YOLOv5-LW: Lightweight UAV Object Detection Algorithm Based on YOLOv5

He Xiao[1,2(✉)], Kai Zhao[1], Xiaomei Xie[1], Peilong Song[1], Siwen Dong[1], and Jiahui Yang[1]

[1] School of Software Engineering, Jiangxi University of Science and Technology, Nanchang 330013, Jiangxi, People's Republic of China
xiaohe804@gmail.com, {kewitt,mavia}@jxust.edu.cn, adsw@mail.jxust.edu.cn
[2] Nanchang Key Laboratory of Virtual Digital Factory and Cultural Communications, Nanchang 330013, People's Republic of China

Abstract. UAV object detection task is a highly popular computer vision task, where algorithms can be deployed on unmanned aerial vehicles (UAVs) for real-time object detection. However, YOLOv5's performance for UAV object detection is not entirely satisfactory due to the small size of the detected objects and the problem of occlusion. To address these two issues in the YOLOv5 algorithm, we propose the YOLOv5-LW algorithm model. Building upon YOLOv5, we replace the FPN-PAN network structure with the FPN-PANS structure. This modification helps mitigate the issue of feature disappearance for small objects during the training process while reducing the model parameters and computational complexity. Additionally, within the FPN-PANS structure, we employ a multistage feature fusion approach instead of the original feature fusion module. This approach effectively corrects the erroneous information generated during the upsampling stage for certain objects. Finally, we replace the SPPF module with the SPPF-W module to further increase the receptive field while maintaining almost unchanged parameters. We conducted multiple experiments and demonstrate that YOLOv5-LW performs exceptionally well in lightweight small object detection tasks using the VisDrone dataset. Compared to YOLOv5, YOLOv5-LW achieves a 4.7% improvement in mean average precision (mAP), reduces the model size by 40%, and decreases the parameters by 40%.

Keywords: Object detection · UAV object detection · Lightweight model · Small object detection

This work was supported by Jiangxi Province Office of Education.

C. Wu et al. (Eds.): MONAMI 2023, LNICST 559, pp. 16–26, 2024.
https://doi.org/10.1007/978-3-031-55471-1_2

1 Introduction

UAV object detection tasks have been widely applied in various scenarios, such as plant protection [1,2], wildlife conservation [3,4], and EU regulation monitoring [5,6]. However, our focus lies more on the development of lightweight real-time object detection.

In recent years, significant progress has been made in object detection tasks using deep convolutional neural networks [7–10]. Benchmark datasets like MS COCO [11] and PASCAL VOC [12] have played a crucial role in promoting the advancement of object detection applications. However, most previous deep convolutional neural network works were designed for natural scene images, and directly applying these models to handle object detection tasks in UAV-captured scenes presents challenges. Some examples in Fig. 1 illustrate this point. Firstly, the scale of objects varies significantly due to different flight altitudes, causing drastic changes in object sizes. Secondly, images captured by UAVs contain dense objects, leading to object occlusion. Lastly, UAV-captured images often contain a large amount of confusing background due to coverage. These three issues make object detection in UAV-captured images highly challenging.

The YOLO [13–17] series of one-stage detectors plays a crucial role in object detection. In this paper, we propose a new lightweight detection model called YOLOv5-LW based on YOLOv5. The proposed model addresses the problem of feature loss for small objects during feature fusion and the generation of erroneous information due to the disappearance of certain object features during the upsampling process. Additionally, we increase the receptive field of the network and reduce the algorithm parameters and model size. Compared to YOLOv5, our algorithm achieves higher accuracy with fewer parameters and model size (as shown in Fig. 1).

Our contributions are as follows:

- We propose the FPN-PANS network structure, which not only reduces the model parameters but also enhances detection accuracy with multiple detection heads.
- We introduce a multistage feature fusion module that effectively reduces information loss during convolution and upsampling processes.
- We propose the SPPF-W module, which increases the field of view compared to the original SPPF module while keeping the parameters unchanged.
- We conduct extensive and effective experiments to demonstrate the superiority of our algorithm compared to related algorithms.

