

Article

Remote Sensing Small Object Detection Network Based on Multi-Scale Feature Extraction and Information Fusion

Junsuo Qu ^{1,*} , Tong Liu ^{1,2} , Zongbing Tang ¹ , Yifei Duan ¹ , Heng Yao ¹  and Jiyuan Hu ¹ 

¹ Xi'an Key Laboratory of Advanced Control and Intelligent Process, School of Automation, Xi'an Robotic Intelligent Systems International Science and Technology Cooperation Base, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; 794624659@stu.xupt.edu.cn (T.L.); t2389360453@stu.xupt.edu.cn (Z.T.); duanyifei@stu.xupt.edu.cn (Y.D.); yao.xupt.edu.cn@stu.xupt.edu.cn (H.Y.); hujiyuan328@stu.xupt.edu.cn (J.H.)

² School of Automation, Xi'an University of Posts & Telecommunications, Xi'an 710121, China

* Correspondence: qujunsuo@xupt.edu.cn

Abstract: Nowadays, object detection algorithms are widely used in various scenarios. However, there are further small object detection requirements in some special scenarios. Due to the problems related to small objects, such as their less available features, unbalanced samples, higher positioning accuracy requirements, and fewer data sets, a small object detection algorithm is more complex than a general object detection algorithm. The detection effect of the model for small objects is not ideal. Therefore, this paper takes YOLOXs as the benchmark network and enhances the feature information on small objects by improving the network's structure so as to improve the detection effect of the model for small objects. This specific research is presented as follows: Aiming at the problem of a neck network based on an FPN and its variants being prone to information loss in the feature fusion of non-adjacent layers, this paper proposes a feature fusion and distribution module, which replaces the information transmission path, from deep to shallow, in the neck network of YOLOXs. This method first fuses and extracts the feature layers used by the backbone network for prediction to obtain global feature information containing multiple-size objects. Then, the global feature information is distributed to each prediction branch to ensure that the high-level semantic and fine-grained information are more efficiently integrated so as to help the model effectively learn the discriminative information on small objects and classify them correctly. Finally, after testing on the VisDrone2021 dataset, which corresponds to a standard image size of 1080p (1920 × 1080), the resolution of each image is high and the video frame rate contained in the dataset is usually 30 frames/second (fps), with a high resolution in time, it can be used to detect objects of various sizes and for dynamic object detection tasks. And when we integrated the module into a YOLOXs network (named the FE-YOLO network) with the three improvement points of the feature layer, channel number, and maximum pool, the mAP and APs were increased by 1.0% and 0.8%, respectively. Compared with YOLOV5m, YOLOV7-Tiny, FCOS, and other advanced models, it can obtain the best performance.

Keywords: small object detection; feature stitching; residual convolution; remote sensing



Academic Editor: Pedro Melo-Pinto

Received: 6 January 2025

Revised: 17 February 2025

Accepted: 20 February 2025

Published: 5 March 2025

Citation: Qu, J.; Liu, T.; Tang, Z.; Duan, Y.; Yao, H.; Hu, J. Remote Sensing Small Object Detection Network Based on Multi-Scale Feature Extraction and Information Fusion. *Remote Sens.* **2025**, *17*, 913.

<https://doi.org/10.3390/rs17050913>

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the Creative Commons

Attribution (CC BY) license

(<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, with the advancement of science and technology, images taken by drones are widely used in remote sensing, autonomous driving, monitoring systems, intelligent transportation, and other fields [1–4]. Generally, people determine a specific object in an

image as a small object by setting a threshold that the object is smaller than this threshold in comparison with the entire image. However, in most cases, since the clarity and size of the images are not entirely consistent, objects with a height or width of less than 32×32 pixels will also be classified as small objects. In the field of remote sensing, due to the complexity of the shooting heights and geographical environments, more small objects are smaller than 20×20 pixels. The excessively small size of these objects makes them prone to being lost during the network downsampling process, thereby weakening the ability to characterize small objects. In complex environments, they are easily confused with background information, which is not beneficial for model detection. Although many scholars have proposed a series of excellent target detection algorithms, such as MAE-Det [5] and the Co-DETR [6] series, with the rapid development of convolutional neural networks and Transformer [7] technology, the accuracy of these algorithms can no longer meet people's needs. In 2012, the emergence of the classical convolutional neural network AlexNet significantly reduced the error rate in the ImageNet large-scale visual recognition competition by about 10% and surpassed that of previous detection methods. This breakthrough demonstrated the great potential of convolutional neural networks in image tasks and promoted the innovation of object detection technology. In 2015, YOLOV1 [8], proposed by Joseph Redmon, marked the beginning of the YOLO series. To further improve the accuracy in single-stage object detection, the academic community has proposed a variety of anchor-based detection algorithms (such as YOLOV4 [9], YOLOV5, YOLOV7 [10], etc.) and anchor-free object detection algorithms (such as FCOS [11], YOLOX [12], YOLOV8, etc.). In this paper, YOLOXs is used as the benchmark network to enhance the feature information of small objects by improving the network's structure so as to improve the detection effect of the model for small objects. In a small object detection algorithm, shallow features are generally used to help the model to accurately locate the position of an object using rich fine-grained information [13], while deep features help the model accurately classify the object through rich semantic information. Combining these two features can improve the model's ability to accurately locate and classify objects at different scales. Therefore, in the field of object detection, researchers have proposed a neck network structure. This structure is mainly responsible for effective information fusion of the different-scale feature layers output by the backbone network, helping the model to understand the internal relations and laws of the data better and transmitting the processed effective feature layer to the detection head network.

Although the features extracted directly from the backbone network can also position and classify an object, they cannot effectively characterise objects of various scales, resulting in problems such as a low detection accuracy and false and missed detections. Many scholars have proposed series of excellent information fusion structures in the neck network, such as the FPN [14], PANet, and Bi-FPN [15]. At present, the neck network usually comprises an FPN (feature pyramid network) or variants of an FPN, as in the feature fusion modules bi-directional feature pyramid network (Bi-FPN), adaptively spatial feature fusion [16] (ASFF), and neural architecture search feature pyramid networks [17] (NAS-FPNs). These networks fuse feature maps of different scales through up-sampling [18] and down-sampling operations [19]. However, this type of fusion method can easily lead to losses in the information transmission between non-adjacent feature layers. Although the features of adjacent layers can be effectively fused through up-sampling or down-sampling, non-adjacent feature layers can only be fused indirectly through the middle layer. Adjacent layers will discard some unnecessary information when fusing the information, which may guide the characteristics of subsequent layers, thus affecting the detection performance of the model.

In order to improve this problem, this paper proposes a feature fusion and distribution network (FFDN) module. This module replaces the information transmission path from deep features to shallow features in PANet [20] (path aggregation network). The FFDN module first fuses and extracts the different-scale feature layers output by the backbone network to generate global feature information containing multi-scale objects. Then, this global feature information is distributed to each prediction branch to ensure that the semantic information on small objects is retained so as to guide the model to correctly classify shallow, small objects. The experimental results show that integrating the FFDN module into the improved model can significantly improve the detection accuracy of the model for small objects compared with that of other classical models.

The main contributions of this paper can be summarized as follows:

1. We designed a feature fusion and distribution network module, which aims to fuse the four effective feature layers output by the backbone network to obtain high-resolution features with rich fine-grained information. Then, these high-resolution features are passed to the corresponding feature layer through the distribution mechanism to realize the small object prediction function.
2. We optimized the information transmission path of PANet, replaced the path from deep to shallow, and retained the transmission path from shallow to deep. This adjustment improves the accuracy of small object detection without significantly increasing the model parameters.
3. We chose FE-YOLO as the benchmark network and integrated the FFDN module without deconvolution operation into FE-YOLO for feature splicing and feature addition in order to find the most effective feature distribution method in the FFDN module.
4. We used a 1×1 convolution operation to reduce the number of channels of the fused features, thereby eliminating useless feature information and enhancing the nonlinear ability of the model. Then, the information is effectively extracted from the spatial dimension through the residual convolution operation, and the feature information that is useful for small object detection is selected.

