A Universal Structure of YOLO Series Small Object Detection Models

Shengchao Hu[1,2], Xiao Liu[1,2], Weijun Wang[1,2], Tianlun Huang[1,2], and Wei Feng*[1,2]

[1] ShenZhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[2] University of Chinese Academy of Sciences, Beijing, China
sc.hu@siat.ac.cn, xiao.liu1@siat.ac.cn, wj.wang@giat.ac.cn, {tl.huang1,wei.feng}@siat.ac.cn

Abstract. The YOLO series detection models play a crucial role in target detection tasks. However, these models are typically trained on datasets with standard angles. For datasets like Visdron2021 and Tinyperson, there are challenges related to small, dense, and numerous objects that conventional object detection models struggle to detect effectively. Therefore, we propose a universal structure for all YOLO series models to enhance their capability to detect small objects. We first use a large-scale feature map as a new detection branch to address the issue of feature loss with small objects. Secondly, we have developed a detail-guide-block (DGB) to enhance the model's ability in detailed detection, along with a feature-refine-module (FRM) aimed at mitigating the problem of feature flattening caused by upsampling. Finally, we removed the fourth detection branch that did not significantly improve detection accuracy, which can to some extent improve the execution speed of the model and reduce its complexity. We have ported our structure on YOLOX, YOLOv7, and YOLOv8, and conducted extensive experiments on Visdrone2021 and Tinypeson datasets. The experimental data demonstrate that our improved models consistently outperform the original model in terms of performance.

Keywords: YOLO Series Model · Small Object Detection · Universal Structure

1 Introduction

Object detection is an important research direction in the field of computer vision and is also the foundation for other complex visual tasks. In some high-level visual tasks such as scene understanding, object tracking, image description, and event detection, the application of object detection is often involved. Despite significant breakthroughs in object detection algorithms, the detection of small objects still needs improvement. Compared to conventional-sized targets, small objects often lack sufficient visual information, making distinguishing them from background or similar targets difficult.

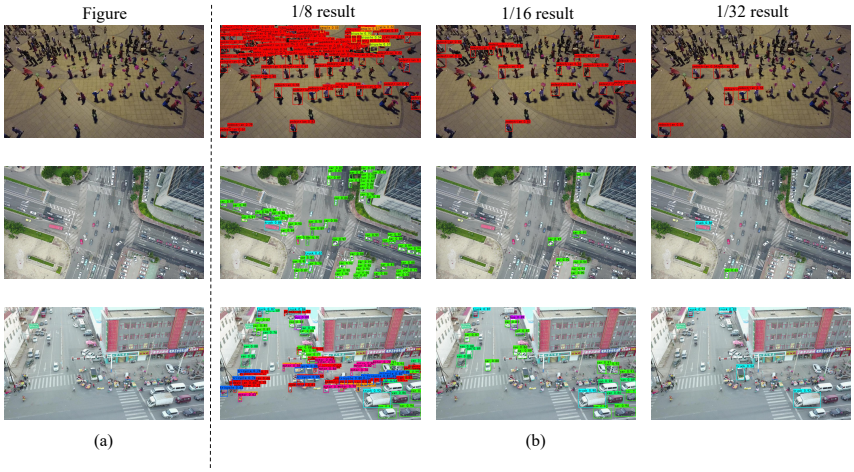

Fig. 1: (a). Explanation of the characteristics of the dataset. (b). From left to right, represent the predicted results of feature maps with 8-fold downsampling, 16-fold downsampling, and 32-fold downsampling, respectively.

In recent years, deep convolutional neural networks have made significant progress in object detection tasks. Some benchmark datasets such as MS COCO [22] and PASCALVOC [6] have greatly promoted the development of object detection applications. These datasets are usually based on the human perspective, and many excellent detection models [21,25,29,32,41] have been proposed based on the above datasets. However, these models are not suitable for some special scenarios, such as drone aerial photography [2] and satellite remote-sensing images [35]. In these datasets, the following challenges are often faced: 1). Few available features. Small targets occupy too little size in the image, lacking sufficient visual information. Low-resolution small target visualization information is limited, making it difficult to extract discriminative features and easily affected by environmental factors, which in turn makes it difficult for detection models to accurately locate and recognize small targets. 2). High positioning accuracy requirement. Due to the small coverage area in the image, even a single pixel offset in the prediction boundary box during the prediction process can have a significant impact on the location of small targets. 3). Object aggregation. In these datasets, the probability of object clustering is high. When object aggregation occurs, the small targets adjacent to the aggregation region will be aggregated into one point after repeated downsampling to the deep feature map, resulting in an indistinct detection result. Which are intuitively illustrated by some cases in Fig. 1(a).

Looking back on the history of object detection models, the YOLO series of detection models [1, 8, 13, 14, 29–31, 34] has played an indelible role of great significance. The core idea of YOLO [29] is to transform object detection into a

regression problem, using the entire image as input to the neural network to get the position and category of the bounding box. To further improve the model's prediction accuracy, subsequent work adopted multi-scale detection similar to SSD [25] to perform object detection on 8-fold, 16-fold, and 32-fold downsampling feature maps, respectively. This design enables the model to better detect targets of different sizes. However, on some special datasets [2,39], feature maps with 32-fold downsampling are almost unable to predict the correct target. We present the test results in Fig. 1(b).