2 Related Work

2.1 Object Detection Algorithm

Object detection algorithm is an important research direction in the field of computer vision. With the development of deep learning technology, object detection

Fig. 1. As shown in the figure, due to the far-distance top-down shooting under different lighting conditions in the VisDrone [5] dataset, it differs from other datasets in terms of shooting angles. Moreover, the objects are small and densely packed, significantly increasing the difficulty of detection.

algorithms have also made significant progress. This article will introduce the development of object detection algorithms and the classification of object detection algorithms.

The development of object detection algorithms can be divided into two stages. The first stage is traditional object detection algorithms, which mainly include template matching, edge detection, color features, and other methods. These methods have limitations in terms of accuracy and robustness. The second stage is the application of deep learning technology, mainly including object detection algorithms based on convolutional neural networks, such as RCNN [18], Fast RCNN [19], Faster RCNN [20], YOLO, etc. These algorithms have greatly improved in terms of accuracy and speed.

The classification of object detection algorithms can be divided into two categories: region-based object detection algorithms and one-stage object detection algorithms. Region-based object detection algorithms mainly generate a series of candidate regions in the image and then classify and regress each candidate region. One-stage object detection algorithms directly classify and regress the entire image. Among them, region-based object detection algorithms include RCNN, Fast RCNN, Faster RCNN, etc.; one-stage object detection algorithms include YOLO, SSD [21], etc.

2.2 Neck

In object detection, the role of the Neck layer is to fuse features from different layers to improve the model's performance. As the field of object detection continues to evolve, there are now various types of Neck layers commonly used. Here are a few examples:

1. Feature Pyramid Network (FPN) [22]: FPN is a widely-used feature pyramid structure that aims to simultaneously capture features at different scales by utilizing both top-down and bottom-up pathways for feature fusion, thereby improving detection accuracy.

2. Path Aggregation Network (PAN) [23]: PAN is another feature pyramid-based Neck structure that focuses on information aggregation and expansion between different feature pyramids, enabling better feature fusion and improving detection accuracy.
3. Neural Architecture Search FPN (NAS-FPN) [24]: NAS-FPN is a feature pyramid structure automatically designed using neural architecture search algorithms, which outperforms traditional FPN structures in terms of performance.
4. Bi-Directional Feature Pyramid Network (BiFPN) [25]: BiFPN is a bi-directional feature pyramid structure that enables feature communication and propagation in both top-down and bottom-up directions, facilitating better information exchange and feature fusion.

In conclusion, with the advancement of deep learning techniques, object detection algorithms have made significant progress. Object detection algorithms can be categorized into region-based and one-stage approaches. These algorithms have wide-ranging applicability in various domains.