The rest of this paper is organized as follows: Section 2 summarizes the related research methods of small object detection; Section 3 describes the improved module and the reasons for its improvement in detail, including the addition of an information fusion module, feature extraction module, and information distribution module. In Section 4, we describe the relevant steps involved in this experiment, analyze the result and discuss the current situation and the shortcomings of the model; Section 5 summarizes the overall content of the work.

2. Related Work

In object detection tasks, the shallow feature layer predominantly retains fine-grained spatial details, assisting the model in achieving precise object localization, whereas the deep feature layer encapsulates comprehensive semantic representations that facilitate accurate object classification [21]. The primary objective of the neck network lies in strategically fusing these heterogeneous feature hierarchies to enhance the model's detection accuracy. The efficacy of the neck network hinges on its feature integration methodology, which amplifies overall performance through systematic aggregation of distinct contextual information [22].

In deep learning, there are two primary methods of feature fusion: feature splicing [23] and feature addition [24]. Feature splicing increases the channel count of the output feature layer. For example, if two input feature layers share identical spatial dimensions (e.g., height and width) but have channel counts of p and q , respectively, the spliced output feature layer will have $p + q$ channels. Feature addition maintains the original channel count of the output feature layer. However, this method requires the input feature layers to

have identical channel numbers and spatial dimensions; otherwise, the fusion will fail [25] (as shown in Figure 1).

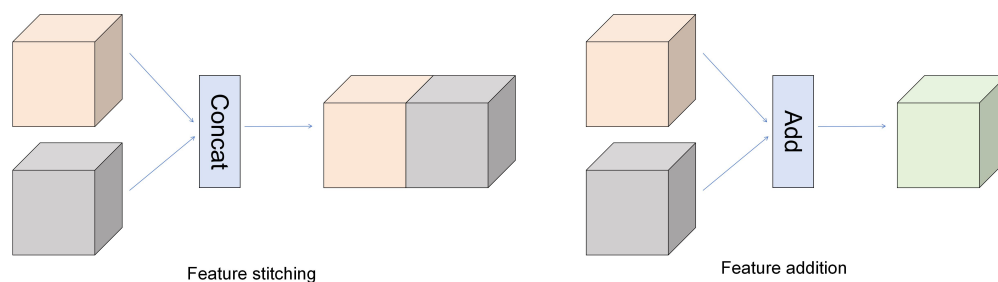


Figure 1. Two common feature fusion diagrams.

In feature fusion, feature splicing retains all information from each feature layer, fully utilizes the expressive power of individual features, and enhances the network’s modeling capacity by increasing feature dimensions. However, this approach inevitably introduces additional computational parameters. In contrast, feature addition preserves the semantic integrity of original features without introducing redundancy. It strengthens critical features while suppressing noise, thereby improving network stability and generalization. Although both methods theoretically improve model detection accuracy, this paper will determine the optimal fusion strategy based on experimental validation [26].

While feature fusion enables the integration of deep and shallow features to augment the model’s representational fidelity toward object characteristics, this procedure intrinsically necessitates spatially uniform dimensionality across fused feature layers. Resolution incompatibility mitigation: When spatial discrepancies (e.g., height or width) arise between feature hierarchies, lower-level layers generally necessitate resampling through upsampling operations. Implemented upsampling methodologies encompass interpolation-based techniques [27] and transposed convolution. Crucially, interpolation-based schemes are predicated on a priori heuristic priors for parameter initialization, which inhibits the convolutional neural networks’ [28] capacity for context-aware feature adaptation.

Deconvolution [29] (or transposed convolution) allows the network to adaptively upsample through gradient backpropagation, which is more suitable for dynamic learning. Therefore, for this paper, we chose deconvolution. Deconvolution can convert a smaller feature map into a larger one, providing more detailed information and helping to detect small objects [30]. Although transposed convolution is not the inverse operation of convolution, it can restore the feature map size before downsampling, but the pixel values of the restored feature map and the original feature map are not exactly the same. Figure 2 illustrates the region mapping relationship between convolution and deconvolution. The convolution operation is a many-to-one mapping, while deconvolution is a one-to-many mapping that enlarges the size of the feature map. Its calculation formulas are shown in Formulas (1) and (2).

$$\mathcal{H}_{out} = (\mathcal{H}_{in} - 1) \times s[0] - 2 \times p[0] + k[0] \quad (1)$$

$$\mathcal{W}_{out} = (\mathcal{W}_{in} - 1) \times s[0] - 2 \times p[0] + k[0] \quad (2)$$

Among them, s represents the pixel distance in which the convolution kernel moves in the specified direction each time, p represents the pixel filling, and k represents the size of the convolution kernel.

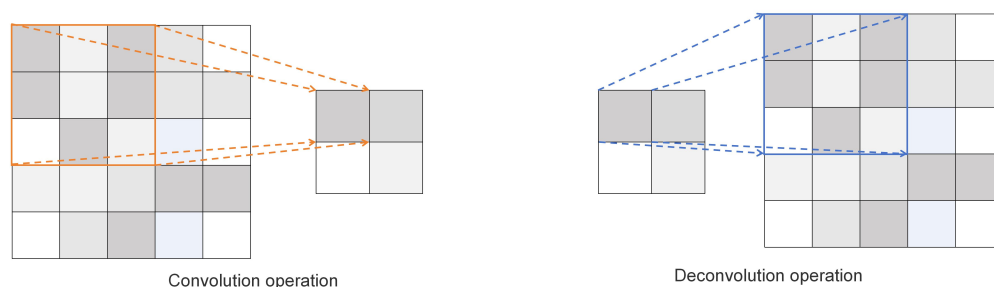


Figure 2. Convolution and deconvolution diagram.

3. Our Work

YOLOX employs PANet as its neck network, which introduces a transmission path from shallow to deep layers based on the traditional FPN (feature pyramid network). PANet effectively integrates features from different levels through this approach. However, Wang et al. [31] demonstrated that this fusion method is more effective for adjacent feature layers. For non-adjacent layers, information fusion can only occur indirectly [32]. As illustrated in Figure 3, PANet's information fusion process involves two distinct scenarios:

When the level-4 and level-3 feature layers are fused, the two can directly perform information fusion through feature splicing or feature addition.

When the level-4 and level-2 feature layers are fused, the level-4 and level-3 feature layers need to be fused first, and then fused with the level-2 feature layer.

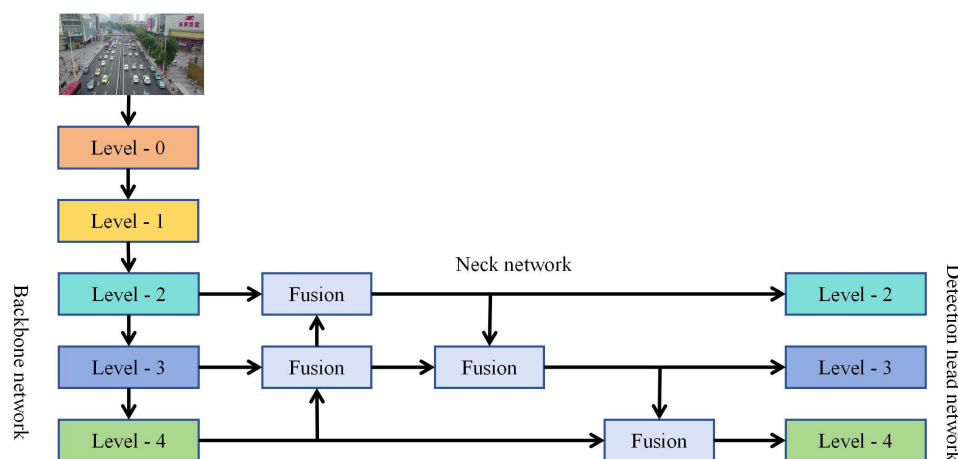


Figure 3. PANet information fusion diagram.