Based on the above observations, we propose a universal YOLO model structure. It can be used in any YOLO series model to improve prediction accuracy for small targets. In our models, we use a 4-fold downsampling feature map instead of a 32-fold downsampling feature map for detection. And use special convolution methods to prevent model speed from decreasing. Fig. 2(b) shows the structure of our models. Our main contributions are summarized as follows:

- We propose a universal model structure for small object detection, which can be used in any YOLO series model to improve the recognition ability of small objects.
- We propose DGB, which can effectively enhance the model's detection ability for small target objects.
- We propose FRM, which can effectively alleviate the problem of feature flattening caused by upsampling.
- We provide useful skill kits and filter out some useless techniques for small object detection tasks.
- We validated on YOLOX, YOLOv7, and YOLOv8, and achieved stable accuracy improvements on the Visdrone2021 and Tinyperson datasets.

2 Related Work

General Object Detection Object detection is one of the core tasks of computer vision and has wide applications in various fields of society. The first attempt to use deep learning for object detection was R-CNN [9]. R-CNN uses Convolutional-Neural-Network(CNN) on region proposals generated by using selective search. Although it has achieved good detection results, it cannot meet the real-time requirements of many tasks. Because each proposed area needs to pass through CNN sequentially, which is very time-consuming. Faster R-CNN [32] specifically proposes to generate refined proposals by designing a region proposal network. It only performs one feature extraction stage for all region proposals, making it faster than R-CNN. R-FCN [3] was introduced to efficiently perform region-wise full convolutions compared to the computations of heavy region-wise CNN during a pooling operation. Mask R-CNN [11] proposes adding a mask prediction branch to improve the performance of object detection and instance segmentation.

The above methods all have a Region-Proposal-Network(RPN) for generating candidate target boxes and a network for classifying and regressing the bounding boxes of these boxes. According to their structural characteristics, they all

belong to two-stage object detection methods in classification. Although they have high detection accuracy, they are usually unable to complete real-time detection tasks. To meet the requirements of real-time performance, researchers have begun to study more efficient algorithms and proposed one-stage object detection algorithms. Contrary to two-stage methods, one-stage methods were proposed to detect objects by directly conducting classification and localization with the predefined anchor boxes. SSD [25] introduces multi-scale object detection based on multi-layer pyramid features, with shallow and deep feature maps used for detecting small and large objects, respectively. RetinaNet [21] proposed a focal loss algorithm to address the imbalance between foreground and background classes. RFBNet [5] was introduced to combine multiple branches of multi-scale receptive fields for enhancing feature representation. To address the complex scale changes in object detection, some methods [18, 26] have explored the exploration of multi-scale pyramid features These methods include additional top-down paths and horizontal connections, and detect objects at each scale from the corresponding layers of these pyramids.

Small Object Detection In small object detection tasks, the targets that need to be detected generally have the problem of small coverage area and high target density. To address the above difficulties, some methods improve the model's recognition of small targets by introducing custom components. These works cover several aspects such as super-resolution schemes [17], loss function optimization [23], feature fusion [40], and multi-scale feature learning [18].

Since the detection of small targets mainly benefits from large-scale feature mapping, some previous small target detectors have adopted super-resolution (SR), which can be roughly divided into image-level SR and feature-level SR. For image-level SR, Hu and Ramanan [12] proposed using bilinear interpolation to achieve large-scale input images. Fooks et al. [7] introduced a method of generating super-resolution facial images using generative adversarial networks. However, these methods increase inference time because the input image is large-scale and not end-to-end trainable. Unlike image-level SR, feature-level SR directly performs super-resolution processing on features. Perceptual GAN [17] proposed to enrich the small objects' features by narrowing the difference in the presentation of small and large objects. EFPN [4] proposes an extended feature pyramid network to utilize additional high-resolution pyramid features for detecting small objects. Although existing methods using additional networks for adversarial training can improve the detection performance of small targets to a certain extent, adversarial training is not stable. The optimization of loss function can help detect small objects. Liu et al. [23] proposed a feedback-driven loss function, taking loss distribution cues as feedback signals, which can be used for balance training of the model. Leng et al. [15] designed a context-guided inference network (CRNet) to explore the relationships between objects and use easily detectable objects to help understand difficult objects. They also proposed PRDet [16], which distinguishes hard regions from ordinary regions under reverse attention guidance and refocuses hard regions with the help of region-

