

A small target detector design for aerial scenarios based on multi-cross adaptive fusion mechanism and high-efficiency feature extraction model

Zikai Li¹, Xiangyu Kong^{*2}[0000-0003-3062-1978], Haolin Chen³ and Xu Peng³

¹ South China Academy of Advanced Optoelectronics, South China Normal University, Guangzhou 510006, China, lizk@m.scnu.edu.cn

² Institute of Information, Guangdong Polytechnic of Science and Trade, Guangzhou, 510430, China, kong@m.scnu.edu.cn

³ Midea Corporate Intelligent Manufacturing Research Center, Guangzhou, 528311, China, chenhl190@midea.com

³ Midea Corporate Intelligent Manufacturing Research Center, Guangzhou, 528311, China, xupeng51@midea.com

Abstract. In the rapidly evolving field of drone technology and artificial intelligence, detecting small, densely packed objects under challenging lighting conditions from an unmanned aerial vehicle (UAV) perspective poses significant challenges. This paper introduces MPB-YOLO, an efficient small target detection algorithm built upon the YOLOv8s model, designed to address these issues. By refining the neck structure with a small target detection head and a multi-scale adaptive fusion neck, the model's ability to detect small targets is substantially improved. Deformable convolutions and a spatial coordinate attention mechanism are integrated into the feature extraction module to enhance the network's perceptual capabilities, particularly for densely distributed and overlapping targets. Evaluated on the VisDrone2019 dataset, MPB-YOLO surpasses other state-of-the-art algorithms, with a 26.3% and 28% increase in mAP50 metrics on the test and validation sets, respectively, while reducing parameters by 16.8% compared to YOLOv8s. These results confirm the efficacy of MPB-YOLO in aerial object detection tasks.

Keywords: tiny object detector; complex scenarios; multi-cross feature fusion; remote sensing.

1 Introduction

The integration of unmanned aerial vehicles (UAVs) with artificial intelligence (AI) has emerged as a prominent area of research[1-2]. UAVs offer a combination of maneuverability and the capacity to surmount natural constraints, such as varied terrains and environmental conditions, leading to their advantage in providing expansive, efficient, and cost-effective monitoring[3-5]. Consequently, applying UAVs with AI-driven object detection algorithms denotes a field with considerable market relevance[6].

Nonetheless, the efficacy of prevalent object detection algorithms when applied to UAV-sourced data has been suboptimal. This shortfall primarily arises from the incompatibility of standard vision task datasets with aerial imagery, resulting in a lack of algorithmic focus on aerial detection tasks[7-9]. Conventional datasets used for benchmarking, like MS-COCO and the VOC series[10-12], are typically curated from a ground-level perspective and possess common characteristics that do not challenge models in the same way aerial imagery does. These characteristics include pronounced object features, substantial target representation in the frame, minimal irrelevant content, favorable lighting, a primary horizontal viewing angle, and a lack of overlapping subjects.

Such datasets reflect everyday visual experiences and are not tailored to train algorithms for the complexities encountered in aerial views, such as intricate object arrangements and vantage points from altitude. This discrepancy suggests a need for specialized datasets to enhance object detection algorithms for aerial applications.

In the context of UAV integration with target detection algorithms, two distinct approaches emerged. The first approach involved the realization of UAV data acquisition via communication transmission, subsequently relaying the data to a local server for the execution of the detection algorithm. The second approach entailed the UAV carrying lightweight detection algorithms onboard. Consequently, there was a dual demand for algorithms: those designed for local deployment, characterized by high precision and substantial computational resources, and lightweight airframe deployment algorithms, which, while less resource-intensive, still maintained a commendable level of accuracy.

In this paper, we made improvements on the lightweight model S based on YOLOv8 and achieved a considerable accuracy improvement with a 16.8% reduction in the number of parameters. Our method effectively improved the model's performance in aerial view and against complex target stacks.

The main contributions of the article include:

1. In order to improve the feature extraction ability when facing the complex occlusion detection tasks, a spatial coordinate attention was proposed to strengthen the sensitive of deformable convolution, which intensify the offset mask behavior in the process of deformable convolution. The deformable convolution reinforced by the attention mechanism was designed to combine with the feature extraction block.
2. A neck structure with multi-cross adaptive fusion mechanism was proposed to enhance the perception ability of the model to the ground target in the high-altitude perspective. The adaptive feature fusion mechanism not only reduces the number of parameters in the model, but also makes full use of the semantic information in each stage. Also, the multi-cross connection structure made full use of spatial features from the lower layers of the backbone network.

Through the ablation experiments, we demonstrated the feasibility and effectiveness of our proposed network optimization designs. Experimental results on the Vis-Drone2019 [33] dataset showed that our designs can intensify the performances of the benchmark model under considerable parameter decline. Additionally, comparative analyses with state-of-the-art detection models and current mainstream models with adjacent parameters demonstrated the superiority of our proposed method.