When performing feature fusion across adjacent network hierarchies, certain information with latent representational value might be unintentionally neglected, despite its capacity for informing subsequent hierarchical computations. This empirical observation indicates that PANet architectures built upon FPN foundations could establish intrinsic barriers to globally optimal feature assimilation. More concretely, suboptimal diffusion of high-level semantic features to govern lower-level feature encoding may manifest, thereby degrading detection efficacy for small-scale objects via inadequate semantic regularization.

To address this problem, our paper proposes a new network structure called a feature fusion and distribution mechanism network (FFDN). The FFDN module obtains high-resolution features containing rich fine-grained information by fusing four effective feature layers (B1, B2, B3, and B4) generated by the backbone network. These fused high-resolution features are then distributed to corresponding prediction layers through a specialized allocation mechanism. The four feature layers B1, B2, B3, and B4 are extracted at different network depths, each carrying distinct information characteristics: B1 (shallowest layer):

Contains rich fine-grained spatial details suitable for precise small object localization, but exhibits weak semantic information; B2: Maintains moderate fine-grained information while developing preliminary semantic features that aid small object classification; B3: Represents deeper features with enhanced semantic information that improves classification accuracy, though at reduced spatial resolution. B4 (deepest layer): Possesses the most comprehensive semantic information for robust global feature extraction and category discrimination, but lacks detailed spatial information. By effectively fusing and strategically distributing these hierarchical features, FFDN leverages complementary information across different network depths to enhance small object detection performance. To optimize accuracy without significantly increasing model complexity, we modify PANet's information pathways by replacing the deep-to-shallow propagation path while preserving the original shallow-to-deep transmission route, as illustrated in Figure 4.

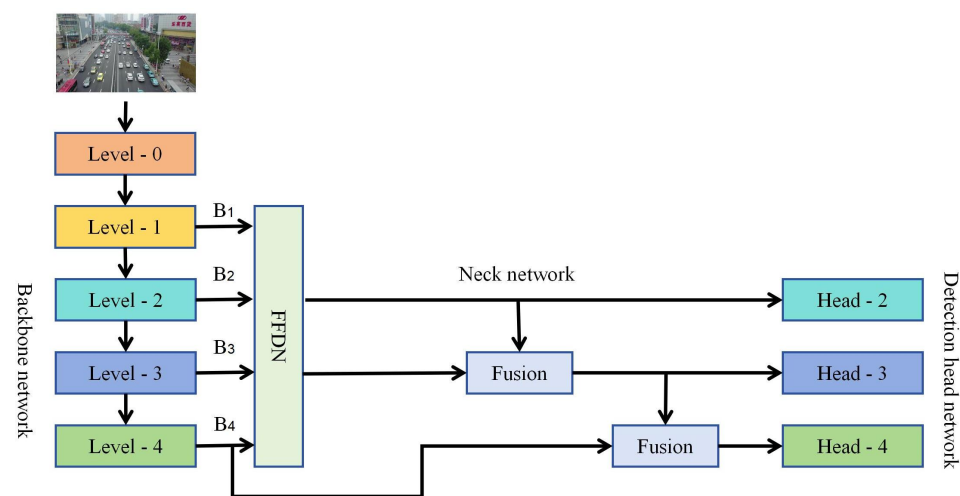


Figure 4. The schematic diagram of the improved neck network.

Figure 5 illustrates the FFDN module's structure, which comprises three core components: the information fusion module, the feature extraction module, and the information distribution module. The workflow proceeds as follows: Firstly, the feature layers of different levels are fused through the information fusion module to obtain features containing global information. Then, the convolutional neural network is used to extract these features. Finally, information filtering is realized by convolution operation to accurately distribute information.

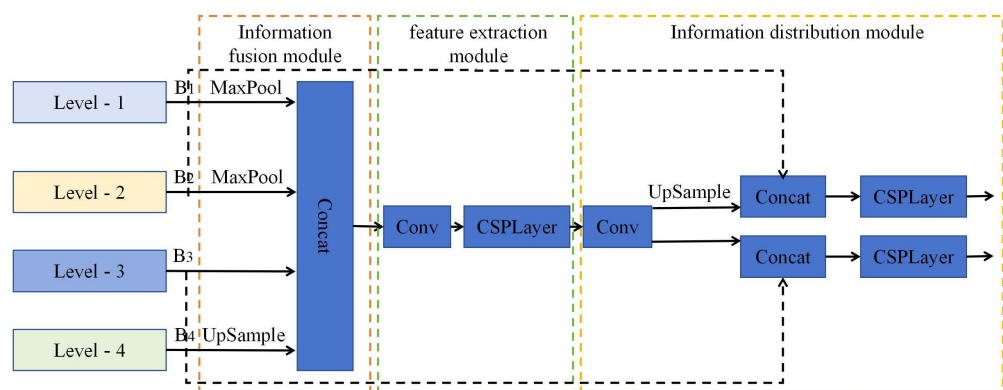


Figure 5. Schematic diagram of FFDN module.

3.1. Information Fusion Module

To make full use of the detailed information of shallow features, this paper selects four effective feature layers of B1, B2, B3 and B4, output by the backbone network for fusion to obtain higher resolution features that retain rich small object information. To effectively fuse the feature layers of different scales, the methods of maximum pooling and deconvolution are adopted, respectively. When selecting the size of the feature layer after information fusion, this paper considers the following two points:

This paper's goal is to improve the detection accuracy of small objects, so it is necessary to maintain a large feature map to retain more fine-grained information.

The larger the scale of the feature map is, the greater the amount of calculation. Therefore, it is necessary to consider the subsequent computational complexity to maintain a smaller feature size.

Therefore, the B3 feature size is selected as the object size after fusion, which not only efficiently aggregates the feature information of different scales but also reduces the computational complexity. The implementation formula is shown in Formula (3).

$$F_1 = \text{Concat}[\text{MaxPool}(B_1), \text{MaxPool}(B_2), B_3, \text{UpSample}(B_4)] \quad (3)$$

Among them, concat represents the feature splicing operation, and F1 represents the feature layer after feature fusion.

3.2. Feature Extraction Module

To enhance the acquisition of detailed small-object features, our approach employs 1×1 convolutional layers to reduce channel dimensionality after feature fusion. This operation eliminates redundant features while strengthening the model's nonlinear modeling capability. Subsequently, spatially aware residual convolution blocks extract and selectively preserve small-object-relevant features through structured spatial operations. The specific operation can be seen in Formula (4).

$$F_2 = \text{CSPLayer}(\text{Conv}(F_1)) \quad (4)$$

Among them, CSPLayer represents the residual convolutional layer in the YOLOX network.

3.3. Information Distribution Module

To effectively allocate global information to different feature layers, this paper adopted the method of feature fusion based on the experimental results of the large kernel residual attention mechanism. The effective feature layer of the backbone network is denoted as B. The specific steps include: First, integrate global information through two different 1×1 convolution operations, adjusting the number of channels to be consistent with the feature branches of the backbone network in order to reduce the amount of computation, as shown in Formula (5). Then, the feature layer is extended to the same size as feature layer B by upsampling for stitching. The upsampled global feature layer is then fused with feature layer B to enhance the small object feature information and assist the model in correctly classifying the object, as shown in Formulas (6) and (7). Finally, in order to improve the learning ability of small objects and reduce computational complexity, this paper uses a residual convolution layer to reduce the number of channels in the feature layer after fusion and extract effective feature information.

$$F_3, F_4 = \text{Conv}_1(F_2), \text{Conv}_2(F_2) \quad (5)$$

$$F_{03} = \text{CSPLayer}(\text{Cat}[B_2, \text{UpSample}(F_3)]) \quad (6)$$

$$F_{04} = \text{CSPLayer}(\text{Cat}[B_3, F_4]) \quad (7)$$

B2 represents the level-2 feature layer illustrated in Figure 5, and B3 represents the level-3 feature layer presented in Figure 5.

