*Journal of Network Intelligence* ©2025 ISSN 2414-8105
**Taiwan Ubiquitous Information** Volume **10**, Number **4**, November **2025**

# An Improved YOLOv8 Algorithm for Detecting Small Targets in UAV Aerial Images

Jin-Ling Yang*, Shi-Wei Xue, An-Hong Wang, Jin-Liang Cao, Lin Li

College of Electronic Information and Engineering
Taiyuan University of Science and Technology
ShanXi 030024, P. R. China
yangjl@tyust.edu.cn, 1569233084@qq.com, ahwang@tyust.edu.cn, ccjl697@tyust.edu.cn, 2023077@tyust.edu.cn

Jing Yu

Security and Prevention Department
Zhejiang Police Vocational Academy
Hangzhou 310018, P. R. China
1326891880@qq.com

*Corresponding author: Jin-Ling Yang

Received July 18, 2024, revised January 6, 2025, accepted June 13, 2025.

Abstract. *Identifying small targets in UAV aerial images is a challenging problem in computer vision due to its complex nature and ambiguous features. This paper presents the YOLOv8 algorithm developed to detect small targets in drone images. Firstly, the InceptionNeXt convolutional module is integrated into the backbone network, where deep convolutional operations are performed on partial channels to capture feature information, achieving a meticulous equilibrium between the model's detection accuracy and speed. Secondly, the neck part is enhanced by integrating the Low-level Feature Alignment mechanism, which retains more shallow features through cross-scale connections, augmenting the model's potential for integrating multi-scale features. Furthermore, the ShareSepHead detection head is employed to share detection head parameters across scales, enhancing the model's ability to accurately identify and detect small-sized objects. Lastly, to improve the training effectiveness of the dataset, the MPDIoU loss function is introduced to fully explore horizontal rectangular geometric features. Experimental results on the VisDrone2019 UAV aerial dataset show that the improved YOLOv8 achieves the mAP50 of 35.8% and the mAP50:95 of 21.3%, representing a 4.5% and 3.7% increase over the original YOLOv8. The presented algorithm effectively improves the accuracy of UAV detection for diminutive targets, thereby facilitating a heightened level of precision in target identification.*

**Keywords:** drone images; small object detection; YOLOv8; feature fusion; VisDrone

## 1. Introduction.

Recently, the rapid development of the drone industry has greatly benefited human production and life. UAV aerial detection technology has been widely applied in civilian fields such as traffic supervision and military areas like emergency rescue, which not only enhances the drones' detection capabilities, but also makes them more intelligence At present, the great progress made by the Internet of Things has led to the emergence of the Internet of Drones [1-5]. With the rapid development of artificial intelligence, deep learning has demonstrated remarkable performance, which leads the third wave of artificial intelligence. Object detection algorithms have evolved from traditional methods of manually extracting features to deep learning-based techniques.

These deep learning-based detection algorithms are categorized into one-stage and two-stage approaches based on whether candidate regions are generated or not. Representative one-stage detection algorithms include SSD [6], YOLO series [7], and RetinaNet [8] et al. Typical representative of two-stage detection algorithms, includes the R-CNN [9] series et al, two-stage algorithms generally exhibit higher average precision, but it has drawbacks such as complex network models, large computational complexity, and slow detection speed, making real-time requirements challenging to meet. Although single-stage detection algorithms may have lower detection accuracy compared to two-stage counterparts, their simpler network structures and faster detection speeds make them more suitable for real-time applications and deployment on UAVs. UAVs have limited computational power for hardware devices, but it demands high real-time and accuracy requirements for object detection models. Lightweight models struggle to improve detection accuracy, while complex network models face challenges in deployment on small UAV devices. Moreover, considering the characteristics of UAV aerial images, such as blur, numerous small targets, and unclear features, it is a critical challenge that how to improve the algorithms so that the model can balance detection speed and accuracy. Many scholars worldwide have conducted extensive research in small target detection in UAVs, producing significant research outcomes. Chen and Li [10] proposed an improved YOLOv5 algorithm, which incorporates C2F modules and the CARAFE upsampling algorithm into the model's backbone network to enhance feature extraction and fusion capabilities. Xie et al. [11] proposed an enhanced algorithm, Drone-YOLO,which adds a multi-scale channel attention mechanism feature fusion module into the original model, to improve the model's feature fusion capabilities. Qiu et al. [12] proposed the ECA attention mechanism into the model and adding the adaptive spatial feature fusion module to the feature pyramid improves the ability of the convolutional neural network to extract effective information within the feature map. Zhang et al. [13] proposed an improved algorithm based on YOLOX, embedding a CA attention mechanism into the model's neck part, introducing the ASF-F module into the feature fusion section, and changing the loss function to strengthen learning from positive samples. Liang et al. [14] devised an enhanced algorithm, EdgeYOLO, by creating a hybrid random loss function to enable real-time object detection on edge devices, thereby enhancing the detection accuracy for small targets. Wang et al. [15] optimized the feature pyramid structure based on RetinaNet to increase attention towards targets, incorporating an attention mechanism. Zhao et al. [16] introduced an improved algorithm, YOLOv8-smr, by reducing the number of layers in the backbone network, which introduces the context enhancement modules and spatial-channel filtering modules, to enhance the localization and classification effects, and improve feature extraction capabilities. The above methods have improved the detection accuracy of the model to a certain extent, but there are still a large number of problems such as missed detection and false detection. To address existing issues in target detection under the UAV perspective, such as low accuracy in detecting small targets, the inability to balance accuracy and real-time performance, and the occurrence of false positives and misses, this paper focuses on enhancing the models' ability to extract and fuse target features. The main innovations of this article are as follows:

1. The InceptionNeXt convolutional module is integrated into the backbone network, where deep convolutional operations are performed on partial channels to capture feature information, thereby balancing the detection accuracy and speed of the model.
2. The neck part is enhanced by integrating the Lowlevel Feature Alignment mechanism, which retains more shallow features through cross-scale connections, enhancing the model's multi-scale feature fusion capability.

3. The ShareSepHead detection head is employed to share detection head parameters across scales, enhancing the model's capability to detect small targets.
4. To improve the training effectiveness of the dataset the MPDIoU loss function is introduced to fully explore horizontal rectangular geometric features.

## 2. YOLOv8 Network Structure.

From the perspective of model size, YOLOv8 has five versions: n, s, m, l, and x. Among them, YOLOv8n has the smallest network width and depth, and requires the least resources, so it is suitable for practical applications in drones. YOLOv8n serves as a benchmark model for improvement. The structure consists of four parts: Input, Backbone, Neck, Head. Mosaic data enhancement, adaptive anchor frame computation and adaptive gray scale filling are performed at the input side. The backbone network consists of Conv, C2f and SPPF. C2f draws on the ELAN idea of YOLOv7, which ensures lighter weight and obtains richer gradient flow information by parallelizing more branches and adding the Split operation, while the Neck module adopts the PANet structure to capture contextual information and strengthen the fusion capability of network features. The output decouples the classification and detection processes. The model structure is shown in Figure 1.

1. Input: Mosaic data augmentation is applied to enrich the dataset for input images. Adopting the Anchor Free strategy reduces the number of predicted boxes, speeding up the NMS process.
2. Backbone: Modules such as C2f, Conv, SPPF are used. The C2f module is lighter than the C3 module and provides more abundant gradient flow information. The Conv module conducts convolution, normalization, SiLU activation on input images. The SPPF module extracts and encodes features from images at different scales.
3. Neck: Includes FPN [17] and PAN [18]. FPN strengthens semantic features from top to bottom, while PAN enhances positional features from bottom to top. A combination of FPN and PAN facilitates effective feature fusion of different stage feature maps.
4. Head: Adopts the Decoupled Head strategy by separating classification and detection heads. Utilizing three different scale feature maps, the model obtains target category and location information.