2 Related work

In the burgeoning field of artificial intelligence, a plethora of neural network-based object detection methods has surfaced. These methods predominantly bifurcate into one-stage and two-stage detection paradigms. One-stage detection, exemplified by the YOLO (You Only Look Once) series [13-18], merges object localization and classification into a singular process, acclaimed for its rapid processing and real-time applicability. This approach, particularly with the advent of YOLOv5 [15], has spurred numerous enhancements aimed at refining detection capabilities and broadening the understanding of algorithmic strategies for diverse tasks.

Conversely, two-stage detection strategies segregate the process into region proposal followed by simultaneous localization and classification. Despite its slower pace compared to one-stage methods, two-stage detection, with technologies like Faster R-CNN [19] and its extension, Mask R-CNN [20], demonstrates superior accuracy by integrating classification, localization, and segmentation tasks into a cohesive network, enabling more precise object detection.

Focusing on densely packed objects, identified by their proximity and quantity as outlined by Goldman et al. [21], presents unique challenges. Wang et al. [22] categorize occlusions encountered in such scenarios as either background-induced or due to crowding, proposing the Repulsion Loss to mitigate these issues by balancing attraction and repulsion among detection predictions and truths. Moreover, Goldman et al. employed the Jaccard Index for evaluating detection quality, introducing a Soft-IoU layer and an EM-Merger unit to refine detection accuracy in crowded scenes.

Addressing the detection of small objects within high-resolution aerial imagery underscores a significant challenge: the dilution of target information on shallow feature maps [23]. Strategies like increasing network input size offer more detail at the cost of computational efficiency. Alternative approaches, such as partitioning images into subgraphs for targeted feature extraction and classification [24], and employing two-level Faster R-CNN models [25] or CPNet for cluster-based region extraction [26], aim to enhance small object detection while mitigating false positives. Li et al. [27] proposed density estimation and image segmentation into subgraphs as another viable solution, illustrating the continuous evolution and adaptation of object detection methodologies to meet the demands of aerial image analysis.

However, when applying these methods to aerial scenarios target detection, the biggest challenge was how to achieve accurate detection in densely distributed objects. The remote sensing images usually had large image size and complex background, and there were a large number of densely distributed small size detection objects. And improving the accuracy of detecting small target objects. In order to achieve this goal, we not only proposed an attention mechanism to enhance the variable convolution and combine it into the feature extraction module; In addition, a neck structure with multi-size adaptive feature fusion was designed to optimize the semantic features, so that the detection head could obtain the feature vector with accurate semantic information, thereby improving the overall detection performance.

3 Proposed designs

As shown in Figure 1, the architecture of our designed MPB-YOLO network model. This network structure was an improvement on the YOLOv8-s model. In the backbone part and the neck part of the network, we designed the AD-C2f block to intensify the feature extraction ability. In the neck part of the network, we introduced a multi-cross adaptive fusion structure to enhance the performance of the benchmark model.

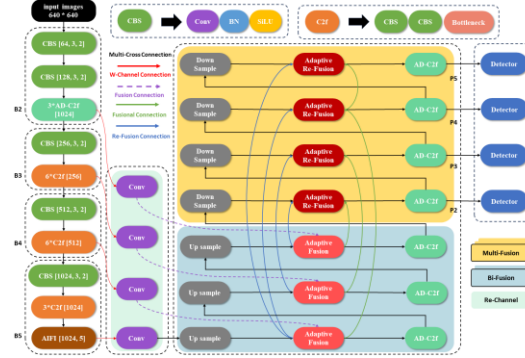


Fig. 1. The proposed structure of MPB-YOLO is depicted in the diagram, where, from left to right, it includes the backbone, neck, and head structures.

3.1 Feature Extraction Improvement

3.1.1. Deformable ConvNets V2.

DCN was first proposed by Dai et al [43]. in 2017. They addressed a critical issue where object distributions in images often deviate from geometric regularity. Traditional square convolutional kernels exhibit poor adaptability to objects with irregular shapes and sizes distributed throughout an image. To tackle this challenge, they introduced a novel convolutional kernel that can dynamically adjust its shape based on the actual context, facilitating the model in extracting features more effectively from the target objects.

Building upon DCN, Zhu [44] et al. discovered that excessive offsets often lead to the convolutional network's receptive field extending beyond the target region, resulting in features unaffected by the actual image content. To address this issue, they proposed two methods: extending the deformable convolution to enhance its modeling capacity and introducing a feature mimicking scheme to guide network learning by constraining the offsets of the deformable convolution, allowing it to focus more precisely on object sizes. In our work, inspired by their contributions, we introduce a Spatial Coordinate Attention designed to reinforce offset adjustments. This attention module enabled the deformable convolution to concentrate more on the scale information of objects, thereby enhancing its capability to extract meaningful features from the target objects.