4. Experiments & Discussion

This research work is based on the PyTorch-1.7.0 framework and uses GPU for training. The specific configuration of the experimental environment is shown in Table 1.

Table 1. Configuration of experimental environment.

Project	Environment
Operating system	Centos7 system
GPU (VRAM and RAM)	NVIDIA GeForce GTX 4090Ti (24G, 128G)
PyTorch version	1.7.0
CUDA	11.0

The learning rate is configured as 0.001, the weight decay parameter is optimized to 0.0005, and the parameters are optimized via the stochastic gradient descent algorithm integrated with a momentum optimizer. The input image dimensions are standardized to 640×640 , the batch size is fixed at 16, the total training iteration count is capped at 200 epochs, and the momentum factor is calibrated to 0.937. To maximize the model's potential, neither mosaic nor mixup data augmentation strategies are employed during the training phase. As the backbone network of the object detection model is structurally modified in this study, the original pre-trained weights from the YOLOX model are excluded from the experiments.

In this experiment, we selected YOLOX, incorporating three improvements—namely an increase in feature layers, an expansion of channel numbers, and the addition of max-pooling layers—as the benchmark network, designated as FE-YOLO. For the comparative experiments, minor parametric discrepancies exist among models owing to variations in model size and original network architectures.

4.1. Optimal Feature Fusion Mode

To study the most effective feature distribution method in the FFDN module, this paper used FE-YOLO as the benchmark network and integrated the FFDN module without deconvolution operation into FE-YOLO to perform comparative experiments of feature stitching and feature addition. The results are shown in Table 2. Both methods have a positive impact on the benchmark model, indicating that they can effectively retain the feature information of small objects and thus efficiently complete the feature distribution task. The feature addition method makes the benchmark model reach 15.2% and 7.7% on the indexes of mAP and APs, respectively. The feature splicing method makes these two indicators increase to 15.5% and 7.9%, respectively, demonstrating the advantages of feature splicing in improving the overall detection performance of the model and the detection accuracy of small objects. This is because feature splicing can accommodate more feature information and effectively preserve the features of small objects. Therefore, considering the model's performance, this paper chose feature splicing as the feature distribution method in the FFDN module.

Table 2. Comparison results of different feature fusion methods.

<i>Concat</i>	<i>Add</i>	<i>AR_{0.5}</i>	<i>AP_{0.5}</i>	<i>AP_{0.75}</i>	<i>mAP</i>	<i>AP_s</i>	<i>AP_m</i>	<i>AP_l</i>	<i>Params</i>
		0.277	0.272	0.148	0.148	0.074	0.224	0.309	9.028M
✓		0.283	0.281	0.157	0.155	0.079	0.238	0.284	13.008M
	✓	0.280	0.277	0.152	0.152	0.077	0.233	0.288	12.884M

4.2. Optimal Downsampling Mode

Based on the experiment described in Section 4.1, we investigated whether the downsampling operation affects the feature information of small objects, thus impacting the detection performance of the model. To this end, this paper continued to use FE-YOLO as the benchmark network and selected the model that performs best, described in Section 4.1 for comparative experiments. We replace the maximum pooling operation in the model with the average pooling operation, and train, validate, and test under the same conditions. The results are shown in Table 3. From the results in the last row, it can be seen that the overall detection performance of the network using the average pooling operation is only slightly higher than that of the benchmark network, indicating that average pooling fails to effectively highlight the object feature information or remove interference, resulting in limited performance improvement of the model. In contrast, the maximum pooling operation can effectively retain the fine-grained information of features and suppress background interference, thus effectively transmitting the feature information of small objects to the subsequent network structure and significantly improving the detection accuracy of the model.

Table 3. Comparison results of different pooling methods.

<i>Maxpool</i>	<i>Avepool</i>	<i>AR_{0.5}</i>	<i>AP_{0.5}</i>	<i>AP_{0.75}</i>	<i>mAP</i>	<i>AP_s</i>	<i>AP_m</i>	<i>AP_l</i>	<i>Params</i>
✓	✓	0.277	0.272	0.148	0.148	0.074	0.224	0.309	9.028M
		0.283	0.281	0.157	0.155	0.079	0.238	0.284	13.008M
		0.280	0.273	0.148	0.149	0.076	0.228	0.294	13.008M

4.3. Optimal Up-Sampling Mode

To study the influence of the deconvolution operation on the FFDN module, we integrated the FFDN module into FE-YOLO and conducted a comparative experiment with the nearest neighbor interpolation method. The experimental results are shown in Table 4. Among them, FFDN represents the feature fusion and distribution network with the deconvolution operation, while FFDN* represents it without the deconvolution operation.

Table 4. Comparison results of different up-sampling methods.

<i>Index</i>	<i>AR_{0.5}</i>	<i>AP_{0.5}</i>	<i>AP_{0.75}</i>	<i>mAP</i>	<i>AP_s</i>	<i>AP_m</i>	<i>AP_l</i>	<i>Params</i>
FE-YOLO	0.277	0.272	0.148	0.148	0.074	0.224	0.309	9.028M
FFDN*-YOLO	0.283	0.281	0.157	0.155	0.079	0.238	0.284	13.008M
FFDN-YOLO	0.304	0.281	0.161	0.158	0.082	0.236	0.296	15.515M

It can be observed that the mAP and APs of FFDN*-YOLO reached 15.5% and 7.9%, respectively, which were 0.7% and 0.5% higher than those of the benchmark model FE-YOLO. This shows that the FFDN* module enhances the model's ability to detect small objects and improves the comprehensiveness of object detection.

The mAP and APs of FFDN-YOLO reached 15.8% and 8.2%, respectively, which were 1% and 0.8% higher than those of FE-YOLO. Although there is no improvement in the AP0.5 and APm indicators, FFDN-YOLO has significantly improved indicators such as AR0.5, AP0.75, APs, and APl, especially with AR0.5 increasing by 2.1%. This shows that the introduction of the deconvolution operation can enhance the detailed information of features by improving the resolution of feature maps, thereby helping the model better identify small objects. Therefore, this paper chose deconvolution as the upsampling method for FFDN.

To empirically demonstrate model performance enhancement, this study conducted systematic validation set analysis during training iterations. Figure 6 presents a compara-

tive assessment of mAP0.5 between baseline FE-YOLO and improved FFDN-YOLO across 200 training epochs. Quantitative results reveal that FFDN-YOLO achieves a marginally accelerated convergence rate compared with FE-YOLO, ultimately attaining stabilized metric values approaching 0.3 mAP, representing statistically significant evidence of its superior capability in rapid small-object feature acquisition and enhanced validation accuracy, validating the architectural modifications' efficacy in discriminative pattern learning.