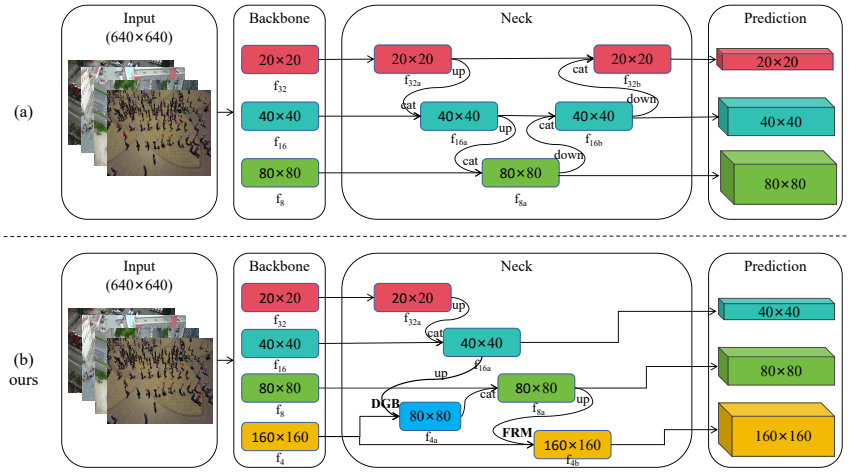

Fig. 2: Comparison diagram between the model structure of YOLO series models and our model structure. Only the flow and size changes of the data in the model are displayed.

specific context. Feature fusion can fuse features from different levels and improve the detection accuracy of the model. FA-SSD [19] uses attention modules and feature fusion to integrate features at different levels. In addition, multi-scale feature learning is also an important means of detecting small targets. SSD [25] utilizes multi-layer features to predict objects of different sizes, which to some extent improves detection accuracy. FPN [20] uses a top-down architecture and horizontal connections to integrate features at different levels to better detect small targets.

3 Method

In recent years, deep convolutional neural networks have made significant progress in object detection tasks, and many interesting and efficient YOLO series models have been proposed. However, most of these models are designed for natural scene images. For some images in special scenarios, as shown in Fig. 1(a), it is unreasonable to directly apply previous models for object detection. Firstly, these images typically contain high-density objects, which can lead to occlusion between objects. Secondly, the target object in the image with a small area is often overlooked after multiple convolution operations.

3.1 Overview of YOLO Series Models

With the rapid development of deep learning, YOLO [29] has also undergone several versions of optimization. The maximum change in model structure is

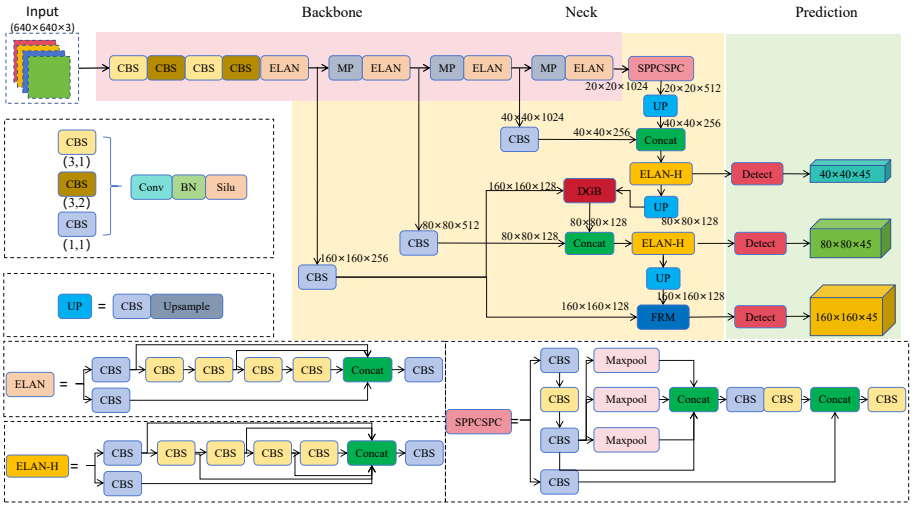

Fig. 3: Improved model based on YOLOv7. We only modified the structure of the model and did not modify the special modules of the original YOLOv7 model. DGB and FRM are special modules we propose.

from YOLO [29] to YOLOv3 [31]. Since YOLOv4 [1], some versions have mostly adopted a model structure similar to YOLOv3 [31], and have designed some special modules based on the previous ones to improve model performance to a certain extent. These models mainly consist of three parts: Backbone, Neck, and Prediction. Backbone is the foundation of the YOLO series models, typically a pre-trained convolutional neural network (CNN) such as ResNet, Darknet, etc. Neck is located between backbone and prediction, and its main function is to connect backbone and prediction, while enhancing the model's feature representation ability. Neck is usually composed of multiple modules, such as FPN [20] (Feature-Pyramid-Network), PAN [24](Path-Aggregation-Network), etc. These modules can integrate features from different levels and improve the model's detection ability for multi-scale targets. Prediction is the final part of the YOLO series of models, which is responsible for object detection based on the extracted features. Prediction typically includes multiple prediction layers, each corresponding to feature maps of different scales. At each prediction layer, the model generates a series of bounding boxes and their corresponding category probabilities and confidence levels. We have shown the structure in Fig. 2(a).