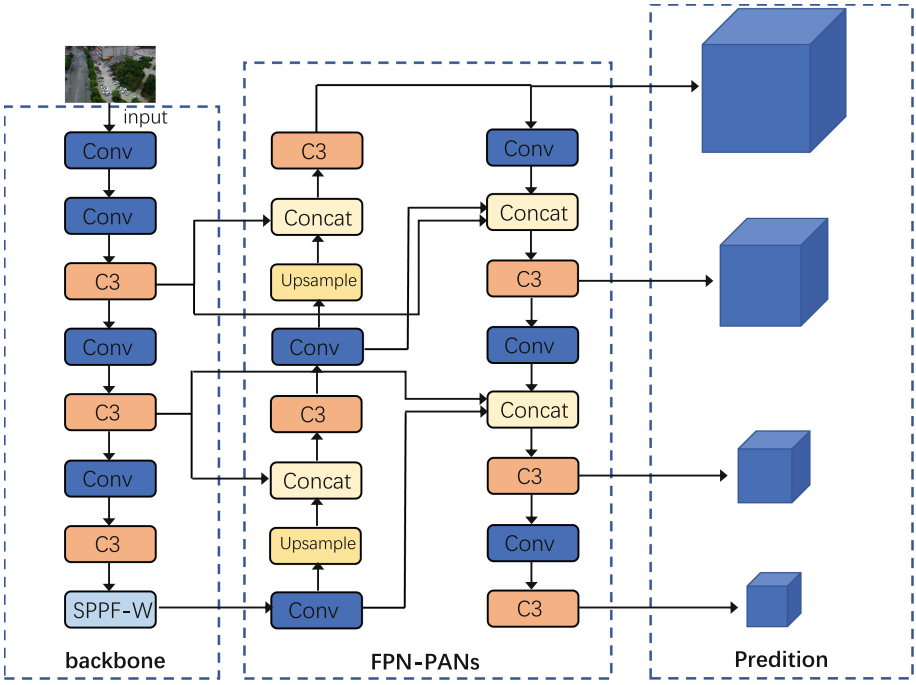

Fig. 2. The network structure of our algorithm consists of three parts: the main network structure, FPN-PANs, and the Head, as shown in the figure above.

3 YOLOv5-LW

3.1 Overview of YOLOv5

For YOLOv5, it includes multiple models such as YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5n, and YOLOv5x. These models only differ in terms of their width and depth, while the overall model structure remains the same. Generally, YOLOv5 uses the architecture of CSPDarknet53 with an SPPF layer as the backbone, PANet as the Neck, and YOLO detection head. In our experiments, we conducted comparisons using models of different sizes, and the results showed significant improvements in our algorithm across multiple models. Please note that the translation might not capture the complete technical details accurately, as it is a complex domain-specific topic.

3.2 YOLOv5-LW

The network architecture of YOLOv5-LW is shown in Fig. 2. We have made some modifications to the YOLOv5 base model to make it more suitable for unmanned aerial vehicle (UAV) object detection. We have also made some adjustments to the parameters of the network structure to prioritize the extraction of shallow features.

FPN-PANS Structure. For the VisDrone dataset, there are only a few large objects present. We have visualized the output of each layer's features as shown in the figure. Through comparison, we have found that in the deep convolution layers, the detection targets and the background are almost indistinguishable, while more useful information is concentrated in the shallow convolution layers. There is limited useful information in the deep convolution layers. Therefore, in the FPN stage, we have decided to abandon the extraction of deep convolution features and focus on the fusion of shallower features. In the PANS stage, however, we perform deeper convolutions and retain the original detection head for large object detection to improve detection accuracy for those large objects. Although there is an increase in computation and parameters in the PANS stage, the overall computational cost and the number of parameters are greatly optimized as we eliminate the significant overhead of deep convolution in the main network stage. The specific network structure is illustrated in the figure.

SPPF-W Module. For SPP [26], RFB [27], SPPF, and SimSPPF, their main purpose is to enhance the perception of the visual field. SPP modules have their origins in spatial pyramid matching (SPM) [28] techniques. The original SPM method divides the feature map into equally sized blocks with dimensions of d*d, where d can take values such as 1, 2, 3, and so on. This division forms a spatial pyramid, and then the bag-of-words features are extracted. In the YOLO series, SPP (implemented in YOLOv3) consists of three parallel max-pooling outputs with sizes of $k \times k$, where $k = 5$, 9, and 13, respectively. SPPF (employed in YOLOv4) consists of three max-pooling outputs with a fixed size of $k \times k$, where $k = 5$. SPPF provides the same field enhancement as SPP but with fewer structural parameters and faster computation. In YOLOv6, the SimSPPF module