## 3. YOLOv8 Improvements.

### 3.1. Improvement of Backbone Networks.

The original YOLOv8 backbone network uses regular convolution, C2f modules, and SPPF modules to extract high-quality features from images. In order to enhance the model's feature extraction capabilities, the InceptionNeXt deep convolutional module. Yu et al. [19] is used to replace the first two C2f modules in the backbone network. Firstly, the large kernel is decomposed into several groups of smaller convolutional kernels, with 1/3 of the channels using a $3 \times 3$ kernel, another 1/3 of the channels using a $1 \times k$ kernel, and the remaining 1/3 of the channels using a $1 \times 3$ kernel. This new, simple, and cost-effective operator is called the Inception Depthwise Convolution. Based on this operator, the InceptionNeXt module is constructed, which decomposes the large kernel depthwise convolution into four parallel branches along the channel dimension, including small rectangular convolutional kernels, two orthogonal stripe convolutional kernels, and an identity mapping. Figure 2 shows the structure of the InceptionNeXt module.

For input $X$, it is divided into four groups along the channel dimension:

$$X_{hw},\, X_w,\, X_h,\, X_{id} = \text{Split}(X) = X_{:g},\, X_{:g:2g},\, X_{:2g:3g},\, X_{:3g:}. \tag{1}$$

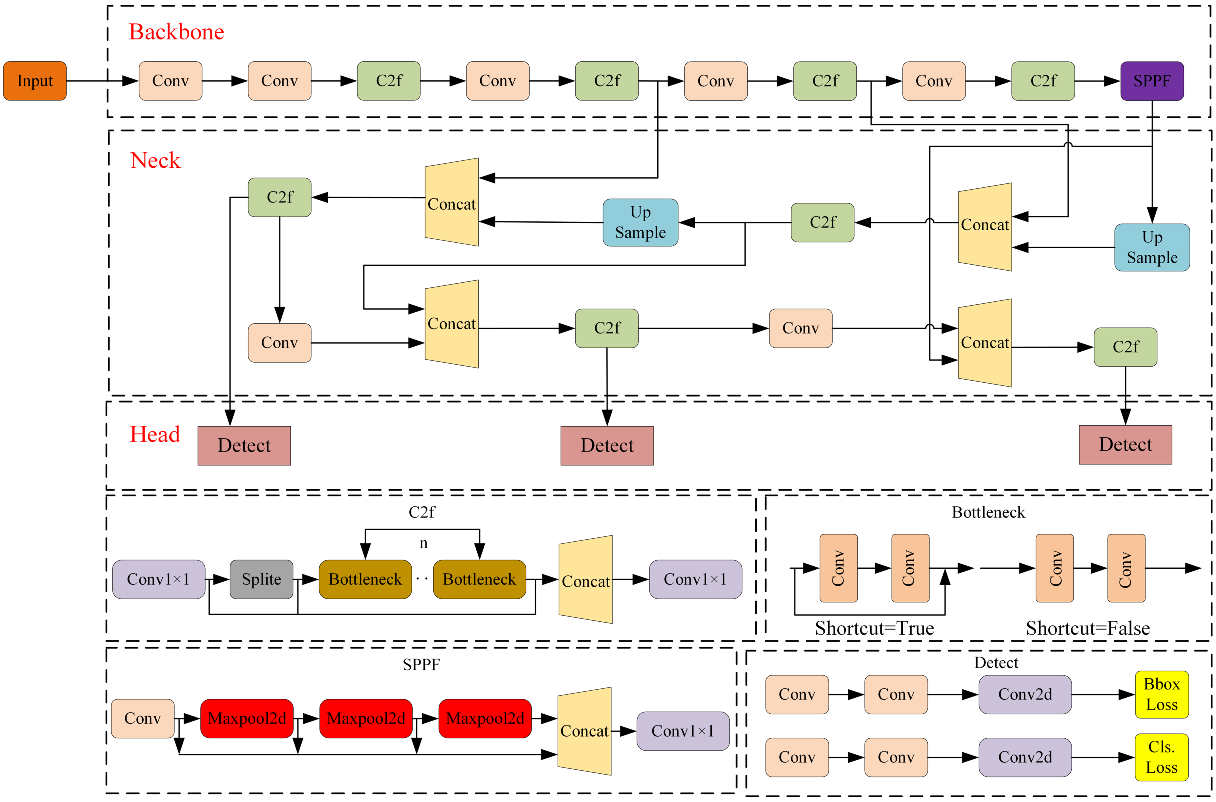


Figure 1. YOLOv8 original network structure

Where $g$ is the number of channels in the convolutional branch. The four features are processed by four different operators:

$$\begin{aligned} X'_{\mathrm{hw}} &= \mathrm{DWConv}^{g\to g}_{k_s\times k_s} g(X_{\mathrm{hw}}), \\ X'_{\mathrm{w}} &= \mathrm{DWConv}^{g\to g}_{1\times k_b} g(X_{\mathrm{w}}), \\ X'_{\mathrm{h}} &= \mathrm{DWConv}^{g\to g}_{k_b\times 1} g(X_{h}), \\ X'_{\mathrm{id}} &= X_{\mathrm{id}}. \end{aligned} \tag{2}$$

Where $k_s$ denotes the small square kernel size set as 3 by default, $k_b$ represents the band kernel size set as 11 by default. Finally, the output of each branch is concatenated:

$$X' = \mathrm{Concat}(X'_{hw}, X'_{w}, X'_{h}, X'_{id}) \tag{3}$$

$X'$ then passes through the Norm layer and the MLP layer consisting of two $1 \times 1$ convolutions to get activated:

$$X'' = \mathrm{MLP}\big(\mathrm{Norm}(X')\big) \tag{4}$$

Finally, the input features are fused with $X''$ by residual linkage.

Standard convolution extracts both spatial and channel features, but DWConv decomposes feature extraction into 2 processes:

1. Using $C$ convolution kernels of size $K \times K$ to do channel-by-channel convolution on the input feature maps, where one of the convolution kernels does convolution on only one channel of the input feature maps to obtain $C \times H \times W$ feature maps.
2. $N$ filters (each consisting of $C \times 1 \times 1$ convolution kernels) are used to do point-by-point convolution on the output feature maps in step 1 to obtain the output feature maps.