3.2 A Universal Structure of YOLO Series Models for Small Object Detection

To address the issue of insufficient detection capability of existing models for small targets, we propose a universal neck for YOLO series models. It can be applied to any YOLO series model to improve the detection accuracy of the

original model for small targets. To make our model inherit the advantages of the YOLO series model, we have retained all parts of the original model except for the neck. After extensive analysis, we have found that the main reason for the low accuracy of small object detection in existing models. Due to the small area occupied by small targets in the input image, some high-level feature maps obtained through multi-layer convolution cannot retain the features of small target objects, resulting in the model ignoring small targets in the final prediction.

To improve the above issues, we consider adding a new prediction branch to the model. As shown in Fig. 2(a), most models adopt feature maps downsampled at 8x, 16x, and 32x for object detection. The smaller the downsampling rate, the more image details are retained. Therefore, we consider object detection on feature maps with 4x downsampling rate. So, we added a new prediction branch based on the original model. Although the model's prediction ability for small targets has been improved, feature maps with excessive size will seriously affect the execution speed of the model during convolution. This method of exchanging speed for accuracy is not advisable, as in most industrial tasks, the execution speed of the model is more important than accuracy. In order to ensure the execution speed of the model, we refuse to use convolution operations with a kernel greater than 1 in the new prediction branch. We present our model structure in Fig. 2(b). In our model, we use the 4x downsampled feature map f_4 as the source of detailed features, as it has more detailed information compared to other levels of feature maps. We also input f_4 and f_{16a} into a DGB to guide f_{16a} in obtaining more detailed features, thereby improving the prediction accuracy of the model. In DGB, we only used 1×1 convolution operation, so it will not have a significant impact on the execution speed of the model. Subsequently, we input feature maps f_{8a} and f_4 containing detailed features into a feature-refinement-module (FRM) to purify and refine feature information. In Fig. 6, we present the comparison results of the feature maps before and after refinement. Finally, input the purified feature map f_{4b} into the prediction for detection.

For the previous YOLO series models, most of them were designed with structures for datasets such as MS COCO [22] and PASCALVOC [6], and feature maps with 32x downsampling were typically responsible for predicting large targets in these models. After some experimental analysis, we found that the feature map with 32x downsampling has almost no effect on detecting small targets. Because it usually cannot effectively predict the target, as shown in Fig. 1(b). Therefore, in our model, we only retained three lower-level detection heads, which did not affect the prediction accuracy of our model. In order to make our model structure clearer, we ported our structure using YOLOv7 [34] as the basic model. As shown in Fig. 3, we have presented the complete model structure. Except for the modifications mentioned above, all others are retained as the original structure of YOLOv7.

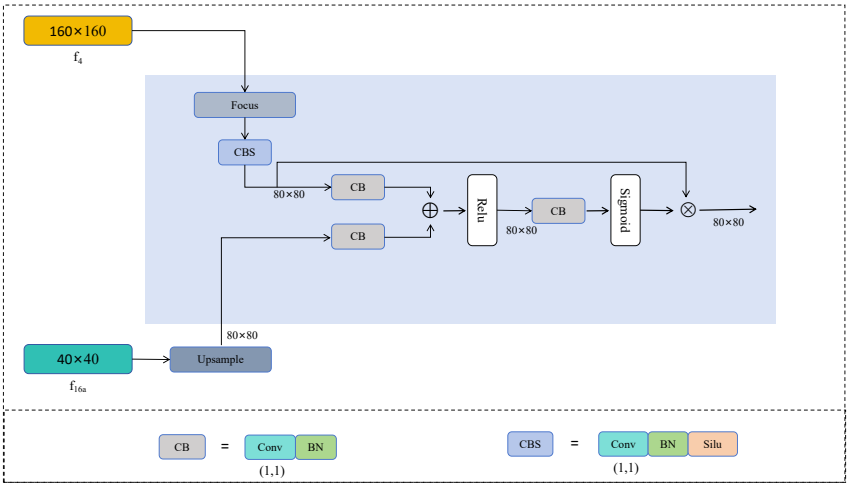

Fig. 4: The architecture of detail guide block(DGB). Convolutional kernels are all 1 in size.

3.3 DGB: Detail Guide Block

Based on experimental analysis, we found that the feature map f_4 with 4x downsampling contains more detailed information than other feature maps. Therefore, its detection ability for small targets is higher than other levels of feature maps. To spread its advantages to other feature maps, we designed a detail-guide-block (DGB) to improve the model's ability to extract small object features. Inspired by attention-unet [28], we have adopted cross-attention as the main body of DGB in our design. Through cross-attention, the input feature map f_{16a} can gradually learn to recognize the detailed features in feature map f_4, thereby propagating the low-level detailed information to the high-level feature map. We show the structure of DGB in Fig. 4, which has two inputs: the low-level feature map f_4 and the high-level feature map f_{16a}. The size of f_4 is twice that of f_{16a}, so we first use a focus [13] operation on f_4 to unify the two sizes. In DGB, we only use convolution operations with a kernel of 1, which not only reduces the impact on model execution speed but also reduces the influence of irrelevant factors on detail features. If we represent a convolution operation and a batchnorm with the symbol CB:

$$CB(x) = BatchNorm\left(Conv\left(x\right)\right) \tag{1}$$

So, the output of DGB can be expressed using the following formula:

$$FS_{out} = CB(focus(f_4)) \tag{2}$$

$$RU_{out} = relu(FS_{out} + CB(f_{16a})) \tag{3}$$

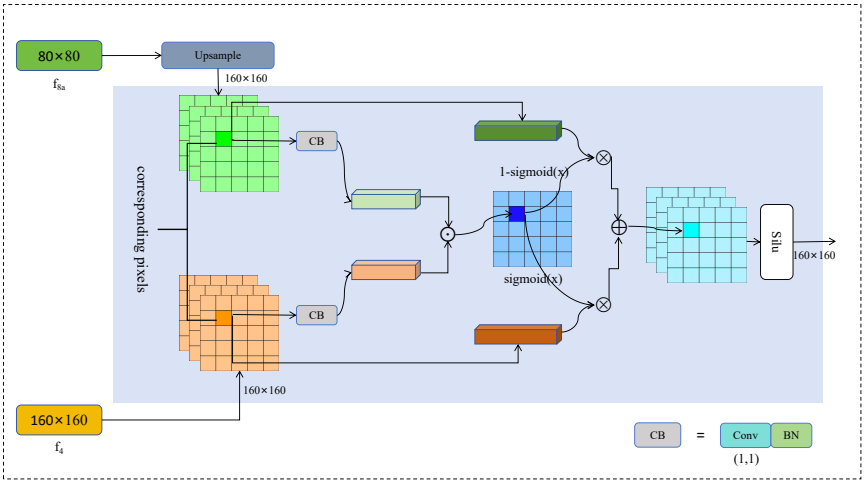

Fig. 5: The architecture of feature refine module(FRM). Convolutional kernels are all 1 in size.

$$DGB_{out} = FS_{out} * sigmoid(CB(RU_{out})) \tag{4}$$

where, f_4 and f_{16a} represent feature maps at different levels, which can be found in Fig. 2(b). To verify its effectiveness, we conducted ablation experiments in Chapter 4 and presented the experimental results in Table. 1.

3.4 FRM: Feature Refine Module

At the end of the model, we need to upsample the feature map f_{8a} after feature aggregation and add it with f_4. However, the upsampled feature map f_{8a} has local similarity, meaning that other pixel features around a certain pixel feature will be very similar to it. This is caused by the upsampling method, which leads to feature flattening. Therefore, we propose a feature-refine-module (FRM) to refine the feature map. The main idea of FRM is to use a learned weight parameter to replace direct addition. The basic concept of FRM comes from attention mechanisms [33]. If the vectors corresponding to pixels in the f_4 and f_{8a} branch feature maps are defined as $\overrightarrow{V_4}$ and $\overrightarrow{V_{8a}}$, respectively, the output of FRM can be represented as:

$$FRM_{out} = \delta * \overrightarrow{V_4} + (1 - \delta) * \overrightarrow{V_{8a}} \tag{5}$$

where, f_4 and f_{8a} represent feature maps at different levels, which can be found in Fig. 2(b). δ represents the learned weight, it can be expressed as:

$$\delta = sigmoid(F_{cb}(\overrightarrow{V_4}) \cdot F_{cb}(\overrightarrow{V_{8a}})) \tag{6}$$

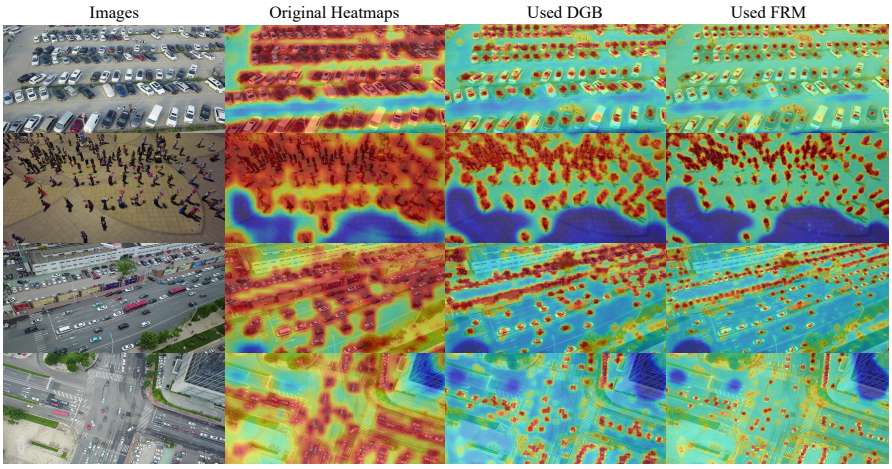

Fig. 6: Visualization results of feature maps. From left to right represents the trend of changes in the feature map.

In order to demonstrate its effectiveness more intuitively, we conducted a detailed comparative experiment in Chapter 4. Visualize the experimental results through Fig. 6 and Table. 1.

4 Experiments

In this section, we first introduce the datasets and the implementation details. Next, we investigate the effects of each component of our proposed method. Finally, we report the comparison results with other object detection algorithms on Visdrone2021 and Tinyperson datasets.