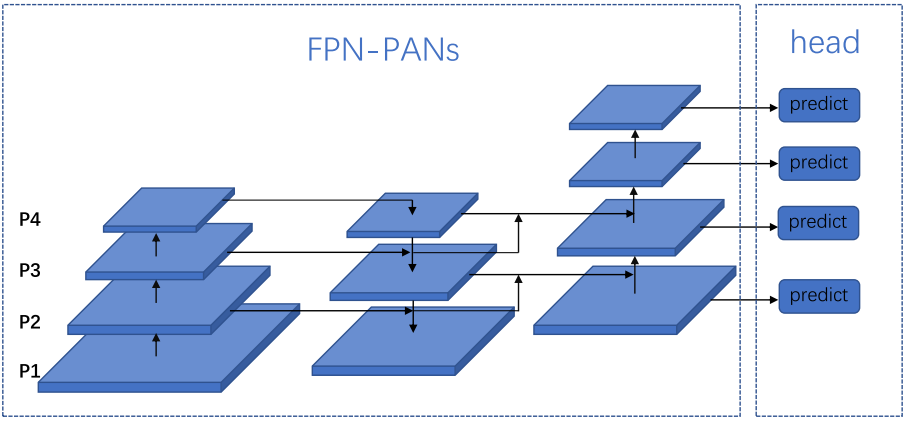

Fig. 3. In the FPN-PANs structure, we primarily focus on integrating shallow-level information and adding a small object detection head. This design enables better performance in unmanned object detection tasks.

simply replaces the SiLU activation function used in SPPF with the ReLU activation function. To further enhance the field of view, we modified the SPPF module in YOLOv6 by setting k to 7 and observed significant improvements in training performance. However, this modification also increased the model's parameter count (Fig. 3).

In this paper, a novel pooling module called SPPF-W is proposed (see Fig. 4). The SPPF-W module is composed of three SPOOL modules, with each SPOOL module consisting of three SP modules. Each SP module comprises a k × k max-pooling operation and a SiLU activation function. For instance, if we assume the input image size is 28 × 28 and apply a 5 × 5 pooling layer with a step size of 1, the resulting feature map size would be $(28 - 5)/1 + 1 = 24$. Similarly, using a 77 pooling layer with a step size of 1, we would obtain a feature map size of $(28 - 7)/1 + 1 = 22$. Finally, when three successive 3 × 3 pooling layers are applied, the feature map sizes would be as follows: $(28 - 3)/1 + 1 = 26$ for the first layer, $(26-3)/1+1 = 24$ for the second layer, and $(24-3)/1+1 = 22$ for the third layer. It is noteworthy that the 7 × 7 pooling layer provides a similar field of view as the combination of three 3 × 3 pooling layers. In terms of parameter count, the three 3 × 3 pooling layers have a total of 3 × (3 × 3) channels, the 5 × 5 pooling layer has 5 × 5 channels, and the 7 × 7 pooling layer has 7 × 7 channels. Therefore, the SPPF-W module achieves a larger perceptual field of view with minimal parameter growth. Specifically, it has approximately 45% fewer parameters compared to using a 7 × 7 pooling layer.

Multi Level Feature Fusion Module (MLF). In the original PAN network architecture, only the relevant features in FPN are fused. However, we found that for small object detection, a significant amount of valuable information is lost during the convolution and upsampling processes. Even with fusion,

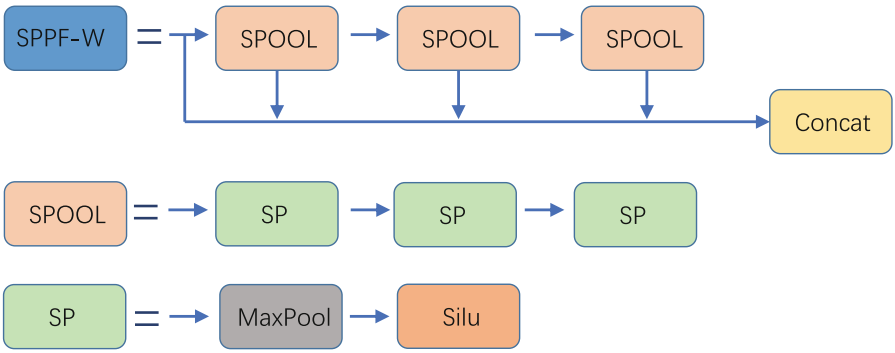

Fig. 4. SPPF-W: We have adopted multiple small pooling layers instead of a larger pooling layer to achieve the purpose of expanding the field of view and reducing parameters.

the fused features are often contaminated with a considerable amount of noise. This is mainly because in deeper convolutions, the detection of the target and background becomes indistinguishable, and upsampling alone is not effective in restoring the features. To mitigate the impact of this noise on the model's accuracy, we propose the PANS network architecture, as shown in Fig. 1. In the convolution process, we incorporate fusion with the features from the main network, allowing a larger amount of valuable information to correct erroneous information and improve the accuracy of the algorithm. Please refer to Fig. 1 for the model structure

4 Experiments

4.1 Experimental Environment

Implementation Details In aerial object detection experiments, we used the pytorch framework and the Ubuntu 22.04 operating system. The model parameters are set as follows: lr = 0.01, epoch = 1000, batch size = 28. All experiments were trained on the GTX3070 GPU.

