YOLO-Rep: Lightweight UAV Visual Tracking Algorithm

Weiqi Yi[1,2], Jin Cui[1*], Yaqiang Sun[2], Mei Yuan[1],Fanshu Zhao[1]

[1]School of Automation Science and Electrical Engineering,Beihang University, Beijing, 100191, China
[2]Institute of Automation, Chinese Academy of Sciences, Beijing,100190, China
**jincui@buaa.edu.cn (Jin Cui)*

Keywords: DRONE TRACKING AND DETECTION, YOLO, MOBILENETV3, MULTI-BRANCH STRUCTURE, STRUCTURE REPARAMETERIZATION

Abstract

In order to attain a more optimal trade-off between detection accuracy and detection speed of existing UAV tracking and detection algorithms, this paper designs a lightweight UAV tracking and detection algorithm YOLO-Rep based on the improved YOLO (You Only Look Once).The algorithm firstly takes the multi-scale fusion layer and the output detection layer of the YOLO as the neck network and the head network, respectively. The MobileNetV3 lightweight CNN is introduced to reconstruct the backbone network, and the MetaFormer architecture of lightweight ViTs (Vision in Transformer) is applied to split the token mixer and channel mixer of MobileNetV3 to achieve better lightweighting. Secondly, the block structure in the training backbone network is optimised to a multi-branch structure, which enables the kernel computation to run in parallel and improves the speed of model training. Finally, to simplify the inference model and further reduce the time used for model inference, the inference model is reconstructed using Structural Re-parameterization, which converts it to a single, more efficient convolutional operation during inference, improving the performance and training stability of the network. The model offers the benefits of both high detection accuracy and rapid processing speed, which is a great improvement over the original algorithm, and has a greater advantage in the UAV detection tracking and detection algorithm.

1 Introduction

As the demand for military applications and the improvement of UAV performance and technology, UAVs are increasingly being used in both civilian and military applications [1]. As single UAVs cannot meet the demands of complex missions due to their low load, small size and small coverage area, research on cooperative formation control of UAV clusters has been developing rapidly in recent years. Compared with single UAVs, multi-UAV clusters are able to comprehensively use multiple sensors to obtain information about the surrounding environment, and then use data fusion technology to obtain higher accuracy, which can provide more adequate preparation for cooperative operations in various scenarios [2]. Distributed obstacle avoidance strategy is commonly used in multi-UAV cluster cooperative flight, which is a kind of strategy in which the long aircraft is responsible for obstacle avoidance and the wingman follows. Although each UAV can sense the surrounding environment independently, the long aircraft senses the environmental obstacles and carries out obstacle avoidance independently, and the wingman senses the position of the long aircraft, and follows the long aircraft to avoid collision and maintain the formation. In practical applications, high-speed UAV clusters need timely response, so it holds substantial research significance to optimise the detection rate and accuracy of the single target detection and tracking algorithm for wingmen.

1.1 YOLOv8 network model

YOLOv8 [3] is the cutting-edge SOTA model released by ultralytics on the 10th of January 2023, building on the achievements of earlier YOLO releases and incorporating new features and enhancements that now support tasks such as target classification, target detection, and image segmentation. Compared to the previously commonly used YOLOv5 [4], YOLOv8 is based on a comprehensive enhancement of the improved YOLOv5 model structure, while maintaining the engineering simplicity and ease-of-use advantages of YOLOv5.

YOLOv8, like the previous YOLO series, is categorized into five magnitudes of models depending on the quantity of parameters in the backbone network: n, s, m, l, and x. YOLOv8 can be used as a model for the backbone network by setting the depth factor (d), width factor (w), and scale factor (r) to adjust the parametric quantities. The architecture of YOLOv8 comprises Backbone, Neck, and Head components. Among them, the Backbone primarily serves for feature extraction, using efficient network structures such as CSPDarknet, EfficientNet, etc. to extract image features of different sizes. Neck is the feature fusion module, which is commonly utilized for feature fusion extracted from images of different sizes with FPN (Feature Pyramid Network) and PAN (Path Aggregation Network). Head is the final prediction module, responsible for generating bounding box, category probability, etc., which is the most suitable for YOLOv8 [5][6]. It is responsible for generating bounding boxes, category probabilities, etc. and outputting the prediction results.

