNet-SM $S-\delta$ [47], and the corresponding core indicators $MR^{tiny}_{[0.5]}$ and $AP^{tiny}_{[0.5]}$ were improved by 3.7% and 2.99%, respectively. Although the indicators $MR^{small}_{[0.5]}$ and $AP^{small}_{[0.5]}$ of the proposed method were 0.36% and 0.74% lower than the corresponding ones of Scaled-YOLOv4-CSP [8] and FRCNN-FPN-SM [14], respectively, the performance of the proposed method was better than the performance of other methods. Compared with the baseline, the indicators $MR^{small}_{[0.5]}$ and $AP^{small}_{[0.5]}$ of the proposed method were improved by 0.72% and 1.99%, respectively. The results confirm that the proposed method can pay more attention to small objects and improve the recognition ability of small objects. Therefore, the proposed method significantly improves the small object detection performance and achieves a better performance than the state-of-the-art methods.

Tsinghua-Tencent 100K. According to the experimental results shown in Table 4, the proposed method can significantly improve the small object detection performance of the baseline. Table 4 shows the experimental results of the proposed SODNet and other state-of-the-art methods on the Tsinghua–Tencent 100K test dataset in detail. Since the object size of the large interval is greater than 96 × 96 pixels and this paper focuses on evaluating the recognition performance of the methods on small objects, the large interval is not evaluated. The object size range of the overall interval in Table 4 is $area \leq 400 \times 400$, the test results in this interval are used to comprehensively evaluate the detection performance. According to Table 4, the proposed method can achieve a similar performance to that of the state-of-the-art method proposed by Noh et al. [15] and achieve a higher real-time performance. The method proposed by Noh et al. [15] is developed based on the two-stage detector FRCNN [2] as the benchmark model. As the source codes are lacking, we were unable to reproduce Noh's method. In addition, compared with the baseline, the F1 scores obtained by the proposed method improved the corresponding performance on small, medium, and overall classes by 1.3%, 1%, and 0.8%, respectively. The performance in the three classes was improved to varying degrees, but the performance improvement of the small class was greater than the corresponding improvements in the other two classes.

Table 4. Performance comparison with the state-of-the-art models on the Tsinghua–Tencent 100K test dataset. Partial experimental data shown in this table are missing, because some models [15,33,45,50] only provide a part of the related data. The best results are marked in bold.

Methods	Small			Medium			Overall		
	Rec.	*Acc.*	*F1*	*Rec.*	*Acc.*	*F1*	*Rec.*	*Acc.*	*F1*
FRCNN [2] +ResNet101 [41]	80.3	81.6	80.9	94.5	94.8	94.7	89.1	89.7	89.4
Zhu et al. [15]	87.0	82.0	84.4	94.0	91.0	92.5	-	-	-
Perceptual GAN [33]	89.0	84.0	86.4	96.0	91.0	93.4	-	-	-
EFPN [45]	87.1	83.6	85.3	95.2	95.0	95.1	-	-	-
SOS-CNN [50]	-	-	-	-	-	-	93.0	90.0	91.5
Noh et al. [34]	**92.6**	84.9	**88.6**	**97.5**	94.5	96.0	**95.7**	90.6	**93.1**
YOLOv5s [9]	88.7	84.1	86.3	95.6	94.7	95.2	92.9	90.0	91.4
SODNet (Proposed)	90.0	**85.5**	87.6	96.6	**95.8**	**96.2**	94.0	**91.2**	92.6