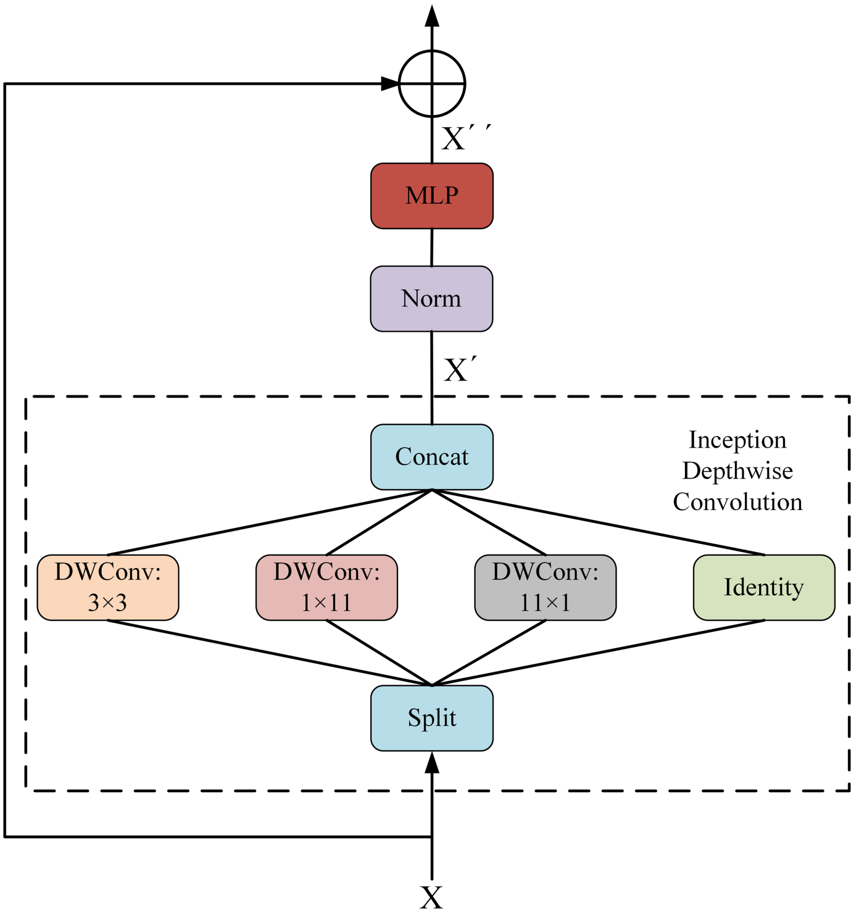


Figure 2. InceptionNeXt module

Given an arbitrary input feature map $F_i \in \mathbb{R}^{C\times H\times W}$, the output feature map $F_o \in \mathbb{R}^{N\times P\times Q}$ is obtained, where $C$ is the number of channels of the input feature map, $H$ is the height of the input feature map, $W$ is the width of the input feature map, the size of the convolution kernel is $K \times K$, $N$ is the number of channels of the output feature map, $P$ is the height of the output feature map, $Q$ is the width of the output feature map. The principle of depth-separable convolution(DWConv)is shown in Figure 3.

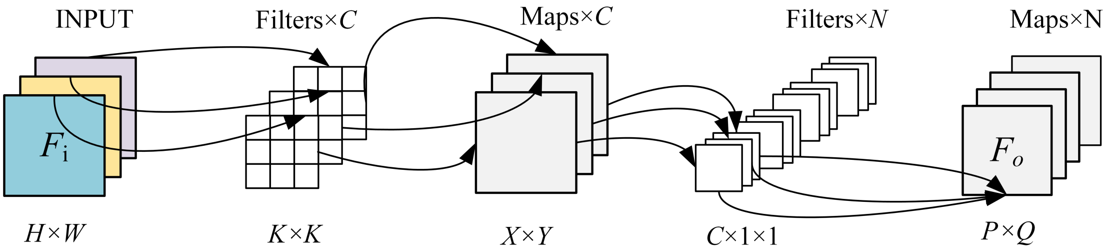
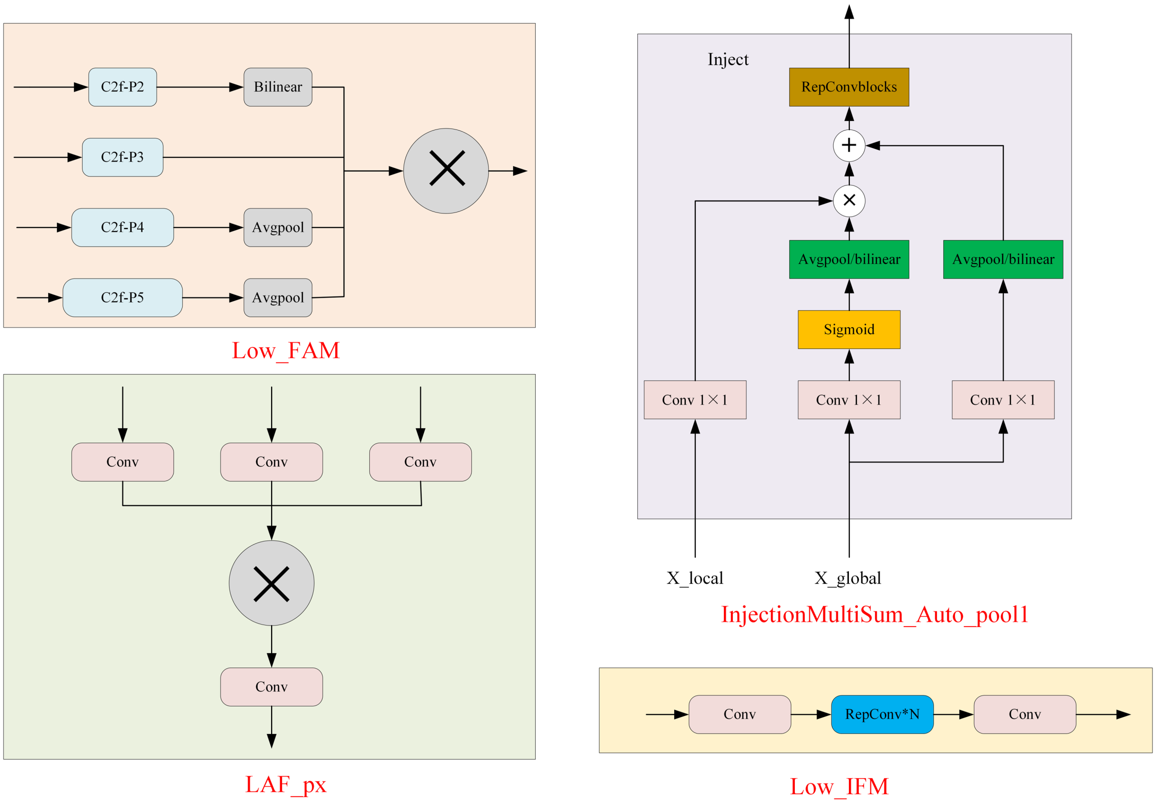


Figure 3. DWConv Structure

3.2. **Improvement of Neck Networks.** The insufficient accuracy is a major issue in aerial image object detection. Due to the deepening of the network, some shallow features are eliminated from the network, while information about small-sized objects primarily exists in shallow positions. This issue can be effectively addressed through multi-scale feature fusion methods. The neck structure of the YOLO series adopts the traditional FPN structure, which consists of multiple branches for multi-scale feature fusion. It can only fully fuse features from adjacent layers, but for information from other layers, it must be obtained indirectly through recursion. This transmission mode may result in a significant loss of information during computation. The information exchange between layers can only swap the selected information of intermediate layers, while the unselected information is discarded during transmission, which leads to a situation where the information from one layer can only help adjacent layers adequately, weakening the assistance provided to other global layers. Consequently, the overall effect of information fusion may be limited.To avoid information loss during the transmission process in the traditional FPN structure, a new gather-distribute mechanism was constructed [20], which collects and fuses information from various levels using a unified module and then distributes it to different levels. This not only avoids the inherent information loss in the traditional FPN structure but also enhances the partial information fusion capability of the neck without significantly increasing latency. It designs three modules: the Feature Alignment Module (FAM), the Information Fusion Module (IFM), and the Information Injection Module (Inject). The collection process are as follows. First, FAM collects and aligns features from various levels. Second, IFM fuses aligned features to generate global information. After the injection module acquires the fused global information from the collection process, it distributes this information to different levels and injects it using a simple attention operation, thereby enhancing the detection capability of the branches. Figure 4 shows the specific structures of each module.



