

An Improved YOLOv8 Network for Real-Time Air-to-Ground Object Detection

Bo Peng¹, Chen Dong^{1*}

¹School of Computer Science and Engineering, Tianjin University of Technology, Tianjin
300380, CHINA
dongc@tjut.edu.cn

Abstract. Real-time target detection algorithms play a crucial role in the field of UAV air-to-ground detection. The operational environment of UAVs necessitates the rapid response capabilities of object detection algorithms. In response to the issues of slow processing speed and high latency encountered by traditional object detection algorithms in rapid response scenarios, this paper introduces an efficient air-to-ground real-time detection algorithm based on YOLOv8. Firstly, to improve the detection speed of the YOLOv8 model, this paper uses a partial convolution technique to improve the C2F module of the model. In order to achieve fast arithmetic while improving accuracy, the SimAM parameter-free attention module is introduced, realizing a lightweight design and high precision for the module. The model is trained and tested using the VisDrone dataset to validate the efficiency and accuracy of the model for the UAV air-to-ground target detection task. The improved YOLOv8 model reduces the number of parameters by 30.56%, GFLOPs by 20.65%, and air-to-ground target detection accuracy by 2.1% compared to the YOLOv8 model, which greatly reduces the computational complexity of the model on the basis of improving the detection accuracy, and realizes a fast and efficient lightweight neural network for air-to-ground target detection task.

Keywords: Air-to-Ground Object Detection; Partial Convolution; YOLOv8; SimAM;

1 Introduction

In recent years, significant progress has been made in the field of air-to-ground real-time surveillance research. The development of UAV air-to-ground real-time monitoring technology stems from the need for fast, efficient and low-cost monitoring means. With the maturation of UAV technology and the reduction of costs, the application of UAVs in various fields has gradually increased. UAVs have the advantages of flexibility, rapid deployment and cost-effectiveness, and therefore have a wide range of potential applications in areas such as agricultural monitoring, disaster response, environmental monitoring, wildlife protection and security monitoring. However, the computational resources on drone platforms are limited, which places high demands on the speed and accuracy of object detection algorithms. Traditional object detection

algorithms often have high computational complexity, making it difficult to meet the real-time processing requirements of drones [1-3].

In order to improve detection speed and accuracy for rapid object detection in real-time scenarios, many optimization strategies have been proposed. Tan et al. (2020) proposed the EfficientDet model, which uses EfficientNet as the backbone network and integrates BiFPN (Bi-directional Feature Pyramid Network) to enhance feature fusion efficiency. While maintaining high accuracy, EfficientDet also excels in terms of speed and model size. This approach optimizes the network architecture and improves the model's efficiency [4]. A significant limitation of EfficientDet is its performance in detecting small objects. Due to the downsampling operation and the focus on reducing the size of the model, the model may struggle to accurately detect small objects in high-resolution images. This is a significant drawback in applications such as aerial surveillance and autonomous driving. For object detection algorithms running on drone terminals, Howard et al. proposed MobileNet. By introducing depthwise separable convolutions, which decompose standard convolutions into depthwise and pointwise convolutions, the model's parameters and computations are significantly reduced. This makes the model more lightweight and suitable for mobile and embedded devices such as drones [5-7]. A significant limitation of MobileNet is its reduced accuracy compared to larger, more complex models. While lightweight architectures greatly improve speed and efficiency, the cost is often reduced accuracy, especially in complex scenes with many overlapping objects or changing lighting conditions.

In air-to-ground target detection, to improve detection accuracy. Fu et al. proposed DANet, which combines spatial attention and channel attention. By using parallel spatial attention and channel attention modules, DANet captures more useful feature information, thereby improving detection accuracy. This network has demonstrated superior performance in both semantic segmentation and object detection tasks [8].