4.2 Experimental Data

We evaluated our model using the testing challenge and testing development subsets of the Vis-Drone2021 dataset. We reported the mean Average Precision (mAP) across all 10 IoU thresholds ranging from 0.5 to 0.95, as well as the mAP at IoU threshold 0.5 (mAP50). In this chapter, we conducted control experiments, including different model sizes of the same algorithm, and ablation experiments to validate the superiority of our algorithm.

4.3 Comparisons with the State-of-the-Art

In this section, we trained and tested multiple YOLOv series models along with our algorithm model on the VisDrone dataset (with images uniformly resized to 512 × 512). The final results are shown in Table 1, which demonstrates the superiority of our algorithm. From the data in the table, it can be observed that our model outperforms most algorithm models in terms of accuracy and has fewer parameters and a smaller model size. Although the YOLOv7 series algorithm models achieve higher accuracy than our model, with a 0.3% higher mAP0.5:0.95 and a 2.2% higher mAP0.5 for the S-sized model, our model only has around 27% of the parameters and approximately 29% of the model size compared to YOLOv7. In terms of model size, our model is slightly less accurate than YOLOv7, with a 2.7% lower mAP0.5 and a 0.8% lower mAP0.5:0.95. However, our model's parameters are only around 27% of YOLOv7, and the model size is about 36% of YOLOv7. Compared with other algorithms, our algorithm is not only more lightweight but also achieves higher accuracy. In conclusion, our model demonstrates better advancement.

Table 1. During the training process, we use 512 × 512 images for training. The bold font represents the best metrics.

VisDrone				
Methods	mAP0.5	mAP0.5:0.95	parameters	model size
Yolov5s	0.286	0.152	7037095	14.4 MB
Yolov6s	0.327	0.191	–	38.0 MB
Yolov7s	**0.398**	**0.214**	9344734	19.0 MB
Yolov8s	0.354	0.208	11129454	22.5 MB
YOLO-LWs	0.376	0.211	**2499348**	**5.5 MB**
Yolov5n	0.242	0.119	1772695	3.8 MB
Yolov6n	0.275	0.156	–	6.2 MB
Yolov7n	**0.33**	0.165	2353014	5.0 MB
Yolov8n	0.301	**0.172**	3007598	6.2 MB
YOLO-LWn	0.303	0.157	**634340**	**1.8 MB**

4.4 Ablation Studies

In this section, we conducted ablation experiments on the VisDrone test dataset to validate the effectiveness of each proposed module. We used the Yolov5 S model for all experiments. The valuable data obtained from these experiments are presented in Table 2.

Effect of SPPF-W: After improving the FPN to FPNS in the model (using Yolov5 S model as an example), there were no significant changes in computational complexity, parameters, or model size. However, the detection performance of the model improved. The mAP at a IoU threshold of 0.5 increased by 0.8%, and the mAP at a IoU threshold of 0.5:0.95 increased by 0.3%. Overall, compared to SPPF, SPPF-W is a superior choice.

Effect of FPN: After modifying the FPN module, the model parameters decreased from over 7 million to just over 2 million. The number of model layers decreased from 214 to 201, and the model size reduced from 14.4 MB to 5.3 MB. However, there was a significant improvement in mAP at a IoU threshold of 0.5 and mAP at a IoU threshold of 0.5:0.95. The mAP at a IoU threshold of 0.5 increased by 7.4%, and the mAP at a IoU threshold of 0.5:0.95 increased by 5.1%. These results indicate that our FPN+ module is more efficient for UAV object detection.