3.1.2. Spatial Coordinate Attention

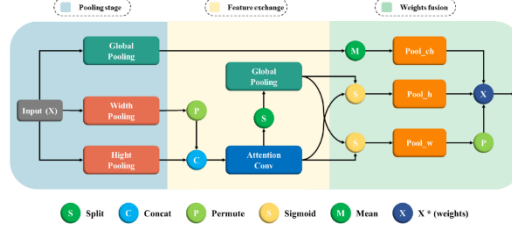


Fig. 2. The constructure of the Spatial Coordinate Attention. (From left to right, the stages include the pooling stage, feature exchange stage, and weighted fusion stage.)

To enhance the feature extraction capability of deformable convolution for small targets, we proposed an offset attention mechanism applied to deformable convolution. The attention mechanism comprehensively captured the scale features of images by integrating horizontal, vertical, and global directional information. As shown in the figure 2, our attention structure consisted of three parts, the pooling stage, feature exchange, and weight fusion.

Pooling stage

The pooling stage was designed with three pooling processes, extracting overall information through global average pooling, capturing feature information separately in horizontal and vertical directions through local pooling.

The equation of the Average pooling on the height dimension:

$$H(x) = \frac{1}{H} \sum_{h=1}^H x_{i,h,j,c} \quad (1)$$

The equation of the Average pooling on the width dimension:

$$W(x) = \frac{1}{W} \sum_{w=1}^W x_{i,j,w,c} \quad (2)$$

The equation of the Global average pooling:

$$G(x) = \frac{1}{h'w'} \sum_{h=1}^h \sum_{w=1}^w x_{i,c,w,n} \quad (3)$$

Feature exchange

After pooling the tensor along the horizontal dimension, feature swapping was conducted to seamlessly combine it with the pooled results along the vertical dimension. The pooled outcomes from both horizontal and vertical dimensions were then concatenated, and vertical features were meticulously captured through a 3×1 convolution operation. Subsequently, a global pooling operation was applied along the channel dimension of the post-convolution tensor. This was followed by a precise calculation of weights through a 1×1 convolution operation.

The concatenation of x_1 , x_2 , along a specified dimension c :

$$C(x_1, x_2, c) = \text{Concat}((x_1, x_2), c) \quad (4)$$

The split operation on the x , along the specified width w and height h from the given channel index i :

$$S(x, i, w, h) = \text{Split}(x, [w, h], i) \quad (5)$$

The permute operation on the x , where the order of dimension is rearranged to $[0, 1, 3, 2]$:

$$P(x) = \text{Permute}(x, [0,1,3,2]) + b \quad (6)$$

Concatenate the features in the height direction with the features in the width direction after permuting. Then, extract features along the height dimension using a 3x1 matrix, followed by obtaining the fused features after extraction:

$$A(x) = w_{(3,1)} \left(C \left(H(x), P(W(x)) \right) \right) + b \quad (7)$$

The features obtained in the previous step (2-4) are globally pooled and split operation is performed along the width and dimension channels to obtain the pooled features in the width and height dimensions:

$$Gw(x), Gh(x) = S(G(A(x), i, w, h)) \quad (8)$$

Weights fusion

Using weights, the features along the horizontal and vertical dimensions were weighted and reset. The weighted features from both horizontal and vertical dimensions, along with the channel features from global pooling, were multiplied to obtain the final output.

The equation of the Sigmoid function:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

Computes the average of the tensor x along the specified dimension c :

$$M(x, c) = \frac{1}{N} \sum_{i=1}^N x_{i,c} \quad (10)$$

Calculate the weights along the width and height direction:

$$qw, qh = \text{Sigmoid}(Gw(x), Gh(x)) \quad (11)$$

Calculate the global weight:

$$qc = \text{sigmoid}(M(G(x))) \quad (12)$$

The height and width features in the Pooling stage are fused with the corresponding weights in the weights fusion stage to obtain the final scale features:

$$Pw, Ph = Gw(x) * qw, Gh(x) * qh \quad (13)$$

The global Pooling features of the Pooling stage are fused with the global weights fusion stage to get the final global features:

$$Pc = G(x) * qc \quad (14)$$

The three features after feature interaction are fused with input x through sigmoid function:

$$X = x * \text{sigmoid}(k), k = [P(Pw), Ph, Pc] \quad (15)$$

This process seamlessly incorporated both global and local information, effectively capturing and integrating features from different dimensions through pooling, convolution, and weighting operations. The specific application of SPA (Spatial Pyramid Attention) involved enhancing the convolutional offset during the computation of DCNv2 (Deformable Convolutional Networks v2) by embedding SPA attention. This reinforcement of convolutional offset was achieved by utilizing the computed offset information, allowing variable convolutions to focus more on targets' scale information.

3.1.3. Replacement Experiment and Result.

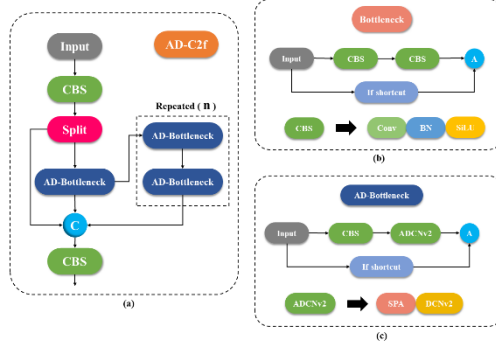


Fig. 3. (a) Structural diagram of AD-C2f; (b) Structural diagram of Bottleneck; (c) Structural diagram of AD-Bottleneck, where ADCNv2 is formed by the fusion of SPA and DCNv2.