One notable limitation of DANet is its impact on processing speed. The parallel attention modules introduce significant computational overhead, which slows down inference speed. The increased complexity and resource consumption make DANet less suitable for real-time applications and deployment on resource-constrained devices, such as drones, where rapid processing and efficiency are crucial. Woo et al. proposed CBAM [9]. CBAM is a lightweight attention module that combines channel attention and spatial attention. It first selects features through the channel attention module, and then refines the feature map through the spatial attention module, thereby enhancing detection accuracy. A notable limitation of CBAM is its impact on processing speed. Although CBAM is considered lightweight compared to other attention mechanisms, the additional computation introduced by the channel and spatial attention modules still reduces the overall inference speed. This is a significant drawback in real-time UAV target detection applications where fast processing is critical. YOLOv8 is an efficient real-time object detection model that predicts multiple bounding boxes and class probabilities in images using a single neural network. It balances high speed and accuracy, making it ideal for applications needing quick responses. However, prioritizing speed and efficiency may lead to reduced accuracy compared to more complex models like Faster R-CNN or SSD. YOLOv8's network architecture and feature pyramid structure

may limit its ability to detect and localize small objects accurately, impacting tasks such as aerial surveillance of small targets like individuals or small vehicles [10].

On the basis of these advances and in view of its limitations, this paper proposes an improved YOLOv8 network for real-time air-to-ground object detection. Firstly, partial convolution is introduced into the C2F module of YOLOv8 network for optimization, which can simultaneously reduce the computational redundancy and memory access of the original model. It selects the characteristics of some channels for regular convolution, and keeps the characteristics of the remaining channels unchanged, which reduces the computational complexity, thus realizing a fast and efficient neural network. At the same time, in order to improve the accuracy of air-to-ground target detection and at the same time ensure the running speed of the algorithm, this paper introduces the SimAM parameter-free attention module, which derives 3D attention weights for the feature maps on the basis of not adding additional parameters. This reduces the computational complexity of the model while improving the detection accuracy, thus providing a fast and efficient lightweight neural network for air-to-ground target detection tasks.

2 Related work

A. Optimization models for convolutional neural networks

Convolutional Neural Networks (CNNs) are the mainstream architecture in the field of computer vision, especially when practical deployment requires a balance between speed and accuracy. Many studies aim to improve the efficiency of CNNs. Popular methods include group convolutions and depthwise separable convolutions, which split standard convolutions into depthwise and pointwise convolutions to reduce the number of parameters and FLOPs [11].

Group convolution has been widely used to improve the efficiency of convolutional neural networks. In AlexNet's architecture, group convolution is used to distribute model parameters across two GPUs, allowing for parallel processing and reducing computational burden. The ResNeXt model utilizes group convolution to build a highly modular architecture that effectively balances model complexity and performance. By dividing the convolution into smaller groups, ResNeXt can capture multiple features without significantly increasing computational cost, making it suitable for scalable and efficient deep learning applications. Similarly, MobileNets uses deeply separable convolutions to achieve a lightweight model architecture that is well suited for mobile and embedded applications. This approach decomposes the convolution operation into two steps: deep convolution and pointwise convolution. This separation greatly reduces computational complexity and helps achieve efficient inference on resource-limited devices [12].

While these methods reduce filter redundancy, they increase memory access requirements when expanding the network width to maintain accuracy. In contrast, partial convolution applies convolution operations to some of the channels of the input feature map while leaving the other channels unchanged, while reducing computational redundancy and memory accesses, improving overall efficiency.

B. Attention mechanisms in object detection

Attention mechanisms have been widely integrated into object detection networks to enhance feature representation and improve detection accuracy. Two prominent examples are the Dual Attention Network (DANet) and the Convolutional Block Attention Module (CBAM).

DANet combines spatial attention and channel attention through parallel modules, enhancing feature information and improving detection accuracy. Despite its strong performance in semantic segmentation and object detection, the parallel modules in DANet introduce significant computational overhead, reducing inference speed and making it less suitable for real-time applications and resource-constrained devices like drones.

CBAM is a lightweight module that combines channel and spatial attention to refine feature maps and improve detection accuracy. However, CBAM still introduces additional computations, reducing overall inference speed. This drawback is particularly significant for real-time UAV target detection applications that require rapid processing.

