Real-time Detection of Tiny Objects Based on a Weighted Bi-directional FPN

Yaxuan Hu[1], Yuehong Dai[1](✉), and Zhongxiang Wang[2]

[1] University of Electronic Science and Technology of China, Chengdu, China
daiyh@uestc.edu.cn

[2] ShenZhen East-Win Technology Co., LTD., Shenzhen, China

Abstract. Tiny object detection is an important and challenging object detection subfield. However, many of its numerous applications (e.g., human tracking and marine rescue) have tight detection time constraints. Namely, two-stage object detectors are too slow to fulfill the real-time detection needs, whereas one-stage object detectors have an insufficient detection accuracy. Consequently, enhancing the detection accuracy of one-stage object detectors has become an essential aspect of real-time tiny objects detection. This work presents a novel model for real-time tiny objects detection based on a one-stage object detector YOLOv5. The proposed YOLO-P4 model contains a module for detecting tiny objects and a new output prediction branch. Next, a weighted bi-directional feature pyramid network (BiFPN) is introduced in YOLO-P4, yielding an improved model named YOLO-BiP4 that enhances the YOLO-P4 feature input branches. The proposed models were tested on the TinyPerson dataset, demonstrating that the YOLO-BiP4 model outperforms the original model in detecting tiny objects. The model satisfies the real-time detection needs while obtaining the highest accuracy compared to existing one-stage object detectors.

Keywords: Tiny object detection · Bi-directional feature pyramid network · Real-time detection · YOLOv5

1 Introduction

Object detection is a critical component of computer vision research that focuses on the recognition and localization of objects in images. Detecting tiny objects is an essential but challenging object detection area with many applications, including surveillance, tracking [17], aided driving [18], remote sensing image analysis [10], and marine rescue [16]. Tiny object detection deals with very small objects in images of extremely low resolution. An increase in resolution yields blurred images, making it difficult to extract enough features for learning. These characteristics pose a significant challenge for tiny object detection.

Current object detection algorithms can be broadly classified into one-stage and two-stage object detection algorithms. One-stage object detection algorithms are fast and enable very rapid detection while ensuring a specific degree of

B. Þór Jónsson et al. (Eds.): MMM 2022, LNCS 13141, pp. 3–14, 2022.
https://doi.org/10.1007/978-3-030-98358-1_1

detection accuracy. These algorithms include SSD [2], RetinaNet [1], and YOLO series (YOLOv3 [3], YOLOv4 [4], YOLOv5 [5]). In contrast, two-stage object detection algorithms are extremely accurate. Thus, algorithms such as Faster-RCNN [6] and Mask R-CNN [7] have higher detection accuracy than one-stage detection algorithms, but their detection speed is typically worse. Many application scenarios, such as personnel tracking and sea rescue, pose strict requirements on the model's detection time. The two-stage object detection algorithm cannot meet the real-time detection requirements, and the one-stage object detection algorithm's detection accuracy is insufficient. Therefore, the one-stage object detection algorithms' detection accuracy improvement has become a significant concern in the tiny object detection domain.

This paper studies the accuracy problems in detecting tiny objects and, as a result, proposes an improved approach and advances the parameter settings in the training process. The proposed algorithm, named YOLO-P4, enhances the tiny object detection accuracy. YOLO-P4 is based on the YOLOv5 one-stage object detection algorithm but includes both a new prediction branch and a module specialized for detecting tiny objects, thus substantially improving the model's effectiveness.

However, it is important to note that YOLO-P4 improves the accuracy at the expense of detection speed. Thus, YOLO-BiP4 is also proposed to improve detection speed. The algorithm builds on a new feature fusion structure called weighted bi-directional feature pyramid network (BiFPN) [2], but the structure was modified for the YOLO-P4 feature layer. As a result, the YOLO-BiP4 algorithm reduces the number of model parameters while maintaining detection accuracy and improving the detection speed to satisfy the real-time detection requirements.

2 Related Work

The proposed algorithm for tiny object detection is based on BiFPN, a multiscale feature fusion approach. This section briefly introduces the two parts of tiny object detection and multiscale feature fusion.