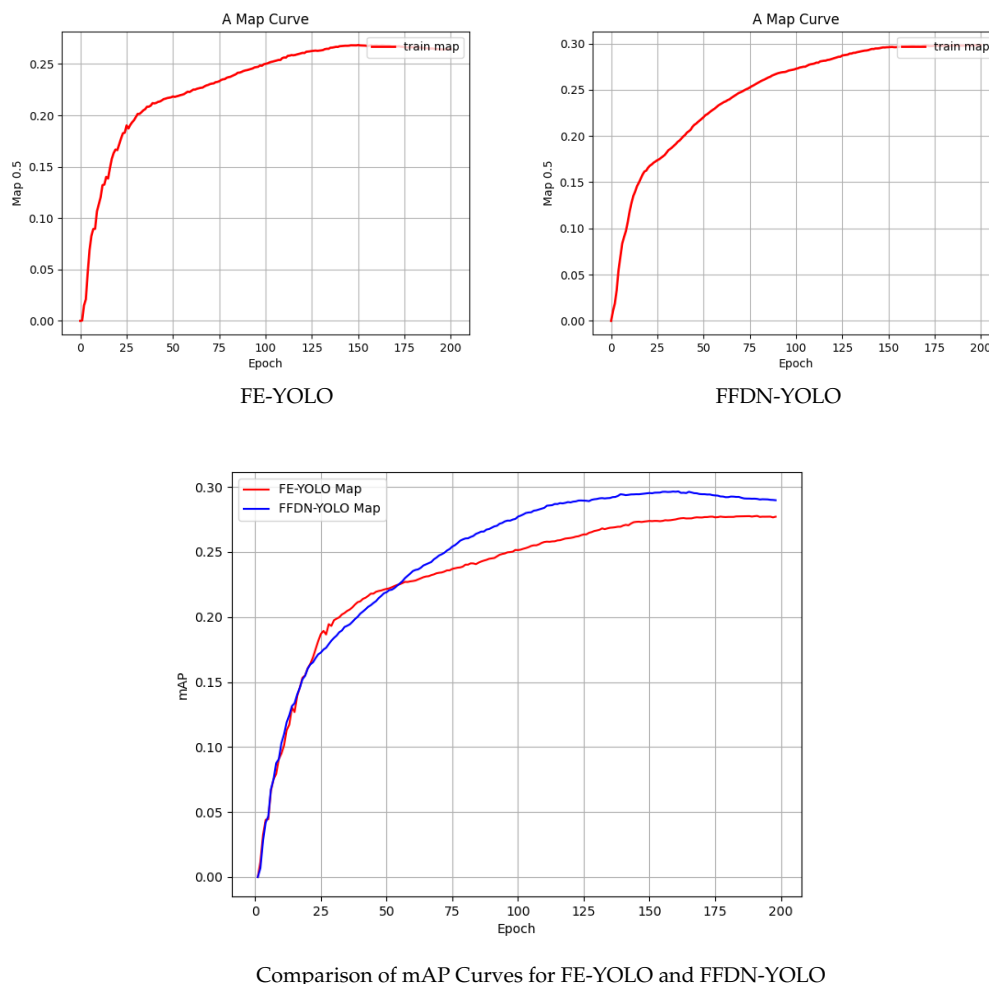


Figure 6. Map of mAP0.5 during the training of Fe-YOLO and FFDN-YOLO.

To demonstrate the detection performance of FFDN-YOLO, we selected an exemplary image containing multiple objects of varying scales from the dataset. Under identical evaluation conditions, both FE-YOLO and FFDN-YOLO were comparatively analyzed (see Figure 7), with the first row displaying the input image alongside ground-truth annotations and the second row presenting detection outputs from both models. Through quantitative comparison, FFDN-YOLO demonstrates superior small-object detection capabilities evident in identifying distant vehicular instances, while simultaneously achieving more precise foreground classification, representing empirical validation that substituting PANet's deep-to-shallow pathways with FFDN modules optimizes semantic guidance from deep features to shallower layers, thereby enhancing overall detection efficacy through improved feature learning dynamics.

4.4. Ablation Experiment

To systematically evaluate contributions from the B1-B4 layers within the FFDN-YOLO architecture, we conducted progressive layer deactivation studies under identical exper-

imental configurations, with iterative retraining and validation procedures detailed in Table 5. As these hierarchical components originate from PANet’s structural blueprint, B1’s feature space remains rich in fine-grained spatial details. B2 incorporates localized descriptors while introducing proto-semantic patterns. B3 demonstrates amplified semantic concentrations critical for categorical discrimination, whereas the deepest B4 layer maintains maximal semantic abstraction and global contextual representation. The ablation results confirm each layer’s indispensability, where any single deactivation empirically impairs detection precision across all metrics, proving that their synergistic operation is essential for coherent multi-scale perception.

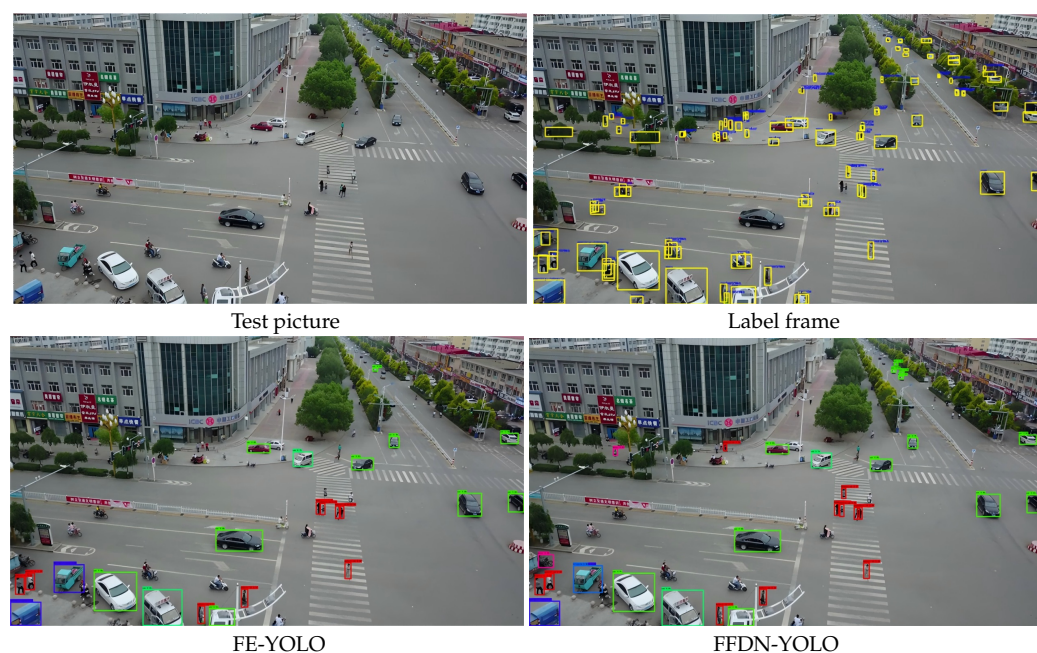


Figure 7. Comparison between FE-YOLO and FFDN-YOLO models.

Table 5. Feature layer closure experiment.

B1	B2	B3	B4	$AR_{0.5}$	$AP_{0.5}$	$AP_{0.75}$	mAP	AP_s	AP_m	AP_l	Params
	✓	✓	✓	0.256	0.243	0.111	0.121	0.062	0.214	0.267	8.557M
✓		✓	✓	0.231	0.228	0.097	0.101	0.054	0.181	0.245	7.379M
✓	✓		✓	0.235	0.230	0.098	0.102	0.055	0.187	0.247	7.047M
✓	✓	✓		0.216	0.206	0.081	0.091	0.036	0.172	0.231	8.254M
✓	✓	✓	✓	0.304	0.281	0.161	0.158	0.082	0.236	0.296	15.515M

4.5. Comparative Experiment

To verify the superiority of the FFDN-YOLO model, we compared it with typical detection models such as Faster-RCNN, SSD, RetinaNet, YOLOv9s, and YOLOv10s. To ensure the fairness of the experiment, all the comparison models were retrained, verified, and tested under the same conditions. The experimental results are shown in Table 6, and it can be seen that each model’s average detection accuracy is relatively low, which again highlights the necessity of developing effective small object detection algorithms. Compared with anchor-based detection algorithms, the overall detection accuracy of anchor-free detection algorithms is higher, indicating that the anchor-free method is more suitable for small object detection. The mAP of the FFDN-YOLO model is 15.8%, which is 2.1%, 1.7%, and 2.9% higher than that of YOLOXs, YOLOv5m, and FOCS, respectively. For the APs index, the detection results of the FFDN-YOLO model are also the best, indicating that the model is more suitable for small object detection in complex environments. Although the

inference speed of the FFDN-YOLO model is not optimal, the trade-off in inference speed is acceptable because the main goal of this paper is to improve the detection accuracy of small objects.