For the Backbone layer, the developer still adopts the idea of CSP in YOLOv5 [7], and YOLOv8 designs the C2f structure with reference to the residual structure of the C3 module and the ELAN idea of YOLOv7 [8], while improving the convolutional kernel, more adjusting layer connections and split operations are added, which can obtain more comprehensive information while guaranteeing lightweight

and adjust the quantity of channels according to the model scale, which significantly improves the model performance [9].

For the Neck layer, its feature fusion module borrows the idea of path aggregation network. Compared with YOLOv5, it omits the convolutional layer in the sampling stage on the PAN-FPN, and this adjustment aims to streamline the model structure and decrease the computational complexity. Meanwhile, YOLOv8 also introduces the more efficient C2f module, replacing the previous C3 module in this part, which further optimises the efficiency and effectiveness of feature processing.

For the Head layer, YOLOv8 introduces a significant improvement: the parameters of the classification and regression tasks are completely independent of each other, a change that helps to more accurately adapt the model to different task requirements, thus improving the overall detection performance. In addition, YOLOv8 abandons the traditional Anchor-Based approach in favour of an Anchor-Free strategy, which simplifies the model structure and diminishes the model's dependence on prior knowledge, making it more flexible and versatile. All these innovations are aimed at improving the adaptability and accuracy of the models in various environments.

1.2 MobileNet network model

MobileNet，a lightweight CNN proposed by the Google team that utilises deeply separable convolution to reduce the model's FlOPs, has been updated to MobileV4. MobileNetV1 [10] was introduced in 2017 and references the deep separable convolution module for the first time. MobileNetV2 [11] introduced the linear bottleneck and inverted residual structure on top of MobileNetV1, further diminishing the amount of parameters in the model and enhancing the precision. MobileNetV3 [12] carries forward the depth-separable convolution from V1 and the linear bottleneck residual structure from V2. It employs the NetAdapt algorithm to ascertain the optimal quantity of convolution kernels and channels, incorporates the SE (Squeeze-and-Excitation) channel-attention mechanism, and replaces ReLU6 with the new activation function h-swish(x). MobileNetV4 [13] was launched in April 2024, and the authors used a two-stage neural network architecture search (NAS) approach to separate coarse and fine-grained searches, which significantly improved the search efficiency, in addition to combining offline distilled datasets, which reduced noise in the NAS reward measurements, thus improving the model quality. As V4 pursues extreme optimisation for specific tasks and is not as effective as V3 in terms of applicability, MobileNetV3 was chosen for further model optimisation.

Depth-separable convolution is the main feature of MobileNetV3 and the key to play its lightweight role, which can be separated into two steps, depthwise convolution (DW) and point-by-point convolution (PW). The quantity of convolution kernels in depthwise convolution matches the input image's number of channels, and the channels and convolution kernels are one-to-one, so the quantity of feature maps of depthwise convolution output is equivalent to the quantity of channels in the input image. point-by-point convolution inputs the feature maps output from depthwise convolution into the convolution kernel of 1×1 for convolution, so that each feature map of the output contains the information of all the feature maps of the input layer.

2. Modular design

2.1 C2fM structure design

YOLOv8, as an efficient target detection model, especially the C2f module obtained by improving the C3 module based on YOLOv5, further enhances the target detection accuracy and rate, and its deployment on mobile devices and embedded systems may face the challenge of limited computational resources, especially for UAVs, which are mobile devices with very demanding requirements on detection accuracy and speed, and it is necessary to do further algorithm optimisation. MobileNetv3 is an efficient CNN specifically intended for mobile devices and embedded systems. By using MobileNetv3 to optimise the C2f module of YOLOv8, the model's inference rate can be improved while preserving better detection accuracy, thus enabling real-time target detection on mobile devices and embedded systems.

MobileNetv3 adopts the SE attention mechanism in the Bottleneck structure and applies it to the Bottleneck structure of C2f, as is shown in Fig. 1.