4.1 Datasets

Visdrone2021. Visdrone2021 [2] is one of the most popular small object detection datasets. It is a city aerial image dataset based on drone perspective. The images were divided into sets of sizes 6,471, 548, and 3,190 for training, validation, and testing. The dataset contains 10 detection categories, mainly vehicles and pedestrians that are more common in the city. The images have different resolutions. But in this paper, we adjusted the size of all pictures to 640×640 in the experiment.

Tinyperson. TinyPerson [39] is a small-scale dataset composed only of ultra-small objects. The objects in this dataset are all less than 20 pixels. This dataset contains a total of 1610 images, of which 794 were selected as the training set, 596

as the testing set, and 220 as the validation set. This dataset mainly involves two types of detection objects, both of which are humans but are divided in detail into people on land and people in water.

4.2 Implementation Details

Training Details. We have implemented all of our improved models on Pytorch 1.10.1. All of our models were trained and tested using NVIDIA RTX3090 GPU. During the training phase, we will use pre-trained weights from the original model. Since we made improvements to the model in the Neck and Prediction sections, we only used the weights from the original model backbone. We use mini-batch stochastic gradient descent (SGD) with momentum 0.9, and weight decay. The batch size of the Visdrone2021 and Tinyperson datasets is set to 8.

Data Augmentation. In all the experiments, we only resized the images without cropping them. We use mosaic data augmentation to expand the training set. We set 300 training epochs for all models and used mosaic data augmentation in the top 70% of the training epochs. We will freeze the weight of the backbone for the first 50 training epochs and set the batch to 16.

4.3 Ablation Study

In this section, we analyze the effectiveness of each component proposed in our method.

Effectiveness of DGB. To demonstrate the effectiveness of DGB, we endeavor to substitute it with Add while maintaining the remainder of the model unchanged. The results are shown in Table. 1. Utilizing DGB, our models can provide more detailed low-level features to enhance the recognition of small targets by high-level features, thereby improving the model's capability. To get a more convincing result, we visualized both the DGB-processed feature maps and the original feature maps. As shown in Fig. 6, the DGB-processed feature maps put more attention on the target object. This greatly improves our model's ability to handle detail. Therefore, our model can be more accurate for small objects.

Effectiveness of FRM. In FRM, we replace traditional feature addition with a learned weight. This can help our model better handle the features of detailed positions and reduce the impact of feature flattening caused by upsampling in the model. To demonstrate the effectiveness of FRM, we use Add instead of FRM. In Table. 1, using FRM has higher prediction accuracy than using Add directly. At the same time, we visualized the feature map after FRM and the feature map after DGB. As shown in Fig. 6, the feature maps processed by FRM pay more attention to the boundary information of the target object than the DGB-processed feature maps.

Table 1: Ablation study of DGB and FRM.

Base Model	Resolution	Module1		Module2		AP50(%)	mAP(%)
		Add	DGB	Add	FRM		
yolov7_l [34]	640×640	√		√		39.2	22.8
yolov7_l [34]	640×640		√	√		40.7	23.5
yolov7_l [34]	640×640	√			√	41.8	24.1
yolov7_l [34]	640×640		√		√	**43.7**	**25.6**

Table 2: Ablation study of model structure and prediction head nums.

Base Model	Resolution	Head Nums		GFLOPs	Params/M	AP50(%)	mAP(%)
		Three	Four				
yolox_l(ours)	640×640	√		135.7 **(-24.8)**	43.4 **(-11.4)**	42.9	25.0
yolox_l [8]	640×640		√	160.5	54.8	43.0 **(+0.1)**	25.0 **(+0.0)**
yolov7_l(ours)	640×640	√		91.5 **(-19.8)**	29.85 **(-8.65)**	43.7	25.6
yolov7_l [34]	640×640		√	111.3	38.5	43.9 **(+0.2)**	25.7 **(+0.1)**
yolov8_l(ours)	640×640	√		149.6 **(-20.3)**	36.8 **(-7.7)**	46.4	29.1
yolov8_l [14]	640×640		√	170.3	44.5	46.4 **(+0.0)**	29.2 **(+0.1)**

Effectiveness of Extra Prediction Head. In our models, we use the 4x downsampling feature map instead of the 32-fold downsampling feature map in the original model to detect small targets. Because we found that the 32x downsampled feature map cannot effectively predict small targets, so we used the 4x downsampled feature map with more detailed information. To verify the effectiveness of removing the detection head of the 32x downsampled feature map, we will compare the use of four detection heads with the use of three detection heads. As shown in Table. 2, the use of four detection heads hardly improves the detection accuracy but significantly increases the size and computation of the model.

4.4 Comparison

In this section, we compare our method with other existing state-of-the-art methods on the Visdrone2021 and Tinyperson datasets.

Visdrone2021. As shown in Table. 3, we demonstrate the detection accuracy and inference speed of our proposed methods on the Visdrone2021 validation

Table 3: Comparisons with other state-of-the-art methods on Visdrone2021. - indicates the method do not publish the results. The model marked with * are tested on our platform.