Compared to the attention mechanisms of DANet and CBAM, SimAM offers several distinct advantages. Firstly, SimAM features a parameter-free design, eliminating the need for additional parameters that increase complexity and computational load, unlike the parameter-heavy DANet and CBAM. Secondly, SimAM is more computationally efficient, optimizing the calculation process to reduce computational complexity and memory access, thereby improving processing speed. This contrasts with the significant computational overhead introduced by DANet's parallel modules and the additional computations required by CBAM's channel and spatial attention modules. Finally, SimAM strikes a better balance between accuracy and efficiency, enhancing target detection precision without compromising speed, making it ideal for real-time applications such as UAV air-to-ground surveillance.

3 Method

A. Implementation of partial convolution

Partial Convolution (PConv) is designed to optimize the computational cost of convolutional operations by leveraging the redundancy in feature maps [13], [14]. The approach involves applying convolution operations to only a portion of the input channels while keeping the remaining channels unchanged. This method reduces computational redundancy and improves efficiency without compromising the integrity of the feature extraction process. Select a subset of the input channels for the convolution operation. For efficient memory access, the first or last continuous channels are typically chosen as representatives of the entire feature map. Apply standard convolution (Conv) to the selected subset of channels to extract spatial features. For the remaining channels, bypass the convolution operation, and keep them unchanged. Combine the convolved channels with the unchanged channels to form the output feature map. This ensures that

critical spatial information is captured while reducing the computational load. Fig. 1 illustrates the implementation of partial convolution.

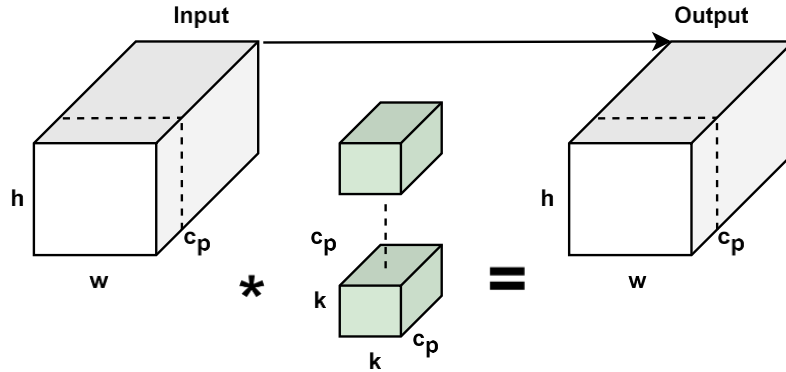


Fig 1. The process of partial convolution.

Let h and w represent the height and width of the input feature map, determining the spatial extent over which convolution is applied. The parameter k denotes the kernel size, influencing the receptive field size and affecting feature extraction. Additionally, c_p signifies the number of channels selected for convolution. In general, assuming the input and output feature maps have the same number of channels, the FLOPs of PConv are:

$$h \times w \times k^2 \times c_p^2 \quad (1)$$

and for a typical case where $r = \frac{1}{4}$, PConv's FLOPs are only $\frac{1}{16}$ of those of regular Conv. Similarly, for the memory access of PConv, which are computed as follows:

$$h \times w \times 2c_p + k^2 \times c_p^2 \quad (2)$$

Under typical conditions where $r = \frac{1}{4}$, PConv is only $\frac{1}{4}$ of the regular case. PConv greatly reduces the need for reduced computational redundancy and memory access, advantages that make PConv particularly suitable for application scenarios with limited computational resources, such as real-time applications on UAV devices.