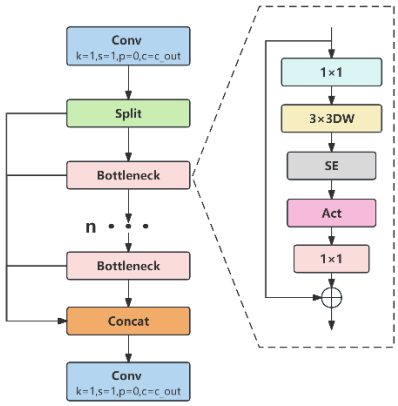

Fig. 1 Structure of C2fM

The Bottleneck block structure in C2fM uses a 1×1 extended convolution, and a 1×1 projection convolution to achieve inter-channel interaction, which is designed to diminish the quantity of channels by expanding and decreasing the quantity of channels, while still effectively enhancing the feature representation while reducing the parameters. A 3×3 depth convolution is configured after the 1×1 extended convolution for information fusion. The SE attention is configured after the 3×3 deep convolution, where the global features are first extracted by compressing the feature information via the pooling layer, and then the global features are assigned weight coefficients to each channel through two fully connected layers, so that the model is more focussed on the channels with high weights and the training accuracy is improved. Subsequently, the h-swish activation function is used and its expression is shown in equation(1).

$$h-swish[x] = x\frac{ReLU6(x+3)}{6} \quad (1)$$

Using the nonlinear swish function instead of the ReLU function can remarkably enhance the precision of the neural network. However, the swish function is not suitable for use on mobile devices due to its large computational effort. To address this issue, the h-swish function uses a segmented linear hard simulation to approximate the sigmoid in place of swish. This approach effectively reduces the computational complexity and allows the neural network model to be better deployed on mobile, as is shown in Fig. 2.

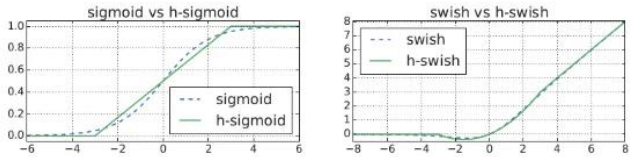

Fig. 2 Comparison plot of activation functions

2.2 C2fM_Rep structure design

By studying the lightweight ViT [14] (Vision Transformer), it was found that the block structure of the best lightweight ViTs contains a key design attribute, specifically, separate token mixer and channel mixer. Based on a current study [15], researchers have demonstrated that the effective parts of ViTs mainly stems stem from their separate token mixer and channel mixer architecture, that is the MetaFormer architecture, rather than being outfitted with a specific token mixer. Based on these studies, in order to enable further lightweighting of C2fM, this study simulates the effectiveness of existing ViTs by setting up C2fM with separate token mixer and channel mixer to mimic the existing MetaFormer architecture for lightweight ViT.

For the Bottleneck block structure in Figure 1, the 1×1 extended convolution, 3×3 deep convolution, and 1×1 projected convolution are designed so that the token mixer and channel mixer are coupled together. Based on the research [16]. in order to keep them separate and run independently to get the MetaFormer structure, the depth convolution is firstly shifted forward, and the SE attention structure is shifted forward with it, as it depends on the spatial information interaction. To further improve the performance, the 3×3 depth convolution is optimised using a multi-branch topology, i.e., adding residual connections and 1×1 depth convolution. The role of 1x1 depth convolution is to perform feature reallocation and scaling, which is equivalent to a bottleneck layer that effectively reduces the amount of computation and preserves the spatial information in the features by combining it with depth convolution. By combining this with the 3x3 deep convolution, the model is able to further optimise the feature representation, leveraging the characteristics of convolution to optimize the expressiveness of the network, as is shown in Fig. 3.

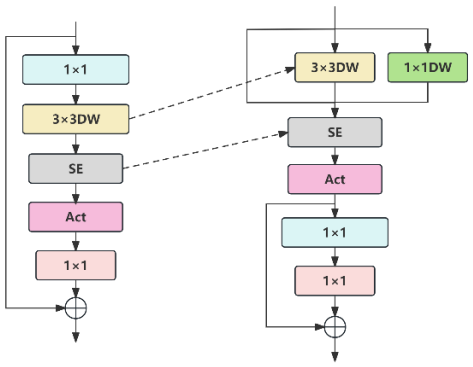

Fig. 3 MetaFormer structure optimisation for C2fM

Multi-branch topology improves accuracy during training, while saving memory cost by reducing extra computation during inference is more important. A study [17], proposes a method to decouple the training time and inference time structure by structural reparameterization technique, which ensures the high accuracy of the training model while allowing the inference model to have a faster speed with a good accuracy and speed trade-off.