Since the VisDrone2021 [33] dataset contains some indistinguishable object categories, such as “pedestrian” and “people”, the similarity between these two categories is very high, which makes it challenging for the object detection model to distinguish between them. This paper argues that the objective of an object detection algorithm is to detect the category and location of objects in an image, rather than to identify human behavior. Therefore, the “pedestrian” and “people” categories can be combined into a single class to better evaluate the model’s performance. Additionally, because of the limitations of manual labeling, not all small objects in the images are labeled. As a result, when the model detects objects without corresponding labels, the calculation method based on the MS-COCO dataset will treat the detection result as an error detection sample, since there is no corresponding label data to compute it. Thus, there are some limitations in simply using the data in the table to evaluate the model. The detection map of the experimental results from Section 3 is shown. As illustrated in Figure 8, the white elliptical box represents the object detected by the model, which, however, is not marked in the truth label box.

Table 6. Experimental results of different models.

<i>Model</i>	<i>AR_{0.5}</i>	<i>AP_{0.5}</i>	<i>AP_{0.75}</i>	<i>mAP</i>	<i>AP_s</i>	<i>AP_m</i>	<i>AP_l</i>	<i>Params</i>	<i>FPS</i>
YOLOXs	0.266	0.253	0.137	0.137	0.069	0.213	0.290	8.941M	199.3588
CenterNet	0.171	0.193	0.093	0.100	0.037	0.160	0.286	32.665M	225.9244
YOLOV5s	0.226	0.220	0.100	0.110	0.050	0.172	0.279	7.047M	216.7523
YOLOV5m	0.275	0.269	0.133	0.141	0.067	0.217	0.327	20.908M	163.7005
YOLOV7-Tiny	0.224	0.207	0.101	0.107	0.051	0.168	0.257	6.038M	268.9494
SSD-300	0.126	0.110	0.045	0.053	0.010	0.080	0.178	24.81M	231.8313
SSD-640	0.250	0.214	0.117	0.116	0.052	0.182	0.245	24.815M	102.7208
RetinaNet	0.134	0.126	0.083	0.076	0.011	0.131	0.266	36.517M	99.7465
FCOS	0.230	0.230	0.130	0.129	0.052	0.207	0.340	32.132M	98.9351
Faster-RCNN	0.171	0.170	0.067	0.081	0.015	0.138	0.255	136.873M	42.4578
YOLOV3	0.153	0.173	0.058	0.076	0.033	0.113	0.260	61.572M	112.8928
YOLOV4	0.154	0.172	0.080	0.087	0.036	0.134	0.210	63.986M	85.4265
YOLOV7s	0.283	0.270	0.136	0.143	0.073	0.213	0.290	9.159M	192.3077
YOLOV8s	0.257	0.268	0.165	0.158	0.073	0.246	0.374	11.139M	229.9354
YOLOV9s	0.298	0.262	0.154	0.151	0.067	0.235	0.303	15.606M	156.2500
RT-DETR-ResNet18	0.295	0.258	0.142	0.144	0.074	0.213	0.334	20.095M	-
FFDN-YOLO	0.304	0.281	0.161	0.158	0.082	0.236	0.296	15.515M	152.5671

To demonstrate FFDN-YOLO’s excellent performance, this paper selected five models for comparison based on the experimental results in Table 6: YOLOV5m, FCOS, YOLOV7-Tiny, SSD, and YOLOXs. These models were tested on a test image containing multiple small objects, and the test results are shown in Figure 9. Since the categories “pedestrian” and “people” are easily confused in the ground truth labels, we marked them with different colors: the white box represents the “people” category, the purple box represents the “pedestrian” category, and the remaining colors represent other categories, as shown in Figure 9.

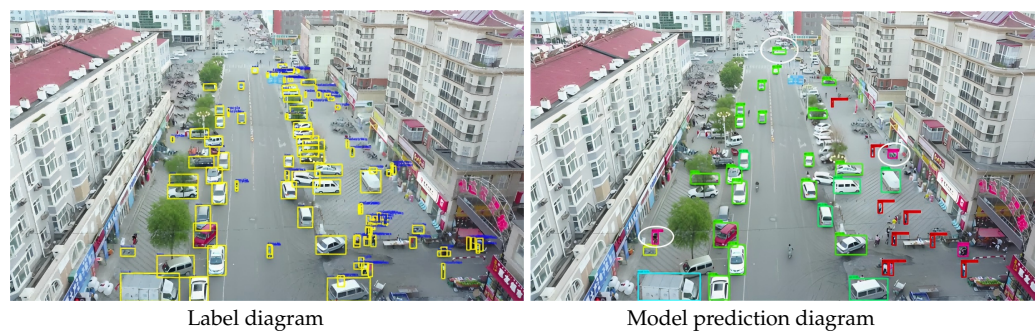


Figure 8. Comparison diagram of model detection and manual labeling.