UAVDT. According to the experimental results shown in Table 5, the proposed method achieved a state-of-the-art performance on the UAVDT [16] dataset. In this dataset, the indicator $AP_{[0.5,0.95]}$ was used to evaluate the experimental results of all the targets in the dataset, and the indicator $AP^{small}_{[0.5,0.95]}$ was used to evaluate the experimental results of the targets within the size interval $area \leq 32 \times 32$. Specifically, the results of the first four rows in Table 5 were calculated by the indicators used in MS COCO [10] for the experimental results provided by Du et al. [16]. According to Table 5, compared with the best performance achieved by ClusDet [52], the proposed method improves the main evaluation indicator $AP_{[0.5,0.95]}$ of MS COCO [10] by 3.4%. For the baseline, the proposed method improves the indicator $AP_{[0.5,0.95]}$ by 4.8%. For the indicator $AP^{small}_{[0.5,0.95]}$, the proposed method in this paper shows improvements of 2.8% and 2.1%, respectively, compared to ClusDet and baseline. This verifies the improvement in the proposed method for small objects. In addition, the indicator means that the detection is correct when the ratio of the bounding boxes' IoU to ground truth is greater than or equal to 0.75. This means that the indicator $AP_{[0.75]}$ has strict requirements for positioning accuracy. The results of 0.5 and 0.75 refer to the overlap ratio of the predicted frame to the actual frame, at 50% and 75%, respectively. A higher value indicates a higher overlap ratio. For the indicator $AP_{[0.75]}$, the proposed method shows improvements of 5.5% and 5.6%, respectively, over ClusDet and baseline. This also verifies the improvement in the proposed method regarding the accuracy of small object positioning. The main reason for this improvement is that the proposed ASPConv and FMF modules optimize the spatial information of small objects.

Table 5. Performance comparison with the baselines and proposed method on the UAVDT test dataset. The best results are marked in bold.

Methods	$AP_{[0.5,0.95]}$	$AP_{[0.5]}$	$AP_{[0.75]}$	$AP^{small}_{[0.5,0.95]}$
R-FCN [1]	7.0	17.5	3.9	4.4
SSD512 [4]	9.3	21.4	6.7	7.1
RON [51]	5.0	15.9	1.7	2.9
FRCNN [2]	5.8	17.4	2.5	3.8
FRCNN-FPN [27]	11.0	23.4	8.4	8.1
ClusDet [52]	13.7	26.5	12.5	9.1
YOLOv5s [9]	12.3	22.4	12.4	9.8
SODNet (Proposed)	**17.1**	**29.9**	**18.0**	**11.9**

MS COCO. The experimental results on MS COCO [10] dataset are shown in Table 6. The baseline of $AP_{[0.5,0.95]}$ and $AP^{small}_{[0.5,0.95]}$ is 35.2% and 18.8%, respectively. The proposed method improves the baseline by 1.2% and 1.3% on $AP_{[0.5,0.95]}$ and $AP^{small}_{[0.5,0.95]}$, respectively. The proposed method also makes some improvements to the indicators $AP^{medium}_{[0.5,0.95]}$ and $AP^{large}_{[0.5,0.95]}$. For $AP_{[0.5]}$ and $AP_{[0.75]}$, the proposed method improves the corresponding baseline by 2.3% and 1.6%, respectively. $AP_{[0.5]}$ and $AP_{[0.75]}$ represent that the detection is correct when the IoU ratio between bounding boxes and ground-truth is greater than or equal to 0.5 and 0.75, respectively. This confirms that the proposed method can effectively improve the spatial information of features, thereby improving the positioning accuracy of objects. According to Table 6, the network Noh et al. [34] focused on small objects and achieved good results on the Tsinghua–Tencent 100K dataset but did not achieve good results on the MS COCO dataset. Compared with Noh et al. [34], the result obtained by SODNet is 3.9% higher than Noh et al. [34] on the small object interval. In addition, according to the experimental results of FRCNN-FPN [27] in Table 6 and the FPS testing results in Table 7, the proposed method achieved a similar detection accuracy to FRCNN-FPN, which is about three times higher than the FPS obtained by FRCNN-FPN. Therefore, the experimental results on the MS COCO dataset also confirm that the proposed method can effectively improve the accuracy of small object detection while ensuring a certain real-time performance.

Table 6. Performance comparison with the proposed method and baseline on MS COCO 2017 *test-dev* dataset. The best results are marked in bold.