Model	GPU	Resolution	GFLOPs	Params/M	FPS	AP50(%)	mAP(%)
VAMYOLO [38]	RTX 3090	640×640	-	57.53	-	39.8	24.4
DCYOLOv8 [27]	RTX 3090	640×640	-	-	-	41.5	24.7
LVYOLO [36]	RTX 3090	640×640	-	36.6	-	41.7	25.6
YOLOERF [37]	RTX 3090	640×640	-	5.9	-	42.0	23.6
CSYOLOv8 [10]	RTX 3090	640×640	-	-	-	42.6	25.7
yolov5_x* [13]	RTX 3090	640×640	214.0	87.3	62.2	33.1	18.4
yolox_m* [8]	RTX 3090	640×640	69.7	25.1	80.1	38.8	22.2
yolox_l* [8]	RTX 3090	640×640	150.5	54.0	65.3	40.2	22.9
yolox_l(ours)	RTX 3090	640×640	**135.7 (-14.8)**	**43.4 (-10.6)**	**74.3 (+9.0)**	**42.9 (+2.7)**	**25.0 (+2.1)**
yolov7_l* [34]	RTX 3090	640×640	101.3	37.2	83.3	41.2	23.3
yolov7_l(ours)	RTX 3090	640×640	**91.5 (-9.8)**	**29.85 (-7.35)**	**90.1 (+6.8)**	**43.7 (+2.5)**	**25.6 (+2.3)**
yolov8_s* [14]	RTX 3090	640×640	23.6	11.1	125.2	41.4	25.1
yolov8_l* [14]	RTX 3090	640×640	160.3	43.6	66.2	44.0	27.2
yolov8_l(ours)	RTX 3090	640×640	**149.6 (-10.7)**	**36.8 (-6.8)**	**73.5 (+7.3)**	**46.4 (+2.4)**	**29.1 (+1.9)**

set. We compared our original model in terms of speed and accuracy, and our model has shown some improvement in speed and accuracy. And our model will also reduce the size and complexity of the model to a certain extent. We also compared the prediction accuracy with models used for small object detection in the past two years and achieved the best results. To compare fairness, we used the same experimental platform and input size. In Table. 3, our model based on $yolov8-l$ achieved 29.1% mAP, which has surpassed the prediction accuracy of all the above models.

Tinyperson. We also compared our models with other models on the Tinyperson dataset. As shown in Table. 4, with an input size of 640×640, the prediction accuracy of our models has been steadily improved compared with the original models. Our models are consistent across both the Visdrone2021 dataset and the Tinyperson dataset, so the size and computational complexity of the model do not change depending on the dataset.

5 Conclusion

In this paper, a universal small object detection model structure is proposed which is suitable for all YOLO series models. We redesigned the structure of the Neck portion of the YOLO series model and used a larger 4x downsampling feature map to predict small targets. In our newly designed Neck, we have used

Table 4: Comparisons with other methods on Tinyperson. The model marked with * are tested on our platform.

Model	GPU	Resolution	AP50(%)	mAP(%)
yolov5_x* [13]	RTX 3090	640×640	14.21	4.32
yolox_l* [8]	RTX 3090	640×640	16.82	5.43
yolox_l (ours)	RTX 3090	640×640	**20.22 (+3.4)**	**6.71 (+1.28)**
yolov7_l* [34]	RTX 3090	640×640	17.5	5.65
yolov7_l (ours)	RTX 3090	640×640	**22.43 (+4.93)**	**7.28 (+1.63)**
yolov8_s* [14]	RTX 3090	640×640	16.42	5.5
yolov8_l* [14]	RTX 3090	640×640	20.65	6.82
yolov8_l (ours)	RTX 3090	640×640	**24.50 (+3.85)**	**8.56 (+1.74)**

a detail-guide-block(DGB) between 4x downsampling features and 16x downsampling features to help our models learn more detail features. Therefore, our model's ability to detect small targets has been greatly improved. At the same time, we used a feature-refine-module(FRM) between the 8x downsampled feature map and the 4x downsampled feature map to reduce the effect of feature flattening caused by upsampling. Compared with the original models, our improved models have improved prediction accuracy, inference speed, model size, and computational complexity. The structure proposed by us can be used in any YOLO series model to improve the detection accuracy of small targets.

Acknowledgement This work was supported by the National Key R&D Program of China (2023YFB4705002), the National Natural Science Foundation of China (U20A20283), the Guangdong Provincial Key Laboratory of Construction Robotics and Intelligent Construction (2022KSYS013), the CAS Science and Technology Service Network Plan (STS) - Huangpu Special Project (No. STS-HP-202302), the Science and Technology Cooperation Special Project of Hubei Province and the Chinese Academy of Sciences (2023-01-08), and the Open Fund of Guangdong Key Laboratory of Modern Control Technology (GDKLMCT202402).