Low_FAM

LAF_px

InjectionMultiSum_Auto_pool1

Low_IFM

Figure 4. Four types of submodules

In the Low-Level Feature Alignment Module (Low-FAM), downsampling of input features is achieved by average pooling (AvgPool) operation to get a uniform size. By adjusting the feature sizes to the smallest feature size in the group($R_{B4} = \frac{1}{4}R$), $F_{\text{align}}$ is got to ensure efficient aggregation of Low-FAM module information. The Low-Level Information Fusion Module (Low-IFM) design includes multi-layer reparameterization convolution blocks (RepBlock) and splitting operations. RepBlock takes $F_{\text{align(Channel=sum}(C_{B2},C_{B3},C_{B4},C_{B5}))}$ as input. Produces $F_{\text{align(channel=}C_{B4}+C_{B5})}$, the features generated by RepBlock are split along the channel dimension into $F_{\text{inj\_}P3}$ and $F_{\text{inj\_}P4}$, and then fused with features at different levels. The formula is:

$$F_{\text{align}} = \text{Low\_FAM}([B2, B3, B4, B5]), \tag{5}$$

$$F_{\text{fuse}} = \text{RepBlock}(F_{\text{align}}), \tag{6}$$

$$F_{\text{inj\_}P3},\ F_{\text{inj\_}P4} = \text{Split}(F_{\text{fuse}}). \tag{7}$$

3.3. **Improvement of Head Networks.** In small object detection using drones, the detection head of the object detection model should possess three capabilities: scale perception, spatial perception, and task perception, to accurately detect and identify objects of different scales, positions, and shapes. To enhance the performance of the detection head, this paper replaces the YOLO head with a Shared Detection Head (ShareSepHead [21]). Real-time object detectors typically use separate detection heads for different feature scales to enhance the model's capabilities for higher performance, instead of sharing one detection head across multiple scales. The Shared Detection Head uses shared convolutional weights and separate normalization layers to predict classification and regression results for bounding box detection. The structure of ShareSepHead is shown in the Figure 5.

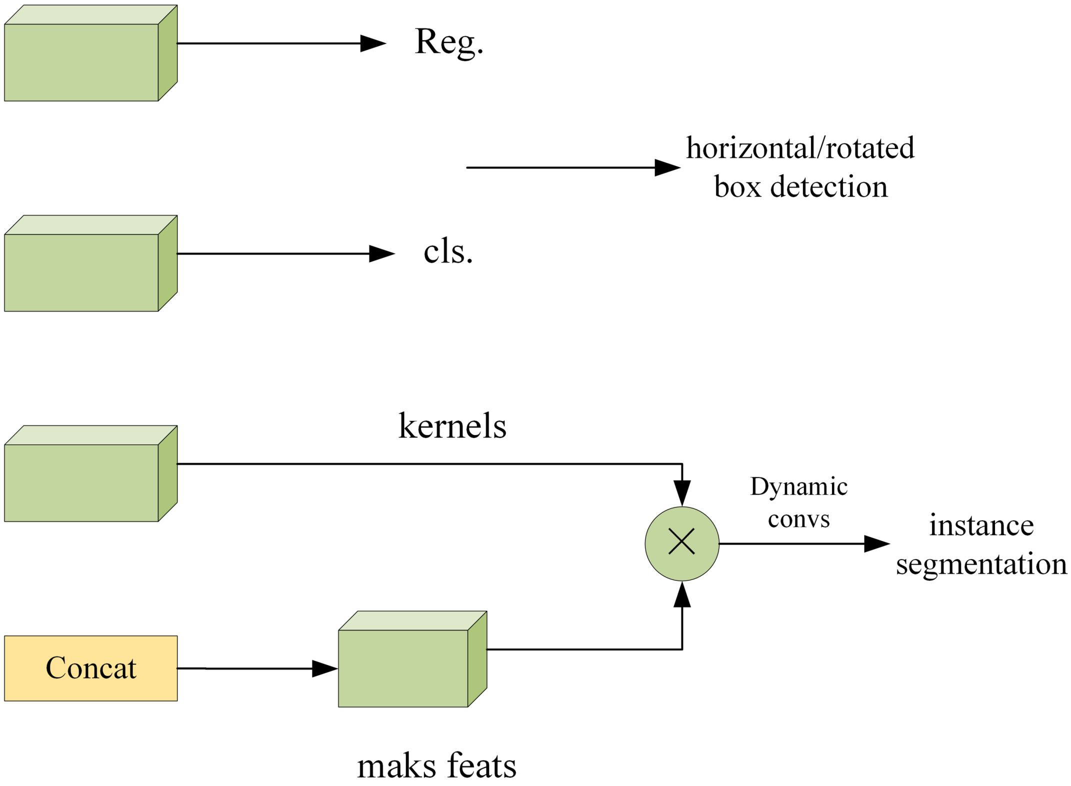


Figure 5. ShareSepHead structure diagram

3.4. **Improvement of Loss Function.** YOLOv8 uses the CIoU Loss [22] as the bounding box regression loss function, where the ground truth box is $\bar{B}_{gt} = [x_{gt}, y_{gt}, w_{gt}, h_{gt}]$, the predicted box is $\bar{B} = [x, y, w, h]$, $\bar{B}_{gt}$ and $\bar{B}$ correspond to the predicted bounding box's center coordinates and dimensions. The original definition of CIoU Loss is given by the following equation:

$$L_{\text{CIoU}} = L_{\text{IoU}} + \frac{(x - x_{gt})^2 + (y - y_{gt})^2}{W_g^2 + H_g^2} + \alpha\nu \tag{8}$$

$$L_{\text{IoU}} = 1 - \text{IoU} = 1 - \frac{W_i H_i}{wh + w_{gt}h_{gt} - W_i H_i} \tag{9}$$

$$\alpha = \frac{\nu}{L_{\text{IoU}} + \nu} \tag{10}$$

$$\nu = \frac{4}{\pi^2}\left(\tan^{-1}\frac{w}{h} - \tan^{-1}\frac{w_{gt}}{h_{gt}}\right)^2 \tag{11}$$

Where $L_{\text{IoU}}$ is used to measure the degree of overlap between the predicted box and the ground truth box, $\alpha$ is the equilibrium parameter, $\nu$ is used to measure the consistency of the aspect ratio, the remaining parameters are defined in the Figure 6.