Methods	$AP_{[0.5,0.95]}$	$AP_{[0.5]}$	$AP_{[0.75]}$	$AP^{small}_{[0.5,0.95]}$	$AP^{medium}_{[0.5,0.95]}$	$AP^{large}_{[0.5,0.95]}$
R-FCN [1]	29.9	51.9	-	10.8	32.8	45.0
SSD512 [4]	28.8	48.5	30.3	10.9	31.8	43.5
YOLOv3 [3]	33.0	57.9	34.4	18.3	35.4	41.9
FRCNN [2]	34.9	55.7	37.4	15.6	38.7	**50.9**
FRCNN-FPN [27]	36.2	**59.1**	39.0	18.2	39.0	48.2
Noh et al. [34]	34.2	57.2	36.1	16.2	35.7	48.1
YOLOv5s [9]	35.2	53.9	37.8	18.8	39.1	44.0
SODNet (Proposed)	**36.4**	56.2	**39.4**	**20.1**	**40.1**	45.7

4.3. Real-Time Comparison

Not all the comparative methods shown in Tables 2 and 3 have public source codes. Therefore, an efficiency comparison is only performed on the methods with public source codes and the proposed SODNet. Table 7 shows the FPS of each model. The proposed SODNet has the second highest FPS, which is considerably better than the other six comparative models. According to the real-time performance mentioned in [2], FPS need to be greater than or equal to 30. Therefore, the proposed SODNet achieved a high real-time performance.

According to Tables 2, 3 and 7, the proposed method only adds a low computational cost to the baseline, but significantly enhances the original baseline performance. According to Table 7, the proposed method is about four times faster than the compared two-stage detectors, such as [14,27], and about three times faster than the compared single-stage detectors, such as [4,8,49], under the same input size.

Table 7. TinyPerson test dataset was performed on a Nvidia Tesla P100 for testing, where the Batch size and IoU thresholds were set to 1 and 0.5, respectively. The best results are marked in bold.

Methods	Default Input Size	FPS	Uniform Input Size	FPS
FRCNN-FPN [27]	1333 × 800	13	512 × 512	24
FRCNN-FPN-SM [14]	1333 × 800	13	512 × 512	22
SSD512 [4]	512 × 512	33	512 × 512	33
FS-SSD512 [49]	512 × 512	30	512 × 512	30
RetinaNet-MSM [14]	1333 × 800	10	512 × 512	23
Scaled-YOLOv4-CSP [8]	640 × 640	33	512 × 512	39
YOLOv5s [9]	640 × 640	**88**	512 × 512	**96**
SODNet (Proposed)	640 × 640	81	512 × 512	91

4.4. Ablation Study

Since the proportion of small objects in the ($area \leq 32 \times 32$) interval on the TinyPerson and UAVDT datasets reached 85% and 61.5%, respectively, the corresponding ablation experiments of both ASPConv and FMF modules on the TinyPerson and UAVDT test datasets are discussed in this section. As shown in Table 8, the ASPConv module improves the baseline by 0.52% ($MR^{tiny}_{[0.5]}$) and 0.99% ($AP^{tiny}_{[0.5]}$), respectively, on the TinyPerson testing dataset. This confirms that the ASPConv module can improve the spatial expression ability of small object features to a certain extent and alleviate the attenuation of spatial information in the feature transmission process, thereby improving the detection ability of small objects. After adding an FMF module to the baseline and ASPConv module, $MR^{tiny}_{[0.5]}$ and $AP^{tiny}_{[0.5]}$ were further improved by 2.16% and 4.95%, respectively. Compared with the baseline, $MR^{tiny}_{[0.5]}$ and $AP^{tiny}_{[0.5]}$ were improved by 2.68% and 5.94%, respectively. Therefore, the combination of the two modules can make a significant performance improvement. As shown in Figure 2, the FMF module fuses the feature map C1 from the ASPConv module and fuses multi-scale feature maps with rich semantic and spatial information, so the model performance is considerably improved. According to Table 8, $MR^{tiny}_{[0.5]}$ and $AP^{tiny}_{[0.5]}$ can be improved by 1.35% and 4.34%, respectively, by using the FMF module only. Since the FMF module effectively integrates spatial information that is conducive to the detection of small objects, this improvement is reasonable. As shown in Table 8, the corresponding FPS is reduced by 10 after adding the ASPConv module, compared with the baseline. The number of convolutions in ASPConv is significantly higher than that of the focus module [9], so the corresponding computation time increases. After adding the FMF module, the FPS is increased by 4 compared with the baseline. Since SODNet uses the FMF module to replace the original PANet [28] and reduces the bottom-up path enhancement, the FPS is slightly improved. When the ASPConv and FMF modules are added, SODNet only reduces the FPS by 7 compared with the baseline, but the detection accuracy of small objects is considerably improved. In addition, the ASPConv module improves the baseline by 1.1%($AP_{[0.5,0.95]}$) and 0.4%($AP^{small}_{[0.5,0.95]}$), respectively, on the UAVDT dataset. However, the corresponding FPS is reduced by 7. After adding an FMF module to the baseline and ASPConv module, $AP_{[0.5,0.95]}$ and $AP^{small}_{[0.5,0.95]}$ are further improved by 2.8% and 1.0%, respectively. The corresponding FPS is increased by 4. After adding both ASPConv and FMF, SODNet improves the baseline by 17.1%($AP_{[0.5,0.95]}$) and 11.9%($AP^{small}_{[0.5,0.95]}$), respectively. The corresponding FPS is only reduced by 4. In the ablation experiments, although the ASPConv module reduces a certain real-time performance, it can enrich the spatial information in the feature maps of $C_i, i = 2,3,4$ and $P_j, j = 1,2,3,4$. The ASPConv module can effectively improve the detection accuracy of small objects when it works with the FMF module. According to the ablation study, when the two proposed modules work together, they can achieve a better improvement in small object detection than any single module.