2.1 Tiny Object Detection

Many research efforts were directed at handling the lack of information and large size differences between the object and the background to improve the accuracy and speed of tiny object detection. For example, Kis et al. [8] used data enhancement to increase the number of tiny objects via oversampling images containing the tiny objects and copy-pasting them. Gong et al. [9] focused on the fact that image feature fusion is affected by the dataset scale distribution, introducing a fusion factor α to enhance the fusion effect between feature pyramid network's (FPN) layers. Liu et al. [10] proposed UAV-YOLO, which modified the ResNet structure by changing the number of layers and the connection of modules. As a result, the network's receptive field was enlarged, and its semantic feature extraction capability improved. Jiang et al. [11] proposed a simple and effective scale matching method to achieve a favorable tiny object representation by aligning the tiny objects' scales in datasets for pre-training and learning. The listed methods approach the tiny object detection from different angles striving to improve its performance.

2.2 Multiscale Feature Fusion

Feature fusion combines features from different image scales. The vast majority of the current target detection algorithms use a high feature layer after repeated downsampling for target classification and regression, significantly affecting the tiny object detection. Note that since the tiny object size is very small, the available features are limited. As the network deepens, it is difficult to preserve enough features at high sampling rates, and their detailed information may be completely lost. This problem is tackled by fusing shallow and deep image features to enhance the tiny object feature extraction. Shallow features have higher resolution and contain more detailed location information. However, these features undergo fewer convolutions, making them noisier and less semantic. In contrast, deeper features contain more robust semantic information but have lower resolution and poorer perception of details. Thus, an efficient fusion of shallow and deep features is vital to ensure tiny object detection accuracy.

FPN [12] is a classical image feature fusion network. It has a top-down network structure with lateral connections and constructs feature maps of various sizes and high-level semantic information. FPN extracts features for images of each scale and produces multi-scale feature representations with strong semantic information for every level's feature map. Nevertheless, it significantly increases the network inference time and occupies considerable memory, seriously affecting the model's operational efficiency.

To overcome the discussed FPN's drawbacks, scholars have proposed various FPN structures. Liu et al. [13] developed a Path Aggregation Network (PANet), which adds a bottom-up path aggregation network to FPN to fully integrate the features from different feature layers and greatly improve the detection. The YOLOv5 model's neck utilizes FPN and PANet. Kim et al. [14] considered the features' contextual information and proposed a parallel FPN. A multi-scale context aggregation module serves to resize the parallel feature mappings to the same size while aggregating their contextual information to obtain each level of the final feature pyramid. As a result, this action reduces the performance differences between features at each level. BiFPN is another FPN improvement. In contrast to treating features of different scales equally, BiFPN introduces a weighting mechanism to balance the feature information. While considering the distinct contributions of features at different scales, weights are assigned to each branch involved in feature fusion to perform adaptive learning. BiFPN deletes nodes with only one input or output edge and adds an extra edge between the output and output nodes at the same level. Thus, more features can be fused without increasing consumption (see Fig. 1).

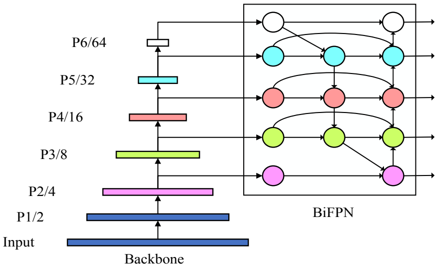


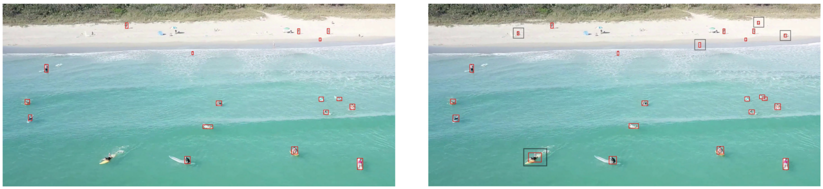
Fig. 1. The structure of BiFPN and backbone

3 The Proposed Model

3.1 YOLO-P4

According to the COCO dataset [15] definition, a small object is an object of fewer than 32 × 32 pixels. Several practical applications require performing remote acquisition of images, which typically have a large resolution, but the object to be detected is small. For example, in the Tiny-Person [16] dataset, the training contains a total of 21599 labels, among which 6872 labels are less than 3 pixels in length or width.

Within this work, the object smaller than 20 pixels is considered a tiny target, and they directly lead to the model's low detection efficiency. The detection results of using the YOLOv5s model to detect an image in Tiny-Person are shown in Fig. 2(a). One can note from the figure that there are several undetected targets. The reasons stem from the YOLOv5s structure (see Fig. 3). In the model's backbone part, four downsampling operations are performed first, and then the P3, P4, and P5 feature layers are fused in the neck part for target prediction. The input image is sized 640 × 640, the detection layer size of P5 is 20 × 20, which serves to detect targets of size 32 × 32 or larger. The P4 feature layer corresponds to a detection layer of size 40 × 40 and is used to detect targets of size 16 × 16 or larger. Finally, the P3 feature layer has 80 × 80 size and can detect targets of size 8 × 8 or larger.