B. Architecture of PCnet module

This paper introduces PCnet to optimize the C2f module in the original YOLOv8. The PCnet module structure consists of a PConv layer and two PWConv layers (1×1 Conv). Together, they form an inverted residual block, with the middle layer having an expanded number of channels, and a Shortcut set to reuse input features. Normalization and activation layers are placed after each intermediate PWConv to maintain feature diversity and achieve lower latency. Additionally, batch normalization (BN) is used, which has the advantage of integrating into adjacent Conv layers to speed up inference while maintaining the same efficiency as other layers. For activation layers, GELU is chosen for smaller PCnet variants based on experience, while ReLU is used for larger PCnet variants, considering both runtime and effectiveness. The last three layers, namely global average pooling, 1×1 convolution, and fully connected layers, are used

together for feature transformation and classification. The specific structure of PCnet is shown in Fig. 2.

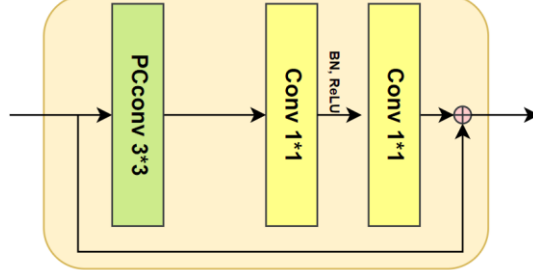


Fig 2. Structure of the PCnet module.

C. SimAM attention mechanism

In UAV (Unmanned Aerial Vehicle) air-to-ground recognition, maintaining high-speed processing and computational efficiency is crucial due to the limited resources available on drone platforms. The need for rapid and accurate detection of ground targets, such as vehicles or individuals, necessitates the use of lightweight and efficient neural network modules. SimAM computes 3D attention weights that capture spatial and channel-wise dependencies, enhancing feature representation and improving the detection accuracy of ground objects. This is particularly beneficial for distinguishing between similar-looking objects or detecting small and distant targets. SimAM is a parameter-free attention mechanism, meaning it does not introduce additional parameters that would increase the model's computational complexity. This is advantageous for applications where computational resources and battery life are limited, such as in UAVs. The simplicity of SimAM allows for the fast computation of attention weights, reducing the latency associated with more complex attention mechanisms like CBAM and DANet. This ensures that UAVs can process data in real-time, which is essential for timely decision-making and actions. The structure of SimAM attention is shown in Fig 3.

SimAM's core idea is based on the local self-similarity of images. In images, adjacent pixels typically exhibit strong similarity, while pixels that are further apart show weaker similarity. SimAM leverages this property by using Euclidean distance to calculate the similarity between each pixel and its neighboring pixels in the feature map, thereby generating attention weights. The similarity $s(f_i, f_j)$ between the i^{th} pixel f_i and the j^{th} pixel f_j is calculated as follows:

$$s(f_i, f_j) = -\|f_i - f_j\|_2^2 \quad (3)$$

In the SimAM attention mechanism, each pixel is assigned an independent weight, and the attention weight w_i for the i th pixel is calculated as follows:

$$w_i = \frac{1}{k} \sum_{j \in N_i} s(f_i, f_j) \quad (4)$$

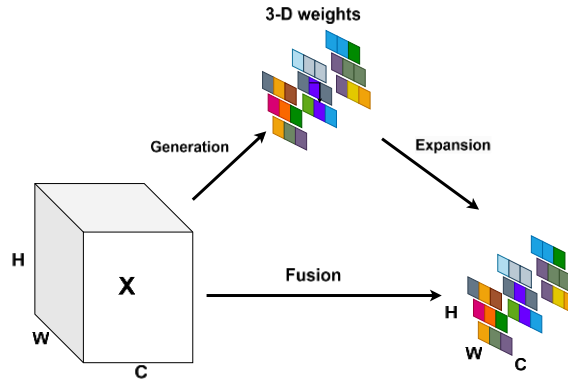


Fig 3. Structure of the SimAM attention.