Table 8. Ablation study of ASPConv and FMF on TinyPerson and UAVDT testing datasets, where the batch size and IoU threshold are set to 1 and 0.5, respectively. The best results are marked in bold.

Methods	TinyPerson				UAVDT			
	$MR^{tiny}_{[0.5]}$	$AP^{tiny}_{[0.5]}$	Input Size	FPS	$AP_{[0.5,0.95]}$	$AP^{small}_{[0.5,0.95]}$	Input Size	FPS
Baseline YOLOv5s	85.98	49.61	640 × 640	88	12.3	9.8	1024 × 540	49
+ ASPConv	85.46	51.60	640 × 640	78	13.4	10.2	1024 × 540	43
+ FMF	84.63	53.95	640 × 640	**92**	15.1	10.8	1024 × 540	**50**
+ ASPConv + FMF	**83.30**	**55.55**	640 × 640	81	**17.1**	**11.9**	1024 × 540	45

4.5. Qualitative Results

As shown in Figure 6, for TinyPerson, the magnified sub-image in the green frame represents the bounding box predicted by the SODNet. For Tsinghua–Tencent 100K, the magnified sub-image in the red frame represents the ground-truth, and the magnified sub-image in the blue frame represents the object frame predicted by the SODNet. For UAVDT, the magnified sub-image in the green frame represents the bounding box predicted by the SODNet. For MS COCO, comparative experiments were performed on the MS COCO test-dev dataset. Since there is no ground-truth on the MS COCO test-dev dataset, all the rectangular boxes in Figure 6j–l are the bounding boxes predicted by the SODNet. For each pair of images, the images on the left- and right-hand sides are the detection results of the baseline and the proposed SODNet. Figure 6 shows some selected testing results for the TinyPerson, Tsinghua-Tencent 100K, UAVDT, and MS COCO test sets. For each pair of figures, the detection results of the baseline and the proposed method are shown on the left-hand and right-hand sides, respectively. Compared with the baseline, the proposed method can achieve a better detection performance on small and dense objects. In Tsinghua–Tencent 100K, the proposed method still detected some existing but unmarked objects, which can be regarded as reasonable examples of false positives.