(a) YOLOv5s detection results

(b) YOLOs-P4 detection results

Fig. 2. Comparison of detection results

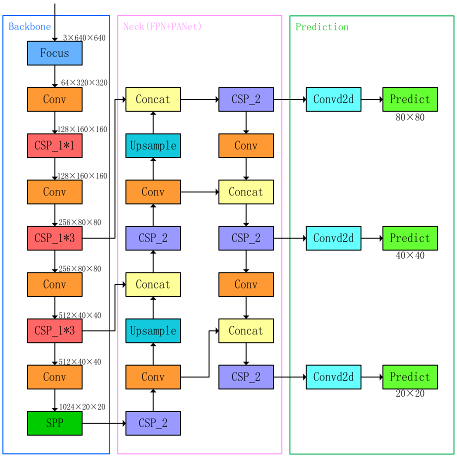


Fig. 3. The structure of YOLOv5s

The current three detection layers typically serve for detecting objects with a size above 8 × 8. When the object size is smaller than 8 × 8, the detection accuracy drops dramatically. Nevertheless, the size of tiny objects is generally smaller than 8 × 8. Further, as noted previously, downsampling drastically reduces the feature information, preventing one from extracting sufficient information. An analysis revealed that the input image size could be upgraded to 960 × 960 so that the detection layer size of the P3 feature layer is 120 × 120. Then, the tiny object size is expanded 2.25 times, and more valuable features can be extracted. The images of P2, P3, P4, and P5 feature layers with input sizes of 640 and 960 are visualized in Fig. 4. One can note that the P3 feature layer of the 960 × 960 image contains more effective information than the 640 × 640 image.

However, although the number of effective features contained in the P3 feature layer is boosted by increasing the input image size, the small scale of a tiny object renders effective features retained after two downsamplings too few to extract sufficient information. Figure 3 shows that the features in P3 provide insufficient information for tiny object feature extraction. Thus, using the information in the P2 feature layer is necessary, ensuring it participates in the feature fusion process.

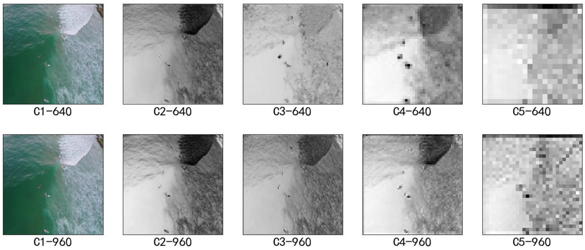


Fig. 4. Feature layer display for different input sizes

Therefore, to involve P2 in feature fusion, the original model's structure was extended by adding a module for detecting tiny objects and a new output detection branch (Predict-Tiny). The improved model is called YOLO-P4, and its structure is depicted in Fig. 5.

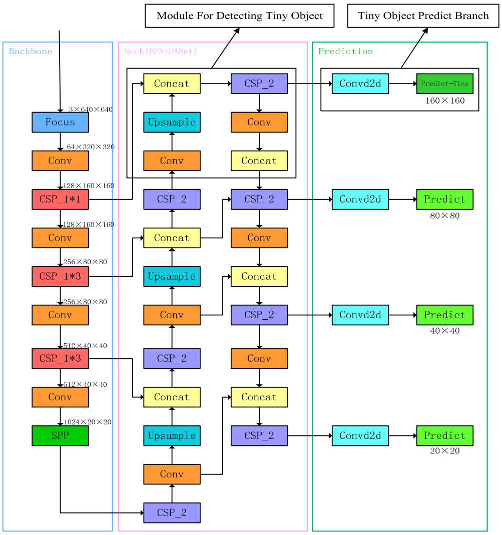


Fig. 5. The structure of YOLOv5s-P4

Compared with the original model in Fig. 3, one can note that YOLO-P4 adds another upsampling before the tiny object detection layers and after the original model is upsampled twice to fuse P2 with tiny objects' rich features. Further, it adds a new output detection branch, which greatly improves the model's detection of tiny objects.