D. Improvements to the YOLOv8 network

The YOLOv8 network structure is composed of three main parts: the backbone, neck, and head. The backbone is responsible for feature extraction from ground images. At the end of the backbone network, the SimAM attention mechanism is employed. This allows the attention mechanism to observe the entire feature map produced by the backbone, providing a global perspective. SimAM is a parameter-free model, which means that performing attention mechanism calculations at the end of the backbone module has minimal impact on processing speed. Secondly, the original C2f module in the model is replaced with the PCnetC2f module. This replacement reduces computational redundancy and improves processing efficiency by selectively applying convolution operations to a subset of channels while keeping others unchanged. This results in faster and more efficient spatial feature extraction, making the network more suitable for real-time applications. Additionally, the integration of the SimAM attention mechanism ensures that the optimized YOLOv8 network maintains or even improves accuracy while achieving faster real-time air-to-ground target detection. This balance of speed and precision makes it well-suited for high-speed UAV operations. The improved YOLOv8 network structure is shown in Fig 4.

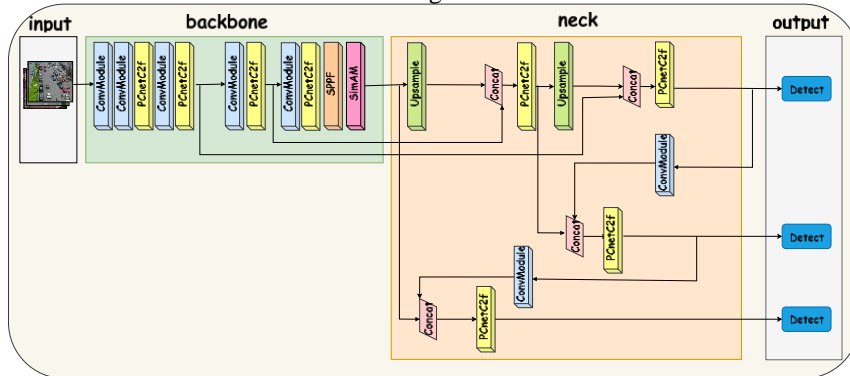


Fig 4. Improved YOLOv8 network architecture.

4 Experiments

A. Benchmarks

In this paper's experiments, the VisDrone Dataset was the primary dataset [15]. Widely used in image processing, it includes 288 video clips (261,908 frames) and 10,209 still images, captured by UAV-mounted cameras across diverse locations and scenes. Data collection occurred under varied scenarios, weather conditions, and lighting with multiple drone platforms. Each frame is annotated with bounding boxes for over 2.6 million targets, including pedestrians, cars, bicycles, and tricycles, with additional attributes like scene visibility, object class, and occlusion to enhance its utility for air-to-ground object detection tasks. A sample Visdrone dataset is shown in Fig 5.

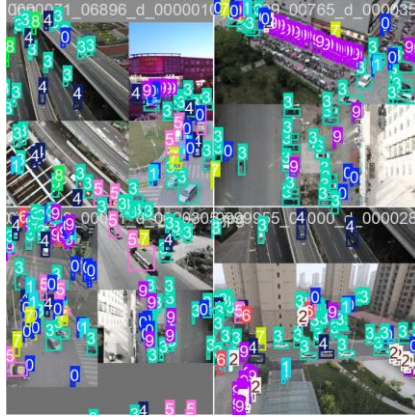


Fig 5. Example of data from the visdrone dataset.

B. Experiment setup

To establish a benchmark for comparison, this paper first used the original YOLOv8n weights as a starting point. The network was then trained on the Visdrone dataset, a dataset specialized for air-to-ground object detection tasks. The results of the training are shown in Fig 6.

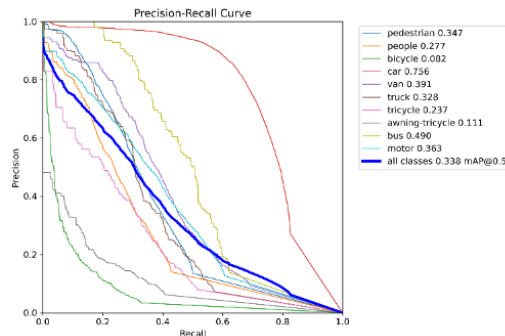


Fig 6. The results of the training.