The structural reparameterization technique is utilised to optimise the inference model by converting the multi-branch architecture applied while training to a unified convolutional structure during inference. During training, each C2fM block contains a 3×3 deep convolution, a 1×1 deep convolution, and a residual branch paired with a Batch Normalisation (BN) layer including mean, standard deviation, learning scale factor, and bias, respectively. In the inference process, by reparametrizing the multi-branch architecture during training, the original multi-branches are combined into a unified structure with a 3×3 convolution, and the parameters of the BN layer are combined with each convolutional layer to directly compute the final 3×3 convolutional weights and biases, to simplify the computation during inference. The specific steps are shown in Fig. 4.

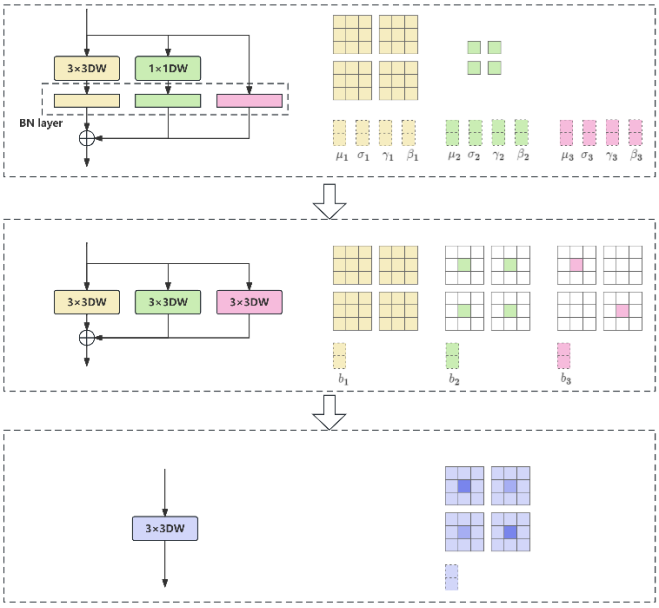

Fig. 4 Flowchart of structure reparameterization

For the initial deep convolution plus BN layer, the output is shown in equation (2):

$$y_i(\mu,\sigma,\gamma,\beta) = \frac{x_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} \cdot \gamma_i + \beta_i \tag{2}$$

The BN layer corresponding to each convolutional layer machine is first transformed into a convolutional layer with bias terms. Let $M^{(1)}$,$M^{(2)}$be the outputs and outputs, the value of the bias term can be obtained as shown in equation (3):

$$b_n(M,\mu,\sigma,\gamma,\beta)_{:,i,:,:} = \left(M_{:,i,:,:} - \mu_i\right)\frac{\gamma_i}{\sigma_i} + \beta_i \tag{3}$$

Afterwards, the weights and biases of the 1×1 convolution and residual branch are mapped into the 3×3 convolution, and finally a reparameterised 3×3 convolution weights and biases are obtained, and the resulting inference model structure, C2fM_Rep, has an input-output relationship as shown in equation (4):

$$O = I \otimes (K_1 + K_2 + K_3) + (b_1 + b_2 + b_3) \tag{4}$$

The final training model as well as the inference model is obtained in Fig. 5 shows.

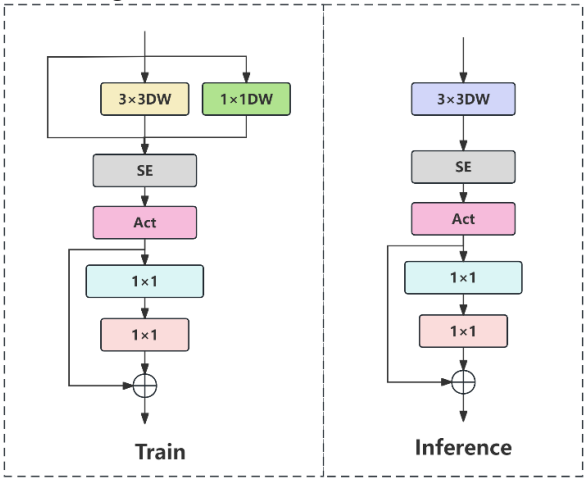

Fig. 5 C2fM_Rep train model and inference model

Applying this model to the YOLOv8 network, the improved YOLOv8 model is obtained as shown in Fig. 6.