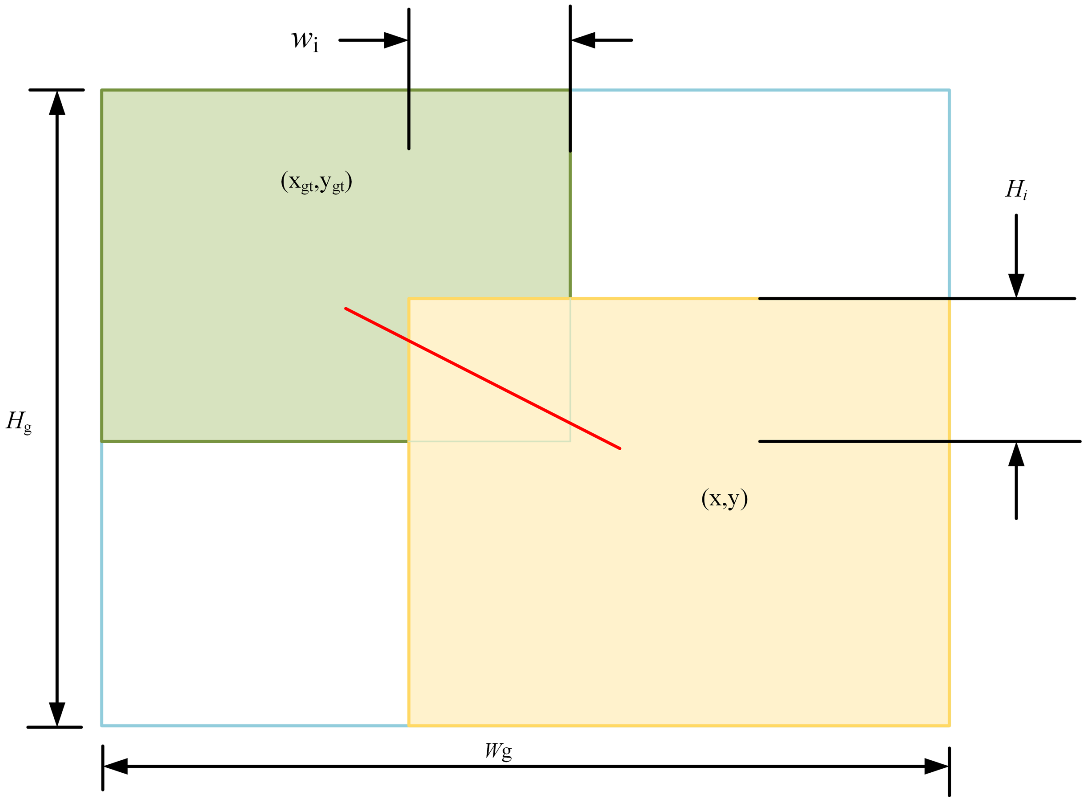


Figure 6. Relevant parameter definitions

CIoU Loss assumes that the training data has high quality, and focuses on enhancing the fitting ability of boundary frame Loss, without considering the harm of low-quality data to model performance. To solve this problem, MPDIoU Loss [23] is adopted to replace CIoU Loss, which incorporates all relevant factors considered in existing loss functions, namely overlapping or non-overlapping regions, center point distance, width and height deviation, while simplifying the calculation process. The calculation process of MPDIoU is shown in Table 1.

Table 1. MPDIoU calculation process

**Algorithm 1** Intersection overUnion with Minimum Points Distance

```
**Input:** Two arbitrary convex shapes: $\mathcal{A}, \mathcal{B} \subseteq \mathbb{R}^n$ ,   width and height of input image: $w, h$
**Output:** MPDIoU
For $\mathcal{A}$ and $\mathcal{B}$ , $(x_1^A, y_1^A), (x_2^A, y_2^A)$ denote the top-left and bottom-right point coordinates of $\mathcal{A}$ , $(x_1^B, y_1^B), (x_2^B, y_2^B)$ denote the top-left and bottom-right point coordinates of $\mathcal{B}$
$d_1^2 = (x_1^B - x_1^A)^2 + (y_1^B - y_1^A)^2$
$d_2^2 = (x_2^B - x_2^A)^2 + (y_2^B - y_2^A)^2$
$MPDIoU = \frac{A(\mathcal{I}_B)}{A(\mathcal{U}_B)} - \frac{d_1^2}{w^2 + h^2} - \frac{d_2^2}{w^2 + h^2}$
```

MPDIoU simplifies the similarity comparison between two bounding boxes, which can be adapted to overlapping or non-overlapping bounding box regression.

So far, this paper has completed the improvement of the model, and the improved network model is shown in Figure 7. First, the InceptionNeXt convolutional module is used to replace the first two C2f modules in the backbone network. Second, the neck of the model is designed based on each sub-module in Section 3.2, which integrates feature alignment, information fusion and information injection to reconstruct the feature pyramid structure and enhance the model's ability to detect objects of different sizes. Again, ShareSepHead is used instead of the original detection head of the model. Finally, MPDIoU Loss is used instead of CIoU Loss to fully exploit the geometric features of the horizontal rectangle and improve the overall performance of the model.

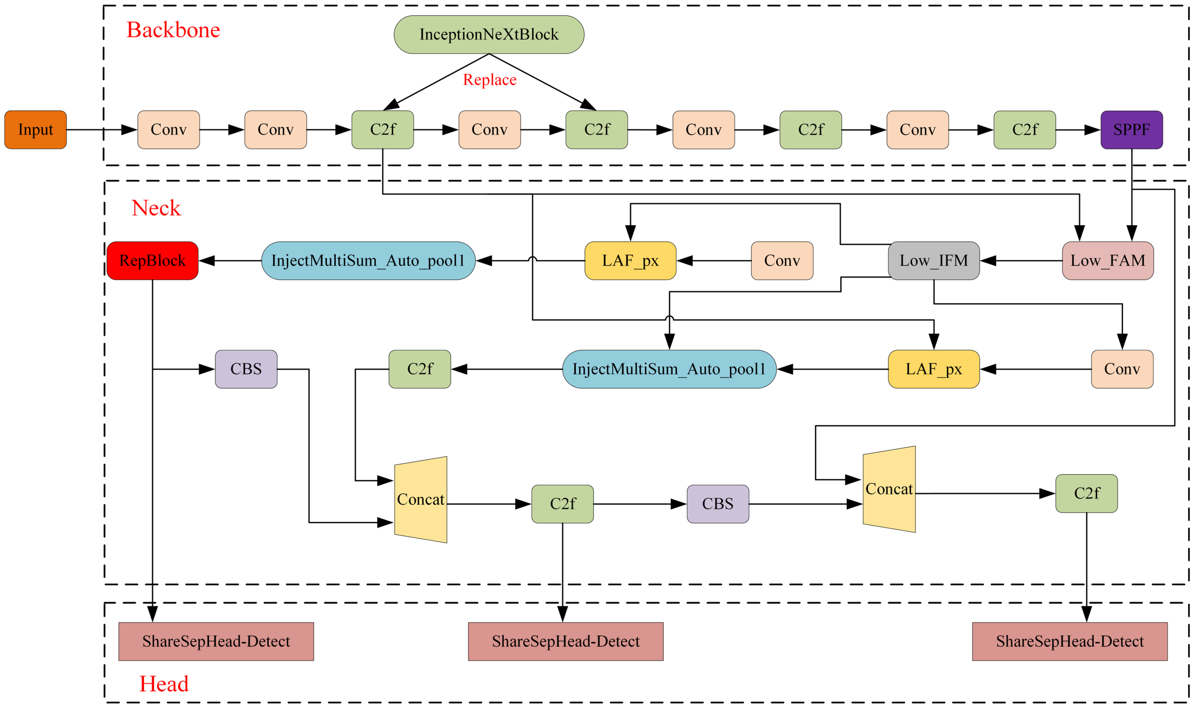


Figure 7. Improved YOLOv8 network structure

## 4. **Experimental Results and Analysis.**

4.1. **Experimental Datasets.** To evaluate the effectiveness and sophistication of the improved algorithmic model, this paper chooses the publicly available dataset VisDrone2019 [24] as the dataset for the experiments. The dataset contains 10 category targets: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus and motor, and the distribution of the categories is shown in Figure 8. The dataset contains training, validation, and testting sets with 6471, 548, and 3190 images, respectively, where the testting set is subdivided into two categories: challenge and dev, with 1580 and 1610 images, respectively. Most of the images in the dataset are small targets, large targets are scarce, the category confusion is serious, and it contains a variety of complex scenes. Therefore, for the significant features of the UAV aerial photography perspective, this dataset is choosed to conduct relevant experiments to verify the effectiveness of the improved algorithm.