To further enhance the feature extraction capability of the network, we integrated the attention mechanism with variable convolutions into the feature extraction module, as illustrated in the Figure 3. Specifically, on the foundation of the original bottleneck, we replaced the second CBS with a variable convolution incorporating attention, introducing a novel design called AD-Bottleneck. Additionally, in the C2f module, we replaced the original Bottleneck with AD-Bottleneck. In order to demonstrated the sensitivity of our design to target scales in small object detection tasks compared to the original feature extraction modules, we conducted experiments to validate the capabilities of our AD-Bottleneck. we performed replacement experiments on the feature extraction modules of B2, B3, B4, B5 in the backbone network, as well as P3, P4, P5 in the neck part of the YOLOv8 network. The evaluation included mAP50 for the four classes with the smallest average size in the dataset and the overall results on the validation set. The experimental data were showed in Table 1:

(All the hyper-parameters and environments are the same, training without using any officially pretrained weight. \surd : C2f, \blacklozenge : D-C2f, \bullet : AD-C2f)

The results showed that after a large number of replacement comparison experiments, we found that after replacing the feature extraction modules of B2, P3, P4 and P5 with AD-C2f, the accuracy of the three types of targets was improved when facing the four types of targets selected, and the mAP50 and mAP95 achieved 0.25% and 0.43% higher than the benchmark model, respectively.

3.2 Neck Structure Improvement

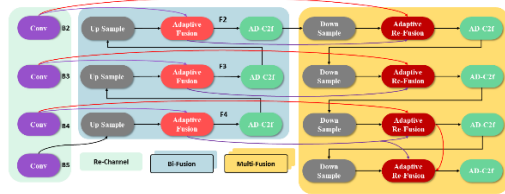


Fig. 4. The structural diagram of MFPN (Multi-scale Feature Pyramid Network), where "Re-Channel" denotes the channel integration stage, "Bi-Fusion" represents the first stage of feature fusion, and "Multi-Fusion" corresponds to the second stage of feature fusion.

3.2.1. Adaptive Feature Fusion.

In the Neck architecture of the YOLOv8 benchmark model, feature fusion was executed through a stacking methodology, where features were stacked along the channel dimension without consideration for spatial relationships. Particularly in the context of small object detection tasks, the egalitarian treatment of features across all scales in this stacking process might lead to inadequate focus on smaller objects, as the concatenated representation may be dominated by relatively larger background features. Drawing inspiration from BiFPN [39], we introduced an adaptive fusion approach to supersede the stacking method. Initially, each input feature undergoes convolutional processing, followed by the utilization of a SoftMax-activated convolutional layer to compute the weights assigned to each feature layer during the fusion process. This dynamic adjustment of the importance of each feature layer aids in strengthening the contributions of different layers during fusion, rendering it more adept for the requirements of small object detection compared to the original concatenation fusion method.

3.2.2. Multi-Cross Connections.

To address the challenge of detecting small and obscured objects in drone-captured images, we developed a multiscale connectivity network that utilizes large-scale feature maps. This network, built on the foundations of PANet, introduces a multiscale connectivity strategy that improves the detection of small targets. Our advanced architecture features a multi-cross connection with three key components: channel integration from the backbone network's feature maps, a Bi-Fusion process, and a subsequent Multi-Fusion phase. The outcomes of this phase, ReF2 through ReF5, are processed by the AD-C2f feature extraction module and forwarded to the detection head. Further details on these mechanisms will be discussed in subsequent sections.

1.Re-Channel

. Due to the requirement of maintaining consistency in the channel dimension for our adaptive fusion method, the channel numbers of the extracted features (b_i) from the backbone network were not uniform. To address this issue, we applied a $1 \times 1 \times C$ convolution operation to the features (b_i). This step was taken to standardize the channel numbers, facilitating subsequent multiscale connectivity and adaptive fusion processes.

2.Bi-Fusion

In the part of Bi-Fusion, there were three fusion stages for different scales, producing F2, F3, and F4. F4 were formed by the fusion of B4 and B5 after up-sampling. F3 was formed by the fusion of F4 (up-sampled and feature-extracted) with B4. F2 was formed by the fusion of F3 (up-sampled and feature-extracted) with B3. The specific formulas were expressed as follows, where \oplus represents adaptive fusion, $U(x)$ denoted up-sampling, and $E(x)$ represented the feature extraction module of AD-C2f.