Table 1. Number of models' parameters

Model	YOLOv5x	YOLOv5l	YOLOv5m	YOLOv5s
Original	88.40M	47.37M	21.47M	7.25M
P4	96.48M	51.72M	23.46M	7.93M
BiP4	**78.26M**	**44.16M**	**21.51M**	8.10M

3.2 YOLO-BiP4

Recall that the YOLOv5 improvement for tiny object detection increases the inference time loss while enhancing the model's detection. Namely, adding an upsampling and a new detection layer introduces many parameters, leading to the model inference time loss. In FPN-PANet, the number of parameters passing through the upsampling layer doubles with each additional upsampling layer. Table 1 compares the number of parameters in the original and the improved YOLO-P4 models. The table shows that there are more parameters in YOLO-P4 than in the original model. The parameters increase the corresponding inference time, which is unsuitable for the task at hand. Therefore, the YOLO-BiP4 algorithm is proposed to ensure sufficient detection accuracy without significantly increasing the inference time.

The analysis reveals that controlling the number of parameters reduces the inference time loss. YOLO-P4's neck part has a structure of FPN+PANet. Not to increase the inference time significantly, BiFPN is introduced.

In contrast to other feature fusion methods, BiFPN introduces a weighting mechanism that enables different features treatment. Each branch involved in feature fusion is assigned a weight based on their contributions at different scales during feature fusion. Then, adaptive learning is employed during model training to ensure the model's detection accuracy. Figure 1 presents the BiFPN structure, which fuses five feature layers in turn. When performing fusion, the weights are set in the following ways. Let P_i denote the feature at layer i, and w_i is that feature's fusion weight.

1) Unbounded fusion: Addition is performed directly on the fusion branches. It is equivalent to the one where each fusion branch's weight is equal to one (i.e., $w_i = 1, \forall$i).

2) Softmax-based fusion: the output follows the formula

$$Output = \sum_{i=1}^{4} \frac{exp(w_i))}{\sum_{j=1}^{4} exp(w_j))} \cdot P_i \tag{1}$$

3) Fast normalized fusion: The ReLU function is introduced before the calculation to ensure that weight is greater than zero. $\epsilon = 0.0001$ is a small value to avoid numerical instability. The output equation is

$$Output = \sum_{i=1}^{4} \frac{w_i}{\epsilon + \sum_{j=1}^{4} w_j} \cdot P_i \tag{2}$$

The improved BiFPN is introduced into the YOLO-P4 model, yielding the YOLO-BiP4 architecture. Specifically, one of the BiFPN's input feature layers is removed, changing the number of feature layers at the input side from five to four (namely, P2, P3, P4, and P5). The BiP4 structure is built using both top-down and bottom-up rules, and the fast-normalized fusion is used to set the weights. Figure 6 depicts the YOLO-BiP4 structure.

Figure 6 demonstrates that YOLO-BiP4 removes the vertices with only one input or output edge, thus reducing the number of parameters by a certain amount. Such a design can effectively improve the model's detection speed.

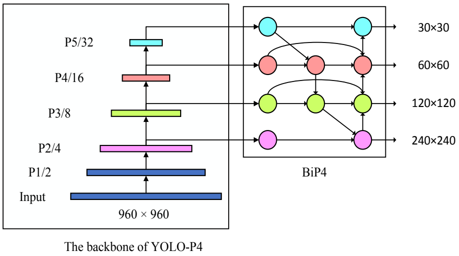


Fig. 6. The structure of YOLO-BiP4

4 Experiments

4.1 Evaluation Methods and Datasets

In the object detection field, the model's detection effectiveness is commonly measured using the *mean average precision* (mAP). Furthermore, in models requiring real-time detection, the number of parameters and the model's detection speed are also crucial. Typically, a model is said to satisfy the real-time detection requirement when its detection speed is greater than 30 *frames per second* (FPS). Therefore, this work studies these three aspects in a multi-dimensional analysis of the model performance.

Tiny-Person is a dataset for tiny object detection collected from high-quality video and web images. It contains 72,651 annotations of human targets with low resolution in visual effects. In total, there are 1610 images, including 794 images in the training set and 816 images in the test set. The average absolute scale of all objects in the Tiny-Person dataset is 18 pixels, while the average size is less than 5 × 5. More than 30% of the objects span less than 3 pixels in height or width, all of which can be called tiny objects.

4.2 Experimental Environment and Parameter Description

The hardware environment used in the experiments reported herein consisted of a CoreTM i7-7700HQCPU@2.80 GHz, GeForce GTX1070 graphics card for training, and RTX3090 graphics card for testing. The main software used was Pycharm 2020.2. The configured virtual environment was based on Python 3.9, and the utilized deep learning framework was Pytorch 1.8.0.