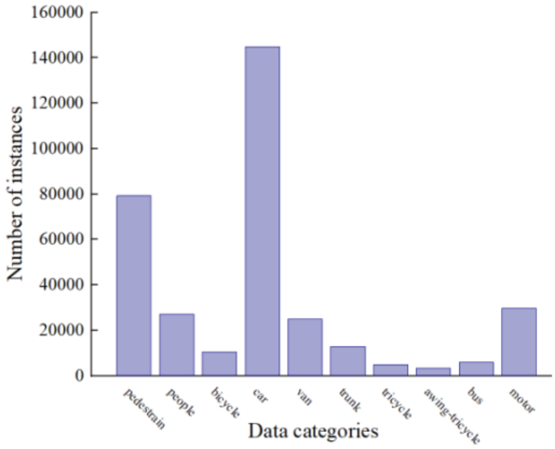


Figure 8. Category Distribution of VisDrone2019 Dataset

4.2. **Experimental Environment and Parameter Configuration.** This paper uses Windows operating system, the compiler is Python3.9.19, PyTorch1.12.1, CUDA11.3, numpy is 1.22.4. The hardware platform environment for the experiment: CPU: 12th Gen Intel(R) Core(TM) i7-12700F, GPU: NVIDIA GeForce RTX 1650, 16GB operating memory. Parameter setting size: input image size is modeled as 640×640 size, batch size is 16, and experiments use consistent hyperparameters.

4.3. **Experimental Evaluation Indicators.** In this paper, *Precision*, *Recall*, *mAP50* and *mAP50:95*, *Parameters*, *GFLOPs* and *FPS* is used as model performance evaluation metrics. Among them, *Precision* is a measure of the accuracy of model detection, *Recall* represents the ability of the model to find the correct samples, *Parameters* is used to measure the consumption of computational memory resources, *GFLOPs* is used to measure the computational complexity of the model training, and *FPS* is used to measure the real-time performance of the network model. The calculation of Precision, recall, AP, and mAP are given in the following equation:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{13}$$

$$AP = \int_0^1 P\, dR \tag{14}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{15}$$

Where $TP$ represents that both model prediction and actual are true, $FN$ represents that both prediction and actual are false, $FP$ indicates that the prediction is true and the actual is false, $N$ is the number of object classes, and $AP_i$ represents the average detection accuracy of the $i$th class of objects.

## 4.4. **Experimental Results.**

4.4.1. *Visualization of Experimental Results Analysis and Detection.* To verify the effectiveness of the improved algorithm, the verification results of the initial algorithm and the improved algorithm are compared, and the comparison indexes are: Precision, Recall, mAP50, mAP50:95, and the comparison results are shown in Table 2. It can be seen that the improved algorithm has improved in these indicators, respectively: 4.1%, 0.9%, 4.5%, 3.7%, demonstrating that the improved algorithm enhances the feature extraction and fusion capabilities of the model and it can effectively improve the detection accuracy of small targets.

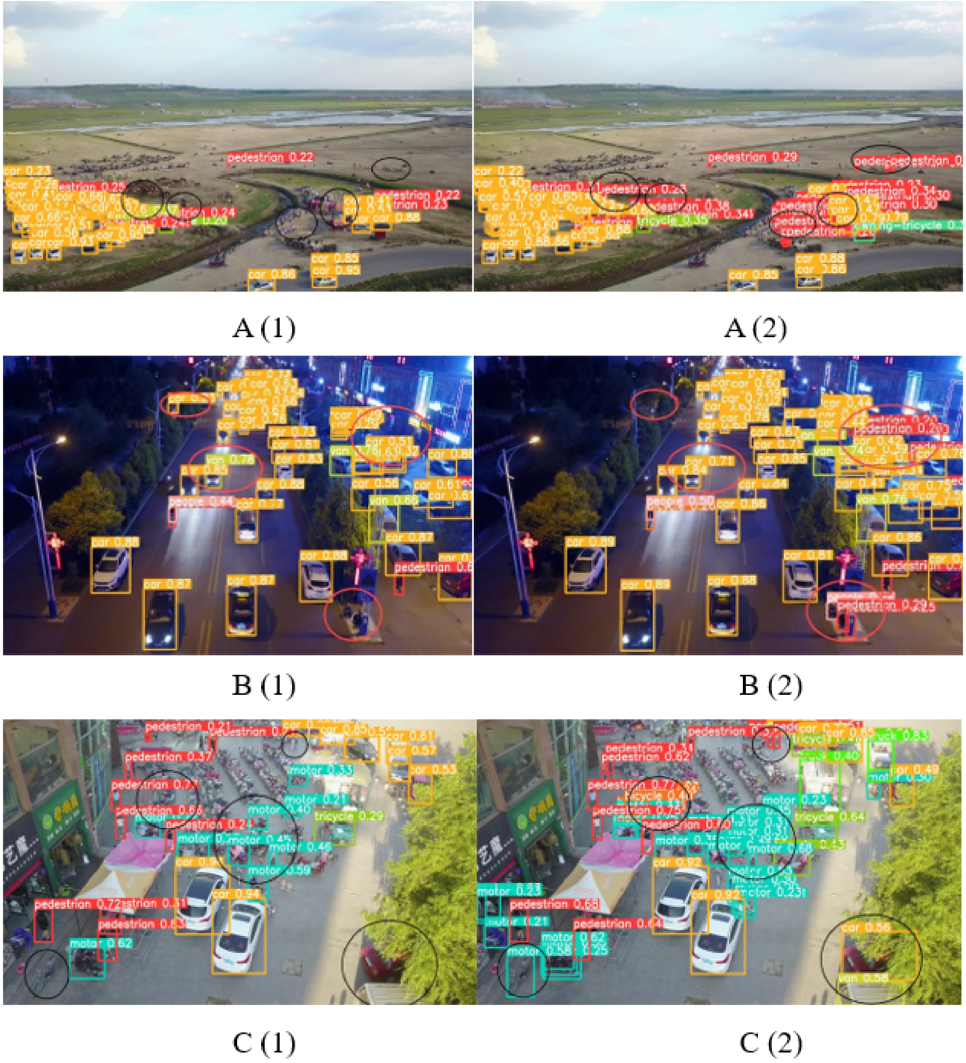
Table 2. TABLE 2. Comparison of Algorithms on the dataset

| | **P/%** | **R/%** | **mAP50/%** | **mAP50:95/%** |
|---|---|---|---|---|
| Initial | 42.2 | 33.1 | 31.3 | 17.6 |
| Improved | 46.3 | 34.0 | 35.8 | 21.3 |

In order to more intuitively demonstrate the detection performance of the improved algorithm in this article, different scenarios from the VisDrone2019 dataset were selected as the test dataset, and the detection results are shown in Figure 9. On the left is the detection result of the original YOLOv8n model is shown on the left, and on the right is the detection result of the improved algorithm proposed in this paper. A, B and C contain many typical targets of different scales, including targets for cars, motorcycles, and pedestrians. Figure B shows the nighttime parking area. Figure 9 B (1) shows that the original algorithm missed some pedestrians and misjudged cars as trucks, while the improved algorithm missed one car in Figure 9 B (2), but recognized more pedestrians and cars. Compared with the original algorithm, the recognition accuracy has been improved. In Figures A and C, the improved algorithm also recognized more pedestrians, cars, and motorcycles. In summary, the improved algorithm exhibits better detection performance and generalization ability in different scale scenarios.