The equation of the F4:

$$F_i = B(i) \oplus U(B(i + 1)), i = 4 \quad (16)$$

The expression of F2, F3.

$$Fi = B(i) \oplus U\left(E(F(i + 1))\right), i = 2, 3 \quad (17)$$

3. Multi-Fusion

. In the Multi-Fusion section, there were four fusion stages for different scales, producing ReF2, ReF3, ReF4, and ReF5. ReF2 was formed by the adaptive fusion of B2, F2, and F2 after feature extraction and down-sampling. ReF3 and ReF4 were formed by the adaptive fusion of Bi with the previous level's ReF(i) and ReF(i) after feature extraction and down-sampling. ReF5 was formed by the adaptive fusion of F4, ReF4, and ReF4 after feature extraction and down-sampling. The specific expressions were as follows, where D(x) represented down-sampling.

The equation of ReF2:

$$ReF(j) = B(j) \oplus D\left(E(F(j))\right) \oplus F(j), j = 2 \quad (18)$$

The equation of ReF3, ReF4:

$$ReFi = B(i) \oplus U\left(E(F(i + 1))\right) \oplus F(j), j = 3, 4 \quad (19)$$

The expression of ReF5:

$$ReF(j) = ReF(j - 1) \oplus D\left(E(ReF(j - 1))\right) \oplus F(j - 1), j = 5 \quad (20)$$

4 Experiments and Results

4.1 Datasets Analyses

Selected for this paper, VisDrone [33], an authoritative dataset in the field of international drone vision, was used as an experimental verification object. At present, drones had been widely used in various fields, such as agriculture, aerial photography, and personalized monitoring. Due to the comprehensive influence of shooting angle, light, background, and other factors, intelligent understanding of UAV visual data was more difficult than conventional computer vision tasks. To improve the performance of drone viewer task, the AISKYEYE team at the Machine Learning and Data Mining Laboratory of Tianjin University, proposed the VisDrone2019 dataset. Which consists with 288 video clips, including 261908 frames and 10209 still images. The dataset used a variety of drones for multisense, multitask shooting, including locations (taken from 14 different cities in China separated by thousands of kilometers), environments (urban and rural), objects (such as pedestrian, truck, bicycles, etc.), density (sparse and crowded scenes), weather (sunny and cloudy), and lighting conditions (day and night).

Unlike conventional detection datasets, each image might contain hundreds of objects to be detected, and the dataset contained a total of 2.6 million manual annotations of bounding boxes. In addition, VisDrone provided some important attributes such as scene visibility, object class, and occlusion to improve the utilization of data in various tasks. Some data examples were shown in Figure 9.

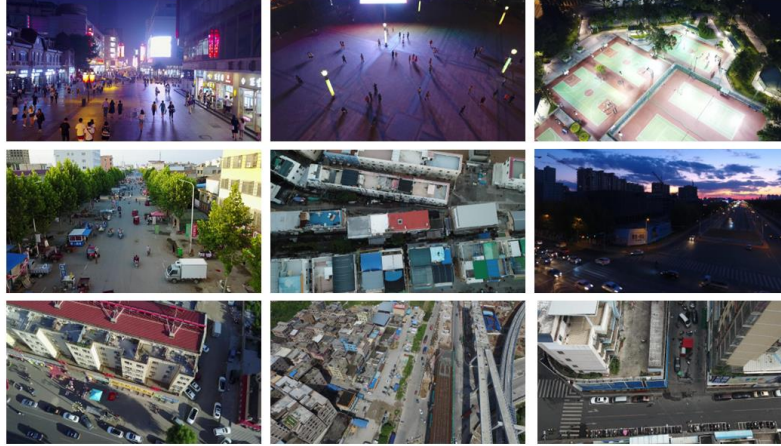


Fig. 5. Some data examples of the VisDrone2019 dataset.

4.2 Experiment Metrics

The experiment used the mean value of average precession at an IOU of 0.5(mAP50), the mean value of average precession crossed a range of IoU thresholds from 0.5 to 0.95(mAP95), frames per second (FPS), number of parameters, and model size as evaluation indices.

In this paper, we choose the Ubuntu 22.04 as the operating system with Python 3.8.18, PyTorch 1.13.1, Cuda 11.7 as the desktop computational software environment. The experiment utilized NVIDIA 3090 graphics cards as hardware. The implementation code of the neural network was modified based on the Ultralytics 8.0.114 version. The hyperparameters used during the training, testing, and validation of the experiment remained consistent. The detailed settings were displayed in the Table 3:

Table 1. Hyper-Parameters Setting table.