The model performs extremely well for cars and trucks, achieving high average accuracies of 0.756 and 0.928, respectively. The detection of pedestrians, buses and motorbikes also has good results with 0.347, 0.469 and 0.353 respectively. the detection of small targets such as bicycles and pushchairs is weak, suggesting that further optimisation is required for the detection of these categories.

C. Comparison with YOLOv8 model

This paper compared the Improved YOLOv8 network with the YOLOv8 model across metrics including precision, recall, mAP50, mAP50-95, and total processing time, as shown in Table 1. The Improved YOLOv8 model outperforms the YOLOv8 model in precision and mAP50 metrics, with particularly significant improvements in mAP50-95. In terms of processing speed, the Improved YOLOv8 model demonstrates a noticeable increase over the YOLOv8 model. These metric improvements demonstrate the Improved YOLOv8 model's leadership in both detection accuracy and operational speed for UAV air-to-ground visual target detection tasks.

The experimental results indicate that the proposed Improved YOLOv8 model not only enhances processing speed but also improves detection accuracy, leading in most metrics. This underscores the Improved YOLOv8 model's superiority in both speed and accuracy for visual target detection tasks, particularly suitable for efficient handling of UAV air-to-ground target detection tasks.

Table 1. Comparison with YOLOv8 model

Model	Improved YOLOv8		YOLOv8	
Metric	Mean	Median	Mean	Mean
Precision	0.422	0.423	0.420	0.421
Recall	0.320	0.337	0.322	0.334
mAP50	0.319	0.331	0.317	0.333
mAP50-95	0.185	0.198	0.183	0.194
Total Time	8280s		8560s	

This paper conducted a comparison of the parameters, gradients, and GFLOPs between the Improved YOLOv8 model and its YOLOv8 counterpart, as illustrated in Fig 7. The Improved YOLOv8 model demonstrates a significant reduction in parameters, gradients, and GFLOPs, achieving a reduction of 30.56% in parameters and 20.65% in GFLOPs compared to the YOLOv8 model. This optimization results in a more lightweight network architecture, facilitating faster training and inference times due to reduced parameter and gradient overheads. Moreover, the reduction in GFLOPs contributes to lower computational requirements and energy consumption.

In the context of UAV air-to-ground target detection, these improvements are particularly advantageous. The streamlined architecture of the Improved YOLOv8 model enhances computational efficiency without compromising detection accuracy. This makes it well-suited for real-time applications on UAV platforms, where efficient

utilization of computational resources and energy conservation are crucial for prolonged operation and effective mission execution.

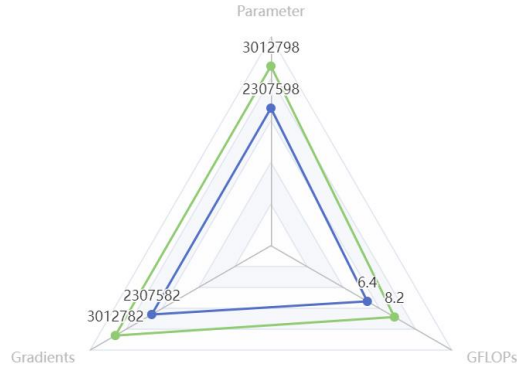


Fig 7. Comparative Analysis of Parameter Count Between Two Models (Blue: Improved YOLOv8, Green: YOLOv8).

D. Results

This paper conducted multifaceted experiments on the Visdrone test set to examine the performance of the model in different environments. For UAV air-to-ground target detection, the ability to recognize and locate small targets greatly affects the accuracy of target detection. This paper selected small target data samples of street pedestrians to test the improved YOLOv8 model, examining its capability in small target recognition. The improved YOLOv8 model accurately identified street pedestrians and other small target objects, demonstrating its capability in recognizing small targets commonly encountered in air-to-ground recognition tasks. The detection results of the model are shown in Fig 8.