Table **??** shows the comparison of different categories of targets by different algorithms on the VisDrone2019 dataset. From the Table **??**, it can be seen that the improved algorithm improves Precision for various categories, especially in trunk, tricycle and pedestrian detection, which are smaller scale categories in the dataset and have a large number of aggregations in many scenarios. This indicates that the improved algorithm can better focus on small targets, improve the detection ability of small targets, improve the detection accuracy, alleviate the leakage and false detection phenomenon of the model in the case of target aggregation, and have important application potential in the fields of urban traffic safety monitoring and human flow density monitoring.

Figure 10 visualizes the changes in the two metrics of mAP50 and mAP50:95 before and after the improvement of the algorithm, where the black curve represents the original YOLOv8 model and the red curve is the model of the improved algorithm. From the comparison of the curves in Figure 10, the model can finally reach the convergence state with the increase of training rounds in the case of 150 rounds of training, and the



A (1)

A (2)

B (1)

B (2)

C (1)

C (2)

Figure 9. FIGURE 9. Visual comparison of detection

Table 3. TABLE 3. Comparison of Precision for various categories

| Categories | YOLOv8n/% | Improved/% | Rise/% |
|---|---|---|---|
| pedestrain | 41.2 | 48.4 | 7.2 |
| people | 48.0 | 52.6 | 4.6 |
| bicycle | 17.0 | 24.1 | 7.1 |
| car | 62.9 | 66.9 | 4.0 |
| van | 44.3 | 51.9 | 7.6 |
| trunk | 38.3 | 50.4 | 12.1 |
| tricycle | 33.5 | 41.8 | 8.3 |
| awning-tricycle | 23.6 | 28.0 | 4.4 |
| bus | 55.8 | 59.5 | 3.7 |
| motor | 43.3 | 49.5 | 6.2 |

convergence values of mAP50 and mAP50:95 of the improved algorithm are higher than those of the original model, which proves that the improved algorithm has an improved detection ability for small targets.

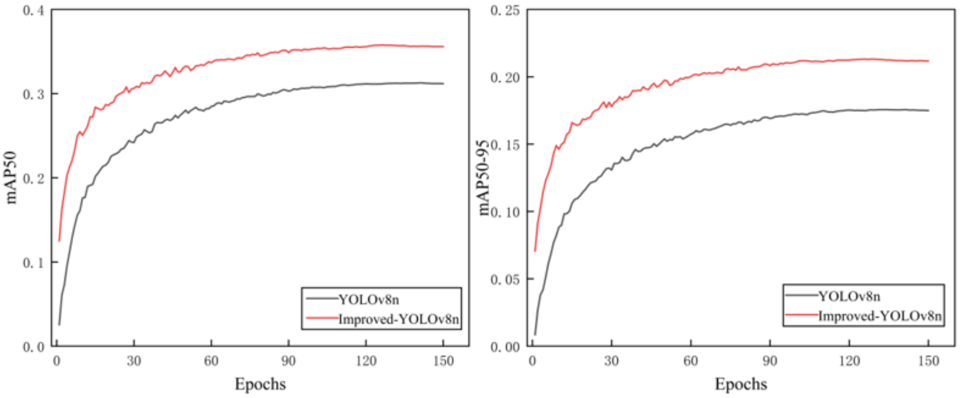


Figure 10. Comparison of Average Precision Mean

4.4.2. *Experimental Comparison of Different Loss Functions.* In order to verify the effectiveness of the introduced MPDIoU loss function, we compared it with other mainstream loss functions, and the comparison results are shown in Table 4. From Table 4, it can be seen that after using the MPDIoU loss function, the precision, recall, mAP50 and mAP50:95 of the model reach the maximum value, and the precision, recall, mAP50 and mAP50:95 are all improved compared with the original model. The reason is that MPDIoU simplifies the computation process by mining the geometric features of the horizontal rectangle, which includes all the relevant factors considered by the existing loss function: overlapping or non-overlapping regions, distance from the center point, and deviation of width and height, and at the same time, speeds up the convergence speed of the model, and thus the MPDIoU loss function is chosen to replace the original loss function in this paper.

Table 4. Comparison of Different Loss Functions

| **Loss Functions** | **P/%** | **R/%** | **mAP50/%** | **mAP50:95/%** |
|---|---|---|---|---|
| MPDIoU | 0.445 | 0.340 | 0.342 | 0.200 |
| Wiou | 0.423 | 0.328 | 0.321 | 0.183 |
| Xiou | 0.441 | 0.316 | 0.320 | 0.186 |
| Piou | 0.417 | 0.324 | 0.318 | 0.182 |
| Inner-iou | 0.429 | 0.317 | 0.312 | 0.175 |
| PolyLoss | 0.379 | 0.291 | 0.283 | 0.168 |
| VariFocalNet | 0.367 | 0.308 | 0.286 | 0.171 |

4.4.3. *Results and Analysis of ablation experiments.* In order to prove the effectiveness of the four improvements respectively, the YOLOv8n network structure is used as the original model, and eight models are designed on the basis of this model, the hyperparameter settings of the different models are all the same, and the difference is only in the structure. Among them, model 1 introduces the InceptionNeXt deep convolution module based on the original model, model 2 improves the neck based on the original model by combining the Lowlevel Feature Alignment mechanism, model 3 uses the ShareSepHead detection head based on the original model, model 4 uses the MPDIoU loss function, model 5 uses

ShareSepHead detection head based on model 1, model 6 improves the neck based on model 1, model 7 uses ShareSepHead detection head based on model 6, and model 8 uses MPDIoU loss function based on model 7. The experimental results are shown in Table 5. From model 1 to model 4, it can be seen that individually each improvement compared to the original model parameter count in different degrees of increase is not too big, the average mean accuracy has been improved, proving that each improvement has an effect. Model 5 uses the ShareSepHead detector head on top of Model 1, and the mAP50 and mAP50:95 are improved by 3.1 and 2.6 percentage points, respectively, which is a significant improvement, at the cost of a 0.0295M increase in the number of parameters in the model and a decrease in detection speed. Model 6 improves the neck based on model 1, and the mAP50 is improved by 3.1 percentage points with an increase of 1.0419M in the number of parameters, which proves that the improved backbone and neck network can better extract and fuse features, and improves the ability of the model to detect small targets. Model 7 uses the ShareSepHead detection head on top of Model 6, mAP50 and mAP50:95 are improved by 0.4 and 0.7 percentage points over Model 6, at the cost of a 0.0019M increase in the number of parameters and a decrease in detection speed. Model 8 uses the MPDIoU loss function over model 7, and mAP50 and mAP50:95 are improved by 1.0 and 0.5 percentage points, respectively, which is more suitable for application to the field of UAV small target detection.

In summary, all four improvement schemes can individually improve the model's ability to detect small targets, and the improved model can finally improve 4.5 and 3.7 percentage points for mAP50 and mAP50:95. The improved algorithm introduces each module so that the number of parameters of the model increases by nearly 1M, and the FPS decreases by about 60 frames, but it can still basically meet the requirements of the real-time UAV target detection, and it can effectively improve the task of detecting small targets on the aerial images of UAVs.