A series of YOLOv5 models were studied, including the original models (i.e., YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x), the model with improved input parameters (i.e., YOLOv5-960), and the models with improved network structure (i.e., YOLOv5-P4 and YOLOv5-BiP4). The trained models result from 200 training epochs on the Tiny-Person dataset using the Adam optimizer with a learning rate set to 0.001.

Table 2. Detection accuracy for different test sizes when the training size is 640. The results reported as mAP (%)

TIS	YOLOv5x	YOLOv5l	YOLOv5m	YOLOv5s
640	27.70	27.63	27.47	26.97
960	35.94	35.26	34.08	33.15
1408	38.77	38.18	36.71	**36.64**
1536	**38.86**	**38.68**	**36.82**	36.49

Table 3. Detection speed for different test sizes when the training size is 640. The results reported as FPS

TIS	YOLOv5x	YOLOv5l	YOLOv5m	YOLOv5s
640	110.51	198.89	328.08	490.83
960	45.43	89.09	143.55	333.72
1408	23.26	45.32	77.58	181.18
1536	15.43	37.52	61.16	126.38

4.3 Comparison of Improved Detection Results

The experiments aimed at improving the input parameters were performed first. The input size ranged from 640 to 960, and the mAP and detection speed were compared for each YOLOv5 model over 200 epochs. The experiments show that increasing the input images' size during testing can effectively improve the models' detection. The experiments comparing the detection speed and accuracy for both the training and test input image sizes are shown in Tables 2, 3, 4 and 5. Here, TIS stands for Test Image Size.

As shown in the table, increasing the input image size enhances the detection accuracy for both training and testing but hampers the detection speed. The model whose training input was of size 960 improves its detection over the course of the training, but its detection speed decreases. The results demonstrate that the detection accuracy and speed are optimal when the input size equals 1408 for the model test. As a result, the following experiments' input parameters were based on a training input sized 960 and a testing input sized 1408.

YOLO-P4 adds a module that fuses the P2 feature layer in the original model to improve the extraction of effective tiny object features. Further, a new output prediction branch called Predict-Tiny that is dedicated to tiny object detection is introduced. Tables 6 and 7 show the YOLO-P4's detection accuracy and speed when the training size equals 960, and the testing size is 1408.

Experiments show that adding the fusion module and the output prediction branch significantly improves the YOLO-P4's detection accuracy but decreases the detection speed in turn. Nevertheless, except for the YOLOx-P4 model, the proposed models can all meet the real-time detection requirement. Compared to the model before the improvement (i.e., YOLOv5), the detection accuracy has

Table 4. Detection accuracy for different test sizes when the training size is 960. The results reported as mAP (%)

TIS	YOLOv5x	YOLOv5l	YOLOv5m	YOLOv5s
640	29.58	29.52	29.30	26.64
960	37.88	37.75	37.37	35.34
1408	**42.59**	**41.95**	41.75	40.89
1536	42.32	41.93	**42.27**	**41.08**

Table 5. Detection speed for different test sizes when the training size is 960. The results reported as FPS

TIS	YOLOv5x	YOLOv5l	YOLOv5m	YOLOv5s
640	108.28	207.18	335.43	497.41
960	54.31	90.32	143.76	323.87
1408	24.34	46.75	84.96	179.62
1536	15.23	37.83	68.38	155.91

Table 6. Model's detection accuracy comparison. The results reported as mAP (%)

Model	YOLOv5x	YOLOv5l	YOLOv5m	YOLOv5s
Original	42.59	41.95	41.75	40.89
P4	**45.90**	**45.69**	**44.79**	42.77
BiP4	45.66	44.61	44.29	**43.26**

Table 7. Model's detection speed comparison. The results reported as FPS

Model	YOLOv5x	YOLOv5l	YOLOv5m	YOLOv5s
Original	24.34	46.75	84.96	179.62
P4	18.36	36.92	45.91	91.51
BiP4	21.08	40.45	62.23	109.78

improved by 7.8%, 8.9%, 7.3%, and 4.6%, respectively. Figure 2(a) and 2(b) show the detection results before and after the improvement, where the differences are highlighted using the black boxes.

Table 1 compares the number of parameters in YOLO-P4 to those in the original YOLOv5 model. Note that detection speed decreases as the number of parameters increases.