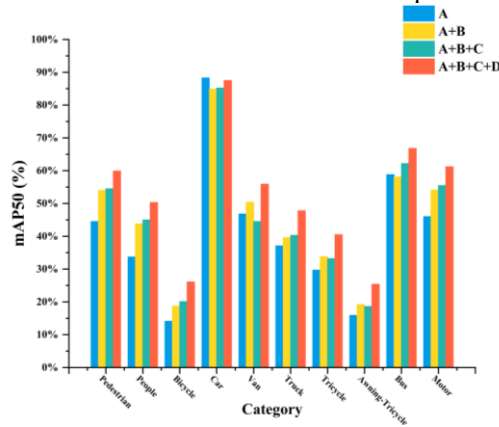
Hyper-Parameters	Setting
Epochs	300
Batch Size	8
Optimizer	SGD
NMS IoU	0.65
Initial Learning Rate	0.01
Final Learning Rate	0.0001
Momentum	0.937
Weight-Decay	0.0005
Image Size	640 x 640

Mosaic	1.0
Image Translation	0.1
Close Mosaic	Last 10 epochs

All the YOLOv8 and our designed MPB-YOLO algorithms had detection results from our experiments. In these experiments, none of these networks used any officially pre-training weight. The remaining data came from relevant cited papers.

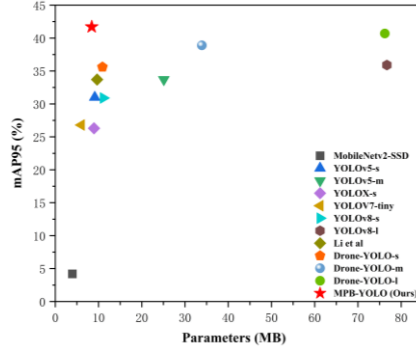
4.3 Ablation Study

The small target detector designed in this paper mainly improved the neck and backbone parts of the benchmark model (YOLOv8-s). To systematically analyzed the improvement of model performance of each unit, the benchmark model A, the improved model A+B(Extra-head), the improved model A+B+C(Extra-head+AD-C2f), the improved model A+B+C+D(Extra-head+AD-C2f+MFPN) were defined in turn, and the changes in the evolution indicators of the for models were quantitatively explored.



4.4 Comparison Experiment

Our experimental results were compared with the results of other models published on this dataset throughout the years, including MobileNetv2-SSD[34], YOLOv3[37], YOLOv4[18], YOLOv5[15], YOLOv6[13], YOLOv7[14], YOLOX[36], MS-YOLOv7[35], Drone-YOLO[28] and Li et al.'s[29]. These methods as well as YOLOv8s and YOLOv8l, were the baseline methods in this experiment. Our proposed MPB-YOLO performs best on mAP₉₅, while performs the second after the MS-YOLOv7[35] on mAP₅₀.



Compared to the SOTA(state-of-the-art) model: Drone-YOLO, our designed model: MPB-YOLO, achieved 2.45% accuracy improvement while using only 11.06% of the Drone-YOLO(large)'s parameters.

5 Visualization

A comparative visual analysis was conducted on the VisDrone2019-Test dataset to ascertain the efficacy of our proposed MPB-YOLO model against a benchmark counterpart. This comparison allowed for a more intuitive assessment of MPB-YOLO's superior detection capabilities from an aerial vantage point. We selected three distinct scenarios to illustrate the enhancements achieved by our model, as well as to highlight certain limitations.

In the first scenario, the image captured from an altitude of 100 meters above a city road exhibited objects appearing notably small, akin to the size of ponies. For both models, we designated three corresponding areas for comparison—regions from the benchmark model are labeled as 1, 2, and 3, while regions from MPB-YOLO detections are labeled as 4, 5, and 6.

As depicted in Figure 6:

(1) The first area comprised several vehicles on the road, where region 4 captured by our MPB-YOLO model demonstrated a higher detection count compared to region 1 from the benchmark.

(2) In the second area, a vehicle was situated within a complex environment. Our MPB-YOLO successfully identified the vehicle amid this intricate backdrop, a feat that the benchmark model failed to achieve.

(3) The third area featured two vehicles on a reflective surface. Here, the objects were discerned by MPB-YOLO in region 6 but went undetected by the benchmark model in region 3.

These scenarios underscore the enhanced detection performance of MPB-YOLO, particularly in challenging environments where the benchmark model falls short.

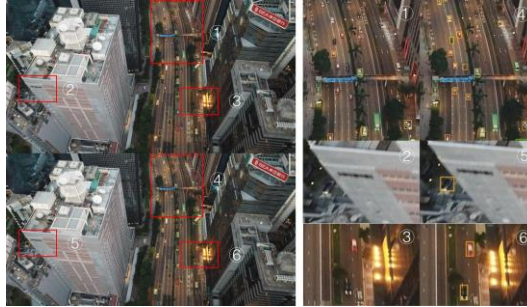


Fig. 6. A daytime urban street scene with normal lighting conditions. The image was captured from an elevated position exceeding 100 meters above the ground. The primary detection objects include cars, buses, and vans.

The second perspective, the picture was taken from about 30 meters above the viaduct at night, there was a parking lot in the middle of the picture, therefore the cars are stacking in a tiny zone of the picture, which bring a huge task for the detection model. In order to discuss the detection results, we picked four regions of the same position for both results, the regions selected from the benchmark model detection results are denoted by 1,2,3,4 and the regions selected from the detection results of MPB-YOLO are denoted by 5,6,7,8.