References

1. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
2. Cao, Y., He, Z., Wang, L., Wang, W., Yuan, Y., Zhang, D., Zhang, J., Zhu, P., Van Gool, L., Han, J., Hoi, S., Hu, Q., Liu, M., Cheng, C., Liu, F., Cao, G., Li, G., Wang, H., He, J., Wan, J., Wan, Q., Zhao, Q., Lyu, S., Zhao, W., Lu, X., Zhu, X., Liu, Y., Lv, Y., Ma, Y., Yang, Y., Wang, Z., Xu, Z., Luo, Z., Zhang, Z., Zhang, Z., Li, Z., Zhang, Z.: Visdrone-det2021: The vision meets drone object detection challenge results. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW). pp. 2847–2854 (2021). https://doi.org/10.1109/ICCVW54120.2021.00319
3. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems **29** (2016)
4. Deng, C., Wang, M., Liu, L., Liu, Y., Jiang, Y.: Extended feature pyramid network for small object detection. IEEE Transactions on Multimedia **24**, 1968–1979 (2021)
5. Deng, L., Yang, M., Li, T., He, Y., Wang, C.: Rfbnet: deep multimodal networks with residual fusion blocks for rgb-d semantic segmentation. arXiv preprint arXiv:1907.00135 (2019)
6. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**, 303–338 (06 2010). https://doi.org/10.1007/s11263-009-0275-4
7. Fookes, C., Lin, F., Chandran, V., Sridharan, S.: Evaluation of image resolution and super-resolution on face recognition performance. Journal of Visual Communication and Image Representation **23**(1), 75–93 (2012)
8. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
10. Guo, J., Lou, H., Chen, H., Liu, H., Gu, J.J., Bi, L., Duan, X.: A new detection algorithm for alien intrusion on highway. Scientific Reports **13** (2023), https://api.semanticscholar.org/CorpusID:259308664
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
12. Hu, P., Ramanan, D.: Finding tiny faces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 951–959 (2017)
13. Jocher, G.: YOLOv5 by Ultralytics (May 2020). https://doi.org/10.5281/zenodo.3908559, https://github.com/ultralytics/yolov5
14. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (Jan 2023), https://github.com/ultralytics/ultralytics
15. Leng, J., Liu, Y., Gao, X., Wang, Z.: Crnet: Context-guided reasoning network for detecting hard objects. IEEE Transactions on Multimedia **26**, 3765–3777 (2024)
16. Leng, J., Mo, M., Zhou, Y., Gao, C., Li, W., Gao, X.: Pareto refocusing for drone-view object detection. IEEE Transactions on Circuits and Systems for Video Technology **33**(3), 1320–1334 (2023)
17. 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. pp. 1222–1230 (2017)

18. Liang, W., Sun, Y.: Elcnn: A deep neural network for small object defect detection of magnetic tile. IEEE Transactions on Instrumentation and Measurement **71**, 1–10 (2022)
19. Lim, J.S., Astrid, M., Yoon, H.J., Lee, S.I.: Small object detection using context and attention. In: 2021 international Conference on Artificial intelligence in information and Communication (ICAIIC). pp. 181–186. IEEE (2021)
20. 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. pp. 2117–2125 (2017)
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
22. 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: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
23. Liu, G., Han, J., Rong, W.: Feedback-driven loss function for small object detection. Image and Vision Computing **111**, 104197 (2021)
24. 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. pp. 8759–8768 (2018)
25. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016)
26. Liu, Z., Gao, G., Sun, L., Fang, L.: Ipg-net: Image pyramid guidance network for small object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 4422–4430 (2020)
27. Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., Chen, H.: Dc-yolov8: Small-size object detection algorithm based on camera sensor. Electronics **12**(10) (2023)
28. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al.: Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999 (2018)
29. 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. pp. 779–788 (2016)
30. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
31. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
32. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
34. 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. pp. 7464–7475 (2023)

35. Wang, J., Yang, W., Guo, H., Zhang, R., Xia, G.S.: Tiny object detection in aerial images. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 3791–3798 (2021). https://doi.org/10.1109/ICPR48806.2021.9413340
36. Wang, J., Liu, W., Zhang, W., Liu, B.: Lv-yolov5: A light-weight object detector of vit on drone-captured scenarios. In: 2022 16th IEEE International Conference on Signal Processing (ICSP). vol. 1, pp. 178–183 (2022)
37. Wang, X., He, N., Hong, C., Sun, F., Han, W., Wang, Q.: Yolo-erf: lightweight object detector for uav aerial images. Multimedia Systems **29**(6), 3329–3339 (Dec 2023)
38. Yang, Y., Gao, X., Wang, Y., Song, S.: Vamyolox: An accurate and efficient object detection algorithm based on visual attention mechanism for uav optical sensors. IEEE Sensors Journal **23**(11), 11139–11155 (2023)
39. Yu, X., Gong, Y., Jiang, N., Ye, Q., Han, Z.: Scale match for tiny person detection. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 1257–1265 (2020)
40. Zeng, N., Wu, P., Wang, Z., Li, H., Liu, W., Liu, X.: A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. IEEE Transactions on Instrumentation and Measurement **71**, 1–14 (2022)
41. Zhang, H., Wang, Y., Dayoub, F., Sunderhauf, N.: Varifocalnet: An iou-aware dense object detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8514–8523 (2021)