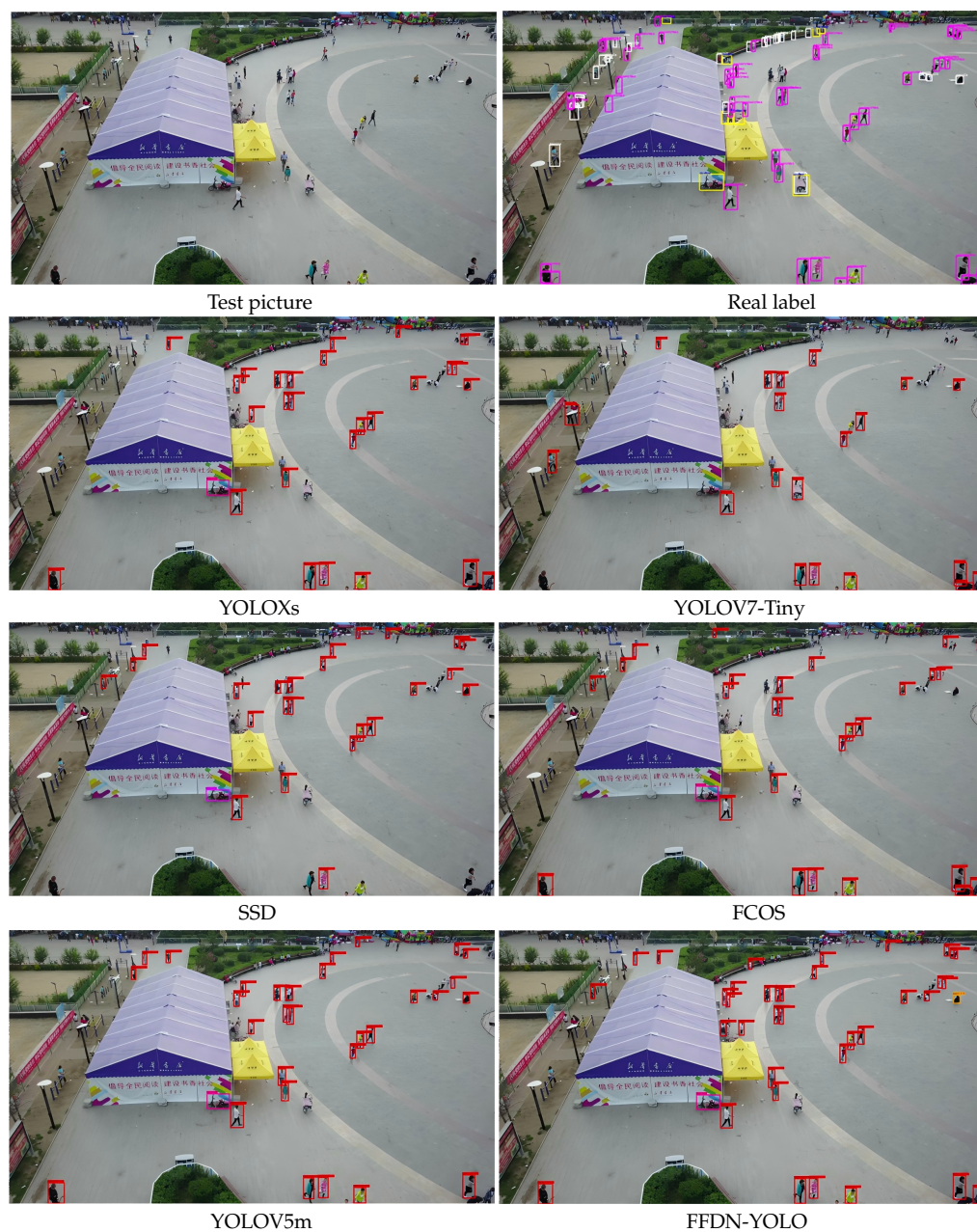


Figure 9. Comparison chart of detection results of different models.

It can be seen from the figure that YOLOXs detected 29 objects, FFDN-YOLO detected 39 objects, and SSD, FCOS, YOLOV5m, and YOLOV7-Tiny detected 26, 28, 31, and 20 objects, respectively. Therefore, the number of objects of interest detected by FFDN-YOLO

is the highest, and the accuracy of the detection boxes is also improved. This shows that FFDN-YOLO can extract more abundant small object feature information through deconvolutional upsampling. Through the FFDN module, the high-level semantic information is reasonably utilized to help the model better learn the characteristics of shallow small objects, thereby improving the classification and regression accuracy of small objects by the detection head.

Finally, this paper shows the loss function comparison diagram of FFDN-YOLO and YOLOXs, as shown in Figure 10. The results show that the FFDN-YOLO model is superior to YOLOXs in terms of convergence speed and convergence value, which indicates that FFDN-YOLO can capture the contextual information of small objects more effectively, thereby learning the global and fine-grained information of small objects more quickly.

4.6. Experiment Summary

To demonstrate the performance of the FFDN-YOLO model more intuitively, the experimental results of the FFDN-YOLO model on the VisDrone2021 dataset are shown in Figure 11. In the case of 1610 samples, the FFDN-YOLO model detected 20,095 car-type targets, 14,378 pedestrian-type targets, 3248 van-type targets, 2998 motor-type targets, 1878 bus-type targets, 1547 truck-type targets, 633 bicycle-type targets, 336 awning-tricycle-type targets, and 318 tricycle-type targets.

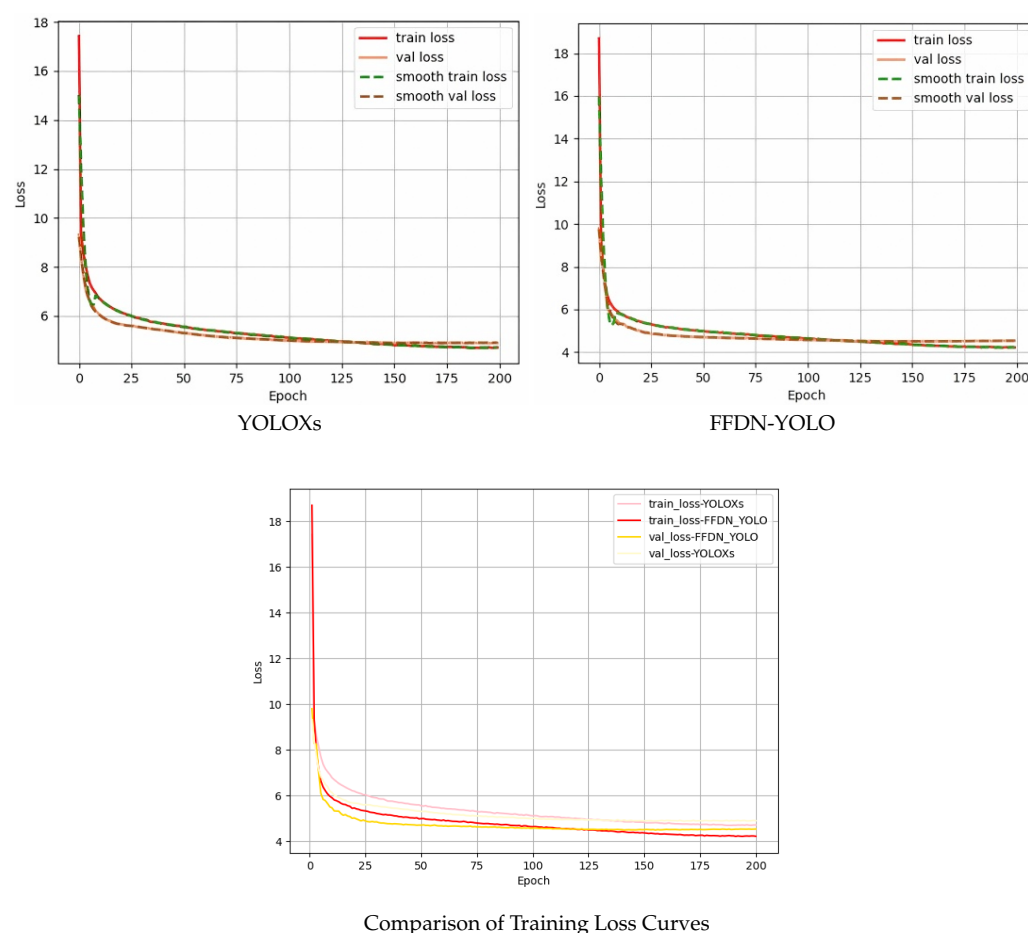


Figure 10. Variation diagram of loss value during model training.

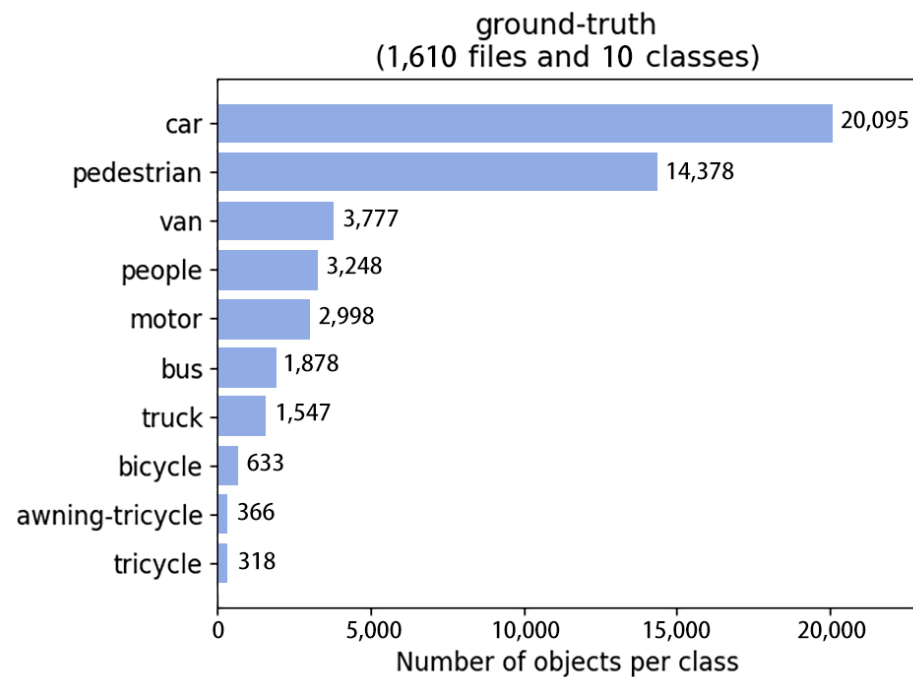


Figure 11. Dataset test diagram.