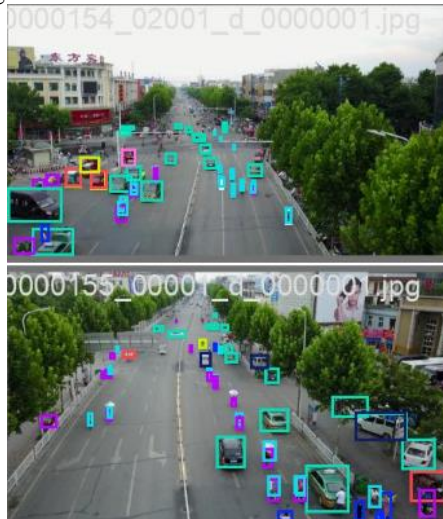


Fig 8. Improved YOLOv8's small target recognition detection results.

Specifically, the model exhibited remarkable efficacy in identifying targets with varying heights, angles, and amidst various environmental conditions, including both urban and rural settings. The results of the validation experiments for the Improved YOLOv8 model are presented in Fig 9. The outcomes of these experiments showcased the robustness and adaptability of the model in accurately detecting objects across a diverse spectrum of scenarios.



Fig 9. The results of Improved YOLOv8 on the different test sets.

5 Conclusion and future work

In the context of air-to-ground target detection tasks using unmanned aerial vehicles (UAVs), computational resources are paramount. The demands on computational resources for target detection algorithms are high. This paper introduces an improved YOLOv8 algorithm that optimizes the C2F module using partial convolution. This optimization reduces the model's parameter count by 30.56% and GFLOPs by 20.65%, significantly reducing memory usage and enhancing detection speed. These advancements enable the model to operate efficiently on UAV terminals requiring high-speed processing. Furthermore, this paper incorporates the SimAM attention module into the backbone of the model, allowing the attention mechanism to encompass the entire feature map generated by the backbone. This integration provides a global perspective, thereby enhancing the accuracy of object detection tasks. SimAM operates as a parameter-free model, meaning that computing the attention mechanism at the end of the backbone module has minimal impact on processing speed and requires no additional memory space. This aspect is particularly advantageous for UAV terminal devices with limited computational resources. On the VisDrone aerial dataset, the Improved YOLOv8 model demonstrates significantly faster processing speeds and higher detection accuracy compared to the YOLOv8 model. It also shows remarkable effectiveness in identifying targets at different heights, angles and environmental conditions, proving the generalization ability and robustness of the model. There are still some problems in this research. The accuracy of small target detection is not satisfactory enough, and in

the future, targeted optimisation for small target detection can be carried out, such as increasing the small target detection head and strengthening the feature extraction capability. So that the model can achieve better results in air-to-ground target detection.

References

1. Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3), 257-276.
2. Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), 3212-3232.
3. Hu, H., Gu, J., Zhang, Z., Dai, J., & Wei, Y. (2018). Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3588-3597).
4. Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10781-10790).
5. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
6. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).
7. Howard, A., Sandler, M., Chu, G., Chen, L. C., Chen, B., Tan, M., ... & Adam, H. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1314-1324).
8. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3146-3154).
9. Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3-19).
10. Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO (Version 8.0.0) [Computer software]. <https://github.com/ultralytics/ultralytics>.
11. Khan, S., Rahmani, H., Shah, S. A. A., Bennamoun, M., Medioni, G., & Dickinson, S. (2018). *A guide to convolutional neural networks for computer vision*.
12. Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492-1500).
13. Chen, J., Kao, S. H., He, H., Zhuo, W., Wen, S., Lee, C. H., & Chan, S. H. G. (2023). Run, don't walk: chasing higher FLOPS for faster neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12021-12031).
14. Liu, G., Reda, F. A., Shih, K. J., Wang, T. C., Tao, A., & Catanzaro, B. (2018). Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 85-100).
15. Cao, Y., He, Z., Wang, L., Wang, W., Yuan, Y., Zhang, D., ... & Liu, M. (2021). VisDrone-DET2021: The vision meets drone object detection challenge results. In *Proceedings of the IEEE/CVF International conference on computer vision* (pp. 2847-2854).