Next, the YOLO-P4 model structure was improved by introducing the BiFPN structure to reduce the number of parameters and enhance the detection speed. The BiFPN structure is improved by removing one input, thus reducing the number of model parameters and significantly increasing the model's detection

Table 8. Comparison of different models detection results in Tiny-Person dataset

Model	mAP (%)	FPS
YOLOx-BiP4 (Ours)	45.66	21.08
YOLOl-BiP4 (Ours)	**44.61**	**40.45**
YOLOm-BiP4 (Ours)	44.29	62.23
YOLOv5l	41.95	46.75
YOLOv4	32.83	44.87
NAS-FPN	37.75	18.06
RetinaNet-101	33.53	5.20
SSD-513	31.2	8.10
Faster R-CNN	47.35	5.80

speed. The improved model's accuracy and speed are shown in Tables 6 and 7, respectively, whereas the corresponding number of parameters is given in Table 1.

Tables 6 and 7 show that, compared to YOLO-P4, the improved YOLO-BiP4 model reduced the number of parameters while increasing the detection speed by 14.8%, 9.6%, 35.5%, and 20.1%, respectively. Furthermore, the improved model maintained the YOLO-P4's detection accuracy. Compared to YOLOv5-960, the YOLO-BiP4's detection accuracy is 7.2%, 6.3%, 6.1%, and 5.8% higher, respectively.

Finally, Table 8 shows the YOLO-BiP4's detection results and contrasts them to other one-stage and two-stage models' performances on the Tiny-Person dataset. Table 8 demonstrates that the proposed YOLO-BiP4 model achieves the highest accuracy of a one-stage model for detecting tiny objects while satisfying the real-time detection requirements.

5 Conclusions and Future Works

Building on the YOLOv5 algorithm, this paper first proposed YOLO-P4, an improved algorithm for tiny object detection. A module dedicated to tiny object detection was added to fuse the P2 feature layer with sufficient object features, and an output prediction branch was added to predict tiny objects. Experiments demonstrate that such modifications improve the YOLO-P4's average accuracy in detecting tiny objects by 7.2%. Then, YOLO-BiP4 is proposed to reduce the number of model parameters and improve the detection speed. YOLO-BiP4 is based on YOLO-P4 but introduces the BiFPN structure to improve the YOLO-P4's feature layer. This modification results in the average reduction of the number of parameters by 10.9%, while the detection speed increased by 19.9% on average. The proposed model achieves the highest accuracy of the one-stage tiny object detectors while considering the real-time detection requirement. However, the detection speed is still insufficient despite the tiny object detection accuracy improvement. Thus, simultaneous improvement of the detection accuracy and speed remains a critical direction for future research.

References

1. Lin, T.Y., Goyal, P., Girshick, R., et al.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
2. 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, pp. 10781–10790 (2020)
3. Redmon, J., Farhadi, A.: Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
4. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
5. Glenn, J.: ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements. https://github.com/ultralytics/yolov5 (2020)
6. Ren, S., He, K., Girshick, R., et al.: Faster r-cnn: towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. **28**, 91–99 (2015)
7. He, K., Gkioxari, G., Dollár, P., et al.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
8. Kisantal, M., Wojna, Z., Murawski, J., et al.: Augmentation for small object detection. arXiv preprint arXiv:1902.07296 (2019)
9. Gong, Y., Yu, X., Ding, Y., et al.: Effective fusion factor in FPN for tiny object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1160–1168 (2021)
10. Liu, M., Wang, X., Zhou, A., et al.: UAV-YOLO: small object detection on unmanned aerial vehicle perspective. Sensors **20**(8), 2238 (2020)
11. Jiang, N., Yu, X., Peng, X., et al.: SM+: refined scale match for tiny person detection. In: ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1815–1819. IEEE (2021)
12. Lin, T.Y., Dollár, P., Girshick, R., et al.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
13. Liu, S., Qi, L., Qin, H., et al.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)
14. Kim, S.W., Kook, H.K., Sun, J.Y., et al.: Parallel feature pyramid network for object detection. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 234–250 (2018)
15. 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: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
16. Yu, X., Gong, Y., Jiang, N., et al.: Scale match for tiny person detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1257–1265 (2020)
17. Chen, L., Ai, H., Zhuang, Z., et al.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2018)
18. Chen, J., Bai, T.: SAANet: spatial adaptive alignment network for object detection in automatic driving. Image Vision Comput. **94**, 103873 (2020)