Analyze the visualization result in Figure 7: (1) Firstly, the regions where contains four cars in a situation that lack of light was choiced to compare the ability of two detection models, in 1 and 5, the benchmark model detected three targets, but one target mistake in the dark area; the MPB-YOLO detected only one target but without mistake. (2) Secondly, the results of the parking lot area was selected to compare because its high stacking proportion, comparing the result of 2 and 6, we can clearly see the detection results of the benchmark model in the face of stacking are not as accurate as the detection result of MPB-YOLO, there are no redundant detection boxes in MPB-YOLO's detection results, further demonstrate the sensitivity of our designed feature extraction model towards to the small size objects. (3) The next is a region containing two truchs and one car, compare 3 and 7, our MPB-YOLO detected accurately, while the benchmark model ignored all targets. (4) Compare the 4 and 8, when the benchmark model detected the truck's shadow as a car, MPB-YOLO detected accurately.



Fig. 7. A nightly urban street scene with poor lighting conditions. The image was captured from an elevated position exceeding 30 meters above the ground. The primary detection objects include cars, buses, and vans.

The third perspective, the picture was taken from about 40 meters above the school entrance area at noon, the pedestrian crowded together, people riding bicycle and motor, the size of these stacked objects is extremely small and difficult to distinguish, to further compare the detection performance of the models, we picked four regions of the same position for both results, the regions selected from the benchmark model detection results are denoted by 1,2,3,4 and the regions selected from the detection results of MPB-YOLO are denoted by 5,6,7,8.

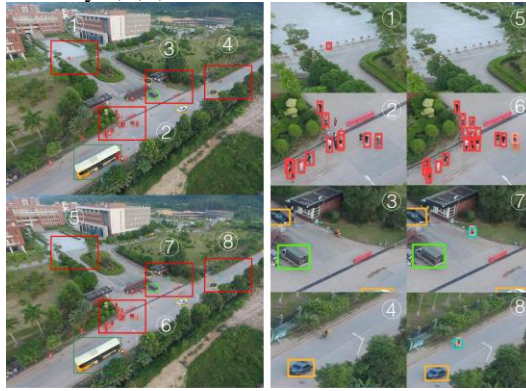


Fig. 8. A daytime school street scene with normal lighting conditions. The image was captured from an elevated position exceeding 20 meters above the ground. The primary detection objects include cars, buses, pedestrian, bicycle, motor, and crowded people.

As shown in Figure 12: (1) Compare 1 and 5, MPB-YOLO did not detect the stone as pedestrian like the benchmark model did. (2) Compare 2 and 6, there were total 18 subjects in this area, including 16 pedestrians and a person on a bicycle. The benchmark model detected about only 11 pedestrian at all, and did not recognized the cyclist as a people and a bicycle; the MPB-YOLO not only detected all the subjects, but also distinguish the cyclist as a people and a bicycle. (3) Compare 3,4 and 7,8, they all have a common characteristic: a motorcyclist in the center of the picture, in contrast to the benchmark model, MPB-YOLO detected the motorclist as a people and a motor, which can effectively reflect the feature sensitivity and the performance superiority of MPB-YOLO compared with the benchmark model.

6 Conclusion

In this study, we introduced MPB-YOLO, a novel feature extraction architecture based on a multi-scale adaptive fusion approach, designed to effectively detect multi-scale and small targets from aerial drone perspectives. MPB-YOLO integrates key innovations including an offset attention mechanism within variable convolutions,

enhancing object detection across various scales during feature extraction. This mechanism significantly improves the network's ability to identify features of differently scaled and closely situated targets, boosting overall detection performance.

Additionally, the use of expansive feature maps increases detection precision for small-sized objects. This is complemented by a multi-scale adaptive feature fusion strategy that leverages global contextual object features to enhance detection accuracy. Extensive testing on the Vis-Drone2019 dataset shows that MPB-YOLO outperforms existing advanced detection models, particularly in environments with complex backgrounds, tiny objects, and partially hidden targets, marking a significant advancement in aerial-target detection.

Future developments will broaden the algorithm's scope to include other detection paradigms such as infrared and hyperspectral imagery. A key goal moving forward is to create more comprehensive and diverse datasets that cover various complex real-world scenarios. This will help to rigorously test and improve the algorithm's generalization capabilities and resilience, enhancing its applicability and performance in diverse operational contexts.

Acknowledgments. This study was funded by the Fund of Guangdong Polytechnic of Science and Trade, Project No. GDKM2022-86, GDKM2022-121, the Fund of Science and Technology Program of Guangzhou City, Project No.SL2023A04J01572, the Fund of Guangdong Province Education Department, Project No. 2022KQNCX195, and the Fund 2023 Guangzhou Basic and Applied Basic Research Project, No. 2023A04J1716e;2023 Qingyuan City Integration of Industry and Education Social Sciences Special Project No.ZJCYJY202362, and the 2023 Guangdong Higher Vocational Education Teaching Steering Committee's Educational and Teaching Reform Research and Practice Projects in the Fields of Electronic Information and Communication, No. 15.