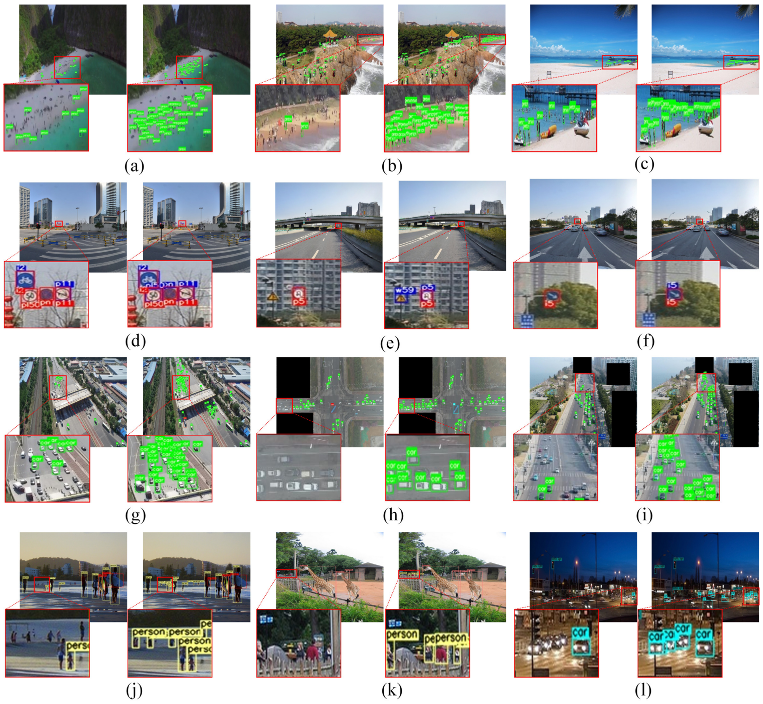


Figure 6. Sample testing results of TinyPerson (shown in subfigures (**a**–**c**)), Tsinghua-Tencent 100K (shown in subfigures (**d**–**f**)), UAVDT (shown in subfigures (**g**–**i**)) and MS COCO (shown in subfigures (**j**–**l**)) test datasets.

5. Conclusions

The proposed method is applied to the single-stage detector YOLOv5s to solve the issues of small object detection. First, an adaptive spatial parallel convolution module (ASPConv) is proposed to extract the multi-scale local context information of small objects and enhance the spatial information of small objects. Second, a fast multi-scale fusion module (FMF) is designed, which effectively integrates the high-resolution feature maps with rich spatial information output from the APSConv module. The low-resolution feature maps with rich semantic information can be efficiently mapped to high-resolution space. Multi-scale feature map fusion is performed to generate high-resolution feature maps with rich spatial and semantic information that are conducive to small object detection. In addition, according to the ablation study results shown in Table 8, the two modules can effectively be integrated to achieve fast and accurate detection. The experimental results of the TinyPerson, Tsinghua–Tencent 100K, UAVDT, and MS COCO benchmark datasets confirm that the proposed method efficiently and significantly improves the detection performance of small objects, and the corresponding results are highly competitive. In TinyPerson-related experiments, compared with the most advanced methods in the literature, the proposed method improves $AP_{[0.5]}^{tiny}$ by 5.94%, and achieves a 91 FPS on a single Nvidia Tesla P100. Therefore, the proposed SODNet can effectively enhance the detection performance of small objects and realize real-time performance. Therefore, the proposed method can be transferred to many small object detection scenes, such as UAV search-and-rescue and intelligent driving. In future research, the optimization and applications of the proposed method will be further explored in more fields.

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

Funding: This work is supported in part by Sichuan Science and Technology Program under Grant 2021YFG0315.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *arXiv* **2016**, arXiv:1605.06409.
2. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497.
3. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
4. 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.
5. 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 (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 821–830.
6. 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 (CVPR), Virtual, 14-19 June 2020; pp. 10781–10790.
7. Lu, X.; Li, B.; Yue, Y.; Li, Q.; Yan, J. Grid r-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7363–7372.
8. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. Scaled-yolov4: Scaling cross stage partial network. *arXiv* **2011**, arXiv:2011.08036.
9. Jocher, G.; Nishimura, K.; Mineeva, T. Yolov5. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 September 2021).

10. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
11. Everingham, M.; Gool, L.V.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
12. Qi, G.; Wang, H.; Haner, M.; Weng, C.; Chen, S.; Zhu, Z. Convolutional neural network based detection and judgement of environmental obstacle in vehicle operation. *CAAI Trans. Intell. Technol.* **2019**, *4*, 80–91. [CrossRef]
13. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 2980–2988.
14. Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; Han, Z. Scale match for tiny person detection. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; pp. 1246–1254.
15. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-sign detection and classification in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2110–2118.
16. 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.
17. Shuang, K.; Lyu, Z.; Loo, J.; Zhang, W. Scale-balanced loss for object detection. *Pattern Recognit.* **2021**, *117*, 107997. [CrossRef]
18. Ma, W.; Wu, Y.; Cen, F.; Wang, G. Mdfn: Multi-scale deep feature learning network for object detection. *Pattern Recognit.* **2020**, *100*, 107149. [CrossRef]
19. Bosquet, B.; Mucientes, M.; Brea, V.M. Stdnet-st: Spatio-temporal convnet for small object detection. *Pattern Recognit.* **2021**, *116*, 107929. [CrossRef]
20. 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]
21. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *arXiv* **2019**, arXiv:1905.05055.
22. Qi, G.-J. Hierarchically gated deep networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2267–2275.
23. Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 385–400.
24. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
25. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284
26. Zhu, Z.; Luo, Y.; Wei, H.; Li, Y.; Qi, G.; Mazur, N.; Li, Y.; Li, P. Atmospheric Light Estimation Based Remote Sensing Image Dehazing. *Remote Sens.* **2021**, *13*, 2432. [CrossRef]
27. 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 (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 2117–2125.
28. 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 (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
29. Zheng, M.; Qi, G.; Zhu, Z.; Li, Y.; Wei, H.; Liu, Y. Image Dehazing by an Artificial Image Fusion Method Based on Adaptive Structure Decomposition. *IEEE Sens. J.* **2020**, *20*, 8062–8072. [CrossRef]
30. Zhu, Z.; Luo, Y.; Qi, G.; Meng, J.; Li, Y.; Mazur, N. Remote Sensing Image Defogging Networks Based on Dual Self-Attention Boost Residual Octave Convolution. *Remote Sens.* **2021**, *13*, 3104. [CrossRef]
31. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316.
32. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 510–519.
33. 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 (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 1222–1230.
34. 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 IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9725–9734.
35. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
36. Zhang, Z.; Wang, Z.; Lin, Z.; Qi, H. Image super-resolution by neural texture transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7982–7991.
37. Zhu, Z.; Wei, H.; Hu, G.; Li, Y.; Qi, G.; Mazur, N. A Novel Fast Single Image Dehazing Algorithm Based on Artificial Multiexposure Image Fusion. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–23. [CrossRef]

38. Islam, M.A.; Rochan, M.; Bruce, N.D.; Wang, Y. Gated feedback refinement network for dense image labeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 3751–3759.
39. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 448–456.
40. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. *arXiv* **2017**, arXiv:1701.04128.
43. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
44. Huang, Z.; Wang, J.; Fu, X.; Yu, T.; Guo, Y.; Wang, R. Dc-spp-yolo: Dense connection and spatial pyramid pooling based yolo for object detection. *Inf. Sci.* **2020**, *522*, 241–258. [CrossRef]
45. Deng, C.; Wang, M.; Liu, L.; Liu, Y. Extended feature pyramid network for small object detection. *arXiv* **2003**, arXiv:2003.07021.
46. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
47. Gong, Y.; Yu, X.; Ding, Y.; Peng, X.; Zhao, J.; Han, Z. Effective fusion factor in fpn for tiny object detection. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Virtual, 5–9 January 2021; pp. 1160–1168.
48. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 9627–9636.
49. Liang, X.; Zhang, J.; Zhuo, L.; Li, Y.; Tian, Q. Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 1758–1770. [CrossRef]
50. Meng, Z.; Fan, X.; Chen, X.; Chen, M.; Tong, Y. Detecting small signs from large images. In Proceedings of the 18th IEEE International Conference on Information Reuse and Integration (IRI), San Diego, CA, USA, 4–6 August 2017; pp. 217–224.
51. Kong, T.; Sun, F.; Yao, A.; Liu, H.; Lu, M.; Chen, Y. Ron: Reverse connection with objectness prior networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017; pp. 5936–5944.
52. Yang, F.; Fan, H.; Chu, P.; Blasch, E.; Ling, H. Clustered object detection in aerial images. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 8311–8320.
53. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 1026–1034.