5. Conclusions

To solve the problem of information loss in the PANet network, this paper designed a feature fusion and distribution module (FFDN), which replaces the deep-to-shallow information transmission path of PANet in the benchmark network. The FFDN module utilizes more global feature information to guide the model in effectively classifying shallow features, thereby reducing the false detection rate and improving the detection accuracy of small objects. To study the influence of different feature distribution methods, we compared the effects of feature splicing and feature addition on the detection model. In addition, to analyze the influence of the downsampling operation on global feature information, this paper compared maximum pooling and average pooling, selecting the optimal downsampling method. Finally, experimental comparison and visual analysis of FFDN-YOLO and various benchmark models were carried out on the VisDrone2021 dataset. The results show that FFDN-YOLO exhibits superior detection performance compared to multiple benchmark models and effectively improves the detection of small objects. However, we believe that there is still significant room for improvement. The next step is to study how to increase detection speed while improving accuracy, in order to contribute more to overall efficiency.

Author Contributions: Conceptualization, J.Q. and T.L.; data curation, T.L., Z.T., J.H. and Y.D.; software, T.L. and J.H.; formal analysis, J.Q.; project administration, J.Q.; supervision, J.Q.; investigation, Y.D., H.Y., and J.H.; visualization, T.L. and Y.D.; writing—original draft, J.Q. and T.L.; writing—review and editing, T.L., Y.D. and Z.T.; funding acquisition, J.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Xi'an Key Laboratory of Advanced Control and Intelligent Process under grant Collaborative Innovation Project of Xi'an Science and Technology Bureau (24KGDW0022) and the Key R&D plan of Shaanxi Province under grant No. 2021ZDLGY04-04.

Data Availability Statement: A vast amount of the research in this paper was based on the publicly available dataset VisDrone2021, and the role of the homemade dataset is for validation purposes only, so the homemade data in this study are available from the corresponding authors upon request.

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

References

1. Zhao, J.; Liang, B.; Chen, Q. The key technology toward the self-driving car. *Int. J. Intell. Unmanned Syst.* **2018**, *6*, 2–20. [\[CrossRef\]](#)
2. Hayes-Roth, B.; Washington, R.; Hewett, R.; Hewett, M.; Seiver, A. Intelligent Monitoring and Control. In Proceedings of the IJCAI, Detroit, MI, USA, 20–25 August 1989; Volume 89, pp. 243–249.
3. Mahaur, B.; Mishra, K. Small-object detection based on YOLOv5 in autonomous driving systems. *Pattern Recognit. Lett.* **2023**, *168*, 115–122. [\[CrossRef\]](#)
4. Koyun, O.C.; Keser, R.K.; Akkaya I.B.; Töreyin, B.U. Focus-and-Detect: A small object detection framework for aerial images. *Signal Process. Image Commun.* **2022**, *104*, 116675. [\[CrossRef\]](#)
5. Sun, Z.; Lin, M.; Sun, X.; Tan, Z.; Li, H.; Jin, R. Mae-det: Revisiting maximum entropy principle in zero-shot nas for efficient object detection. *arXiv* **2021**, arXiv:2111.13336.
6. Zong, Z.; Song, G.; Liu, Y. Detrs with collaborative hybrid assignments training. In Proceedings of the International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 6725–6735.
7. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 5998–6008.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
9. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
10. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
11. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. *arXiv* **2019**, arXiv:1904.01355.
12. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding Yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
13. Liu, R.; Deng, H.; Huang, Y.; Shi, X.; Lu, L.; Sun, W.; Wang, X.; Dai, J.; Li, H. FuseFormer: Fusing Fine-Grained Information in Transformers for Video Inpainting. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; IEEE: New York, NY, USA, 2021.
14. 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.
15. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
16. Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516.
17. Ghiasi, G.; Lin, T.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7036–7045.
18. Yu, H.; Huang, J.; Zhao, F.; Gu, J.; Loy, C.C.; Meng, D.; Li, C. Deep fourier up-sampling. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 22995–23008.
19. Zhou, D.-X. Theory of deep convolutional neural networks: Downsampling. *Neural Netw.* **2020**, *124*, 319–327. [\[CrossRef\]](#) [\[PubMed\]](#)
20. 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, 18–23 June 2018; pp. 8759–8768.
21. Duan, Y.; Yang, D.; Qu, X.; Zhang, L.; Chao, L.; Gan, P.; Yuan, S.; Qin, H.; Qu, J. Lcire-net: Lightweight cross-modal information interaction for road feature extraction from remote sensing images and gps trajectory/lidar. *IEEE Trans. Geosci. Remote Sens.* **2024**. [\[CrossRef\]](#)
22. Duan, Y.; Qu, J.; Zhang, L.; Qu, X.; Yang, D. Lgrf-net: A novel hybrid attention network for lightweight global road feature extraction. *IEEE Trans. Geosci. Remote Sens.* **2024**. [\[CrossRef\]](#)
23. Bonny, M.Z.; Uddin, M.S. Feature-based image stitching algorithms. In Proceedings of the 2016 International Workshop on Computational Intelligence (IWCI), Dhaka, Bangladesh, 12–13 December 2016.
24. Hamed, T.; Dara, R.; Kremer, S.C. Network intrusion detection system based on recursive feature addition and bigram technique. *Comput. Secur.* **2018**, *73*, 137–155. [\[CrossRef\]](#)
25. Yuan, S.; Qin, H.L.; Yan, X.; Akhtar, N.; Mian, A. SCTransNet: Spatial-Channel Cross Transformer Network for Infrared Small Target Detection. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5002615. [\[CrossRef\]](#)
26. Yuan, S.; Qin, H.; Yan, X.; Akhtar, N.; Yang, S.; Yang, S. Irstdid-800: A benchmark analysis of infrared small target detection-oriented image destriping. *IEEE Trans. Geosci. Remote Sens.* **2024**. [\[CrossRef\]](#)
27. Lam, N.S.-N. Spatial Interpolation Methods: A Review. *Am. Cartogr.* **1983**, *10*, 129–150. [\[CrossRef\]](#)

28. Burrus, C.S.; Parks, T.W. *DFT/FFT and Convolution Algorithms*; John Wiley & Sons: London, UK, 1985.
29. Ye, C.; Evanusa, M.; He, H.; Mitrokhin, A.; Goldstein, T.; Yorke, J.A.; Fermüller, C.; Aloimonos, Y. Network deconvolution. *arXiv* **2019**, arXiv:1905.11926.
30. Qu, J.; Tang, Z.; Zhang, L.; Zhang, Y.; Zhang, Z. Remote sensing small object detection network based on attention mechanism and multi-scale feature fusion. *Remote Sens.* **2023**, *15*, 2728. [[CrossRef](#)]
31. Wang, C.; He, W.; Nie, Y.; Guo, J.; Liu, C.; Wang, Y.; Han, K. Gold-YOLO: Efficient object detector via gather-and-distribute mechanism. *arXiv* **2024**, arXiv:2309.11331.
32. Rogova, G.L.; Nimier, V. Reliability in information fusion: Literature survey. In Proceedings of the Seventh International Conference on Information Fusion, Stockholm, Sweden, 28 June–1 July 2004; Volume 2, pp. 1158–1165.
33. Cao, Y.; He, Z.; Wang, L.; Wang, W.; Yuan, Y.; Zhang, D.; Zhang, J.; Zhu, P.; Van Gool, L.; Han, J.; et al. VisDrone-DET2021: The vision meets drone object detection challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 11–17 October 2021; pp. 2847–2854.

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