Disclosure of Interests. The authors declare no conflict of interest.

References

1. Bouguettaya, A.; Zarzour, H.; Kechida, A.; Taberkit, A.M. Vehicle Detection from UAV Imagery with Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* 2021, 33, 6047–6067.
2. Adaimi, G.; Kreiss, S.; Alahi, A. Perceiving Traffic from Aerial Images. *arXiv* 2020, arXiv:2009.07611.
3. Boží c-Štuli c, D.; Maruší c, Ž.; Gotovac, S. Deep Learning Approach in Aerial Imagery for Supporting Land Search and Rescue Missions. *Int. J. Comput. Vis.* 2019, 127, 1256–1278.
4. Chang, Y.-C.; Chen, H.-T.; Chuang, J.-H.; Liao, I.-C. Pedestrian Detection in Aerial Images Using Vanishing Point Transformation and Deep Learning. In *Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, 7–10 October 2018; pp. 1917–1921.

5. Chen, Y.; Lee, W.S.; Gan, H.; Peres, N.; Fraisse, C.; Zhang, Y.; He, Y. Strawberry Yield Prediction Based on a Deep Neural Network Using High-Resolution Aerial Orthoimages. *Remote Sens.* 2019, 11, 1584.
6. Cai, W.; Wei, Z. Remote Sensing Image Classification Based on a Cross-Attention Mechanism and Graph Convolution. *IEEE Geosci. Remote Sens. Lett.* 2020, 19, 1–5.
7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* 2014, arXiv:1409.1556.
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; pp. 770–778.
9. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017*; pp. 2881–2890.
10. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *arXiv* 2014, arXiv:1405.0312. Available online: <http://xxx.lanl.gov/abs/1405.0312>.
11. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. Available online: <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
12. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. Available online: <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
13. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* 2022, arXiv:2209.02976.
14. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, Canada, 18–22 June 2023*; pp. 7464–7475.
15. Jocher, G.; Stoken, A.; Chaurasia, A.; Borovec, J.; Kwon, Y.; Michael, K.; Changyu, L.; Fang, J.; Skalski, P.; Hogan, A.; et al. Ultralytics/Yolov5: V6.0—YOLOv5n 'Nano' Models, Roboflow Integration, TensorFlow Export, OpenCV DNN Support, 2021. Available online: <https://zenodo.org/record/5563715>.
16. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; pp. 779–788.
17. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017*; pp. 7263–7271.
18. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* 2020, arXiv:2004.10934.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* 2015, 28.
20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017*; pp. 2961–2969.
21. Goldman E, Herzig R& Eisenschtat A, et al. Precise Detection in Densely Packed Scenes[C]. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019. 5222~5231.

22. Bodla N, Singh B& Chellappa R, et al. Improving Object Detection with One Line of Code[C]. In: 2017 IEEE 16th International Conference on Computer Vision (ICCV). 2017. 5562~5570.
23. Ünel F Ö, Özkalayci B O& Çiğla C. The Power of Tiling for Small Object Detection[C]. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2019. 582~591.
24. Tang T, Zhou S& Deng Z, et al. Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining[J]. *Sensors*, 2017, 17(2):336.
25. Kong L, Zhu X& Wang G. Context Semantics for Small Target Detection in Large-Field Images with Two Cascaded Faster R-CNNs[J]. *Journal of Physics: Conference Series*, 2018, 1069:12138.
26. Yang F, Fan H& Chu P, et al. Clustered Object Detection in Aerial Images[C]. In: 2019 IEEE/CVF 17th International Conference on Computer Vision (ICCV). 2019. 8310~8319.
27. Li C, Yang T& Zhu S, et al. Density Map Guided Object Detection in Aerial Images[C]. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2020. 737~746.
28. Zhang, Z. Drone-YOLO: An Efficient Neural Network Method for Target Detection in Drone Images. *Drones* 2023, 7, 526.
29. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* 2023, 7, 304.
30. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. 2023. Available online: <https://github.com/ultralytics/ultralytics/blob/main/CITATION.cff>.
31. J. Dai et al., "Deformable Convolutional Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 764-773.
32. X. Zhu, H. Hu, S. Lin and J. Dai, "Deformable ConvNets V2: More Deformable, Better Results," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 9300-9308.
33. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea 27–28 October 2019.
34. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 4510-4520.
35. Zhao, L.; Zhu, M. MS-YOLOv7: YOLOv7 Based on Multi-Scale for Object Detection on UAV Aerial Photography. *Drones* 2023, 7, 188.
36. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* 2021, arXiv:2107.08430.
37. Redmon, J; Farhadi, A. YOLOv3: An Incremental Improvement. *arxiv* 2018, arXiv:1804.02767.
38. Liu, S.; Zha, J.; Sun, J.; Li, Z.; Wang, G. EdgeYOLO: An Edge-Real-Time Object Detector. *arXiv* 2023, arXiv:2302.07483.
39. M. Tan, R. Pang and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 10778-10787.