Effect of Multi-Level Fusion: After adding the multi-level fusion module to our algorithm, there was only a slight increase in model parameters and computational complexity, while the model size remained the same. However, there was a significant improvement in the detection results. The mAP at a IoU threshold of 0.5 increased by 0.8%, and the mAP at a IoU threshold of 0.5:0.95 increased by 0.5%.

Table 2. In the ablation experiments, we validate our proposed modules using the S-sized model, and the results demonstrate the advanced performance of each module.

VisDrone				
Methods	mAP0.5	mAP0.5:0.95	parameters	model size
Yolov5s	0.286	0.152	7037095	14.4 MB
Yolov5s+FPN-PANs	0.363	0.204	2401044	5.3 MB
Yolov5s+FPN-PANs+SPPF-W	0.368	0.206	2401044	5.3 MB
YOLO-LWs	**0.376**	**0.211**	2499348	5.5 MB

From the table, it can be observed that compared to Yolov5, our model is more lightweight, and when applied to UAV detection tasks, it achieves higher accuracy and has greater potential for industrial applications.

References

1. Hird, J.N., et al.: Use of unmanned aerial vehicles for monitoring recovery of forest vegetation on petroleum well sites. Remote Sens. **9**(5), 413 (2017)
2. Shao, Z., Li, C., Li, D., Altan, O., Zhang, L., Ding, L.: An accurate matching method for projecting vector data into surveillance video to monitor and protect cultivated land. ISPRS Int. J. Geo Inf. **9**(7), 448 (2020)

3. Kellenberger, B., Volpi, M., Tuia, D.: Fast animal detection in UAV images using convolutional neural networks. In: 2017 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2017, Fort Worth, TX, USA, 23–28 July 2017, pp. 866–869. IEEE (2017)
4. Kellenberger, B., Marcos, D., Tuia, D.: Detecting mammals in UAV images: best practices to address a substantially imbalanced dataset with deep learning. Remote Sens. Environ. **216**, 139–153 (2018)
5. Audebert, N., Le Saux, B., Lefèvre, S.: Beyond RGB: very high resolution urban remote sensing with multimodal deep networks. ISPRS J. Photogrammetry Remote Sens. **140**, 20–32 (2018)
6. Gu, J., Su, T., Wang, Q., Du, X., Guizani, M.: Multiple moving targets surveillance based on a cooperative network for multi-UAV. IEEE Commun. Mag. **56**(4), 82–89 (2018)
7. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28, pp. 91–99 (2015)
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, pp. 779–788 (2016)
9. Lin, T.-Y., Goyal, P., Girshick, R.B., He, K., Dollar, P.: Focal loss for dense object detection. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017, pp. 2999–3007. IEEE Computer Society (2017)
10. 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)
11. Lin, T.Y., et al.: 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
12. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
13. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7263–7271 (2017)
14. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
15. Alexey, B., Chien-Yao, W., Mark, L.H.-Y.: YOLOv4: optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934 (2020)
16. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: Scaled-YOLOv4: scaling cross stage partial network. In: Computer Vision and Pattern Recognition, pp. 13029–13038 (2021)
17. Junyang, C., et al.: A multiscale lightweight and efficient model based on YOLOv7: applied to citrus orchard. Plants-Basel **11**(23) (2022)
18. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra R-CNN: towards balanced learning for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 821–830 (2019)
19. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 91–99 (2015)

21. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) LNCS. ECCV 2016, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
22. Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2117–2125 (2017)
23. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8759–8768 (2018)
24. Ghiasi, G., Lin, T.-Y., Le, Q.V.: NAS-FPN: learning scalable feature pyramid architecture for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7036–7045 (2019)
25. Tan, M., Pang, R., Le, Q.V.: EfficientDet: scalable and efficient object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
26. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **37**(9), 1904–1916 (2015)
27. Liu, S., Huang, D., et al.: Receptive field block net for accurate and fast object detection. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 385–400 (2018)
28. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 2169–2178. IEEE (2006)
29. Visdrone Team. Visdrone 2020 leaderboard (2020). http://aiskyeye.com/visdrone-2020-leaderboard/