Table 5. Ablation experiment results

| **Model** | **mAP50/%** | **mAP50:95/%** | **GFLOPS** | **Params/M** | **FPS** |
|---|---|---|---|---|---|
| YOLOv8n | 31.3 | 17.6 | 8.1 | 3.0076 | 169 |
| model 1 | 32.9 | 19.1 | 8.8 | 3.0352 | 151 |
| model 2 | 33.1 | 19.3 | 12.9 | 4.0238 | 108 |
| model 3 | 34.1 | 19.8 | 13.0 | 3.2696 | 135 |
| model 4 | 34.2 | 20.0 | 8.1 | 3.0076 | 172 |
| model 5 | 34.4 | 20.2 | 8.8 | 3.0371 | 140 |
| model 6 | 34.4 | 20.1 | 13.6 | 4.0495 | 99 |
| model 7 | 34.8 | 20.8 | 13.6 | 4.0514 | 102 |
| model 8 | 35.8 | 21.3 | 13.6 | 4.0514 | 109 |

4.4.4. *Comparative Experimental Results and Analysis.* In order to verify the effectiveness and superiority of the improved algorithm in this paper, this paper compares the performance of multiple current mainstream target detection algorithms, including other algorithms in the YOLO series, on the VisDrone2019 dataset, and mainly uses mAP50 and mAP50:95 as the evaluation indexes for comparison, and the comparison results are given in Table 6. From Table 6, it can be seen that the detection accuracies of all the algorithms in this paper are higher than the rest of the algorithms, with mAP50 and mAP50:95 reaching 35.8% and 21.3%, respectively. Compared with the original YOLOv8n model, it increases of 4.5% and 3.7%, respectively. In summary, the improved algorithms show obvious advantages in detection accuracy.

Table 6. Comparison of Results of Different Algorithms

| **Algorithm** | **mAP50/%** | **mAP50:95/%** |
|---|---|---|
| SSD | 23.9 | 15.1 |
| RetinaNet | 21.37 | 14.8 |
| CenterNet | 26.2 | 16.2 |
| RefineDet | 27.6 | 16.5 |
| YOLOv4 | 29.5 | 17.1 |
| YOLOv5s | 31.0 | 17.4 |
| YOLOv6 | 30.9 | 17.5 |
| YOLOv7-tiny | 31.2 | 17.7 |
| YOLOv8n | 31.3 | 17.6 |
| Improved algorithm | 35.8 | 21.3 |

5. **Concluding Remarks.** UAV aerial image target detection has important application value in civil and military fields. Aiming at the problem of low accuracy and poor detection of small targets, this paper is based on the YOLOv8n network. By using the InceptionNeXt convolution module in the backbone network and combining it with the Lowlevel Feature Alignment mechanism to reconstruct the neck structure of the model, the paper proposes an improved YOLOv8 algorithm for small target detection in UAV aerial images. The ShareSepHead is used to detect the head, and the MPDIoU loss function is utilized. Experiments on the UAV aerial photography dataset VisDrone2019 show that the proposed algorithm outperforms the benchmark algorithm YOLOv8n all aspects and has the best overall performance compared with other advanced methods, which can meet the demand for real-time and accuracy of UAV target detection. However, the improved algorithm still has much room for improvement in target detection accuracy, especially for small target detection. In addition, it is still necessary to pay attention to the actual performance of the model deployed on the UAV in the later work.

## REFERENCES

[1] J.-H. Chen and X.-H. Wang, "Dense Small Object Detection Algorithm Based on Improved YOLOv5 in UAV Aerial Images," *Computer Engineering and Applications*, vol. 60, no. 3, pp. 100–108, 2024.

[2] X.-L. Li, D.-D. Liu, X.-M. Liu, and Z. Chen, "Target detection algorithm of UAV aerial image based on improved YOLOv5," *Computer Engineering and Applications*, vol. 60, no. 11, pp. 204–214, 2024.

[3] T.-Y. Wu, X.-L. Guo, Y.-C. Chen, S. Kumari, and C.-M. Chen, "Amassing the Security: An Enhanced Authentication Protocol for Drone Communications over 5G Networks," *Drones*, vol. 6, no. 1, pp. 1–10, 2022.

[4] T.-Y. Wu, H.-N. Li, S. Kumari, and C.-M. Chen, "A Spectral Convolutional Neural Network Model Based on Adaptive Fick's Law for Hyperspectral Image Classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19–46, 2024.

[5] F.-Q. Zhang, T.-Y. Wu, J.-S. Pan, G.-Y. Ding, and Z.-Y. Li, "Human Motion Recognition Based on SVM in VR Art Media Interaction Environment," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–40, 2019.

[6] W. Liu, D. Anguelov, and D. Erhan, "SSD: Single Shot Multibox Detector," in *Computer Vision–ECCV 2016.* Springer, 2016, pp. 21–37.

[7] J. Redmon, S. Divvala, and R. Girshick, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2016, pp. 779–788.

[8] T.-Y. Lin, P. Goyal, and R. Girshick, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision.* IEEE, 2017, pp. 2980–2988.

[9] R. Girshick, J. Donahue, and T. Darrell, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2014, pp. 580–587.

[10] F.-K. Chen and S.-X. Li, "UAV Target Detection Algorithm with Improved YOLOv5," *Computer Engineering and Applications*, vol. 59, no. 18, pp. 218–225, 2023.

[11] C.-H. Xie, J.-M. Wu, and H.-Y. Xu, "Small Object Detection Algorithm Based on Improved YOLOv5 in UAV Image," *Computer Engineering and Applications*, vol. 59, no. 9, pp. 198–206, 2024.

[12] H. Qiu, X.-Y. Zhou, L.-H. Huang, and H. Yang, "An Improved YOLOv5n Detection Algorithm for Aerial Photography of Small Targets," *Electronics Optics & Control*, vol. 30, no. 10, pp. 95–101, 2023.

[13] H.-S. Zhang, M.-W. Fan, X. Tan, Z.-J. Zhen, L.-M. Kou, and J. Xu, "Dense small object vehicle detection in UAV aerial images using improved YOLOX," *Journal of Jilin University (Engineering and Technology Edition)*, 2023. [Online]. Available: `https://doi.org/10.13229/j.cnki.jdxbgxb.20230779`

[14] S. Liang, H. Wu, and L. Zhen, "Edge YOLO: Real-time Intelligent Object Detection System Based on Edge-Cloud Cooperation in Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25345–25360, 2022.

[15] B. Wang, G. Yang, and H. Yang, "Multiscale Maize Tassel Identification Based on Improved RetinaNet Model and UAV Images," *Remote Sensing*, vol. 15, 2530, 2023.

[16] J.-D. Zhao, G.-Y. Zhen, and C.-Q. Chu, "Drone Image Target Detection Algorithm Based on YOLOv8," *Computer Engineering*, vol. 50, no. 4, pp. 113–120, 2024.

[17] T.-Y. Lin, P. Dollár, and R. Girshick, "Feature Pyramid Networks for Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2017, pp. 2117–2125.

[18] S. Liu, L. Qi, and H. Qin, "Path Aggregation Network for Instance Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 2018, pp. 8759–8768.

[19] W. Yu, P. Zhou, and S. Yan, "InceptionNeXt: When Inception Meets ConvNeXt," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* IEEE, 2024, pp. 5672–5683.

[20] R. Azad, L. Niggemeier, and M. Hüttemann, "Beyond Self-Attention: Deformable Large Kernel Attention for Medical Image Segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* IEEE, 2024, pp. 1287–1297.

[21] C.-W. Liu, Zhang, and H. Huang, "RTMDet: An Empirical Study of Designing Real-Time Object Detectors," *arXiv preprint* arXiv:2212.07784, 2022.

[22] Z. Zheng, P. Wang, and D. Ren, "Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 8574–8586, 2021.

[23] S.-L. Ma and X. Yong, "MPDIoU: A Loss for Efficient and Accurate Bounding Box Regression," *arXiv preprint* arXiv:2307.07662, 2023.

[24] D. Du, P. Zhu, and L. Wen, "VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results," in *Proceedings of the IEEE/CVF International Conference on Computer Vision.* IEEE, 2019, pp. 213–226.