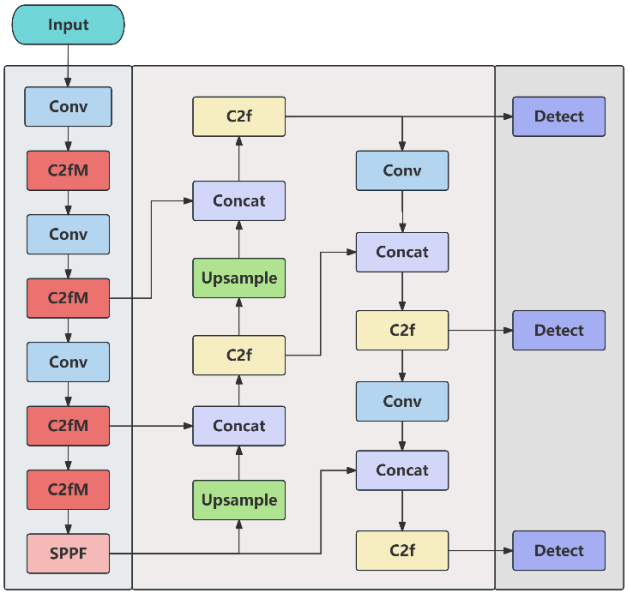

Fig. 6 Improved YOLOv8 model

3 Experiments and analysis of results

3.1 Data set

For the specific UAV used in the project to carry out the detection task, to improve the detection accuracy as much as possible, therefore, this paper adopts a homemade dataset, intercepts 500 UAV images through the simulation platform, and annotates the UAV images in YOLO format using labelImg, the corresponding 500 txt label files are generated, and the simulated UAV images and the corresponding label files are segmented into training dataset, validation dataset and test dataset according to the ratio of 8:1:1.

3.2 Experimental environment

The experiments environment parameters are set as shown in Tabel 1. On this basis, experimental verification will be carried out

Tabel 1 Experiment environment

Factor	Version
Operating system	Windows 10
GPU	NVIDIA GeForce RTX 3080
CPU	Intel(R) Core(TM) i9-13900K
Deep learning framework	Pytorch 2.2.1
Cuda	Cuda 12.1

The training image size is configured to 640×640, the training round epoch is 300, the number of specified pause training observation rounds is 200, and the training batch is 16.

3.3 Assessment of indicators

In this paper, to estimate the performance of the model, we quoted P for precision, R for recall, AP for average precision, mAP for mean Average Precision, and FPS for frames per second. Additionally, mAP50 and mAP50-95 are quoted to further evaluate the precision metrics, in which mAP50 denotes the mAP value when the IoU is 50%, and mAP50-90 denotes the mAP value when the mAP value when the IoU is between 50% and 95%. Where P, R, AP, mAP can be derived from the following equation:

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$AP = \int_0^1 P(R)dR \tag{7}$$

$$mAP = \frac{\sum_{i=0}^{n} AP(i)}{n} \tag{8}$$

where TP signifies the quantity of targets that were anticipated to be accurate. FP indicates the quantity of targets that were predicted to be correct that were not, and FN indicates the quantity of targets that were not predicted to be correct that were.

3.3 Ablation experiments and comparative analysis

In this paper, two scale models, YOLOv8s and YOLOv8l, are selected as the benchmark models, in which the depth factor of YOLOv8s is configured to 0.33, the width factor is configured to 0.5 and the maximum quantity of channels is 1024. Besides, the depth factor of YOLOv8m is configured to 1, the width factor is configured to 1, and the maximum number of channels is 512. Ablation experiments are used to validate the proposed improved offence method. For the effects of different scale models, the setup experiments are as follows:

- Experiment 1: Original YOLOv8s model.
- Experiment 2: Adding C2fM module to Experiment 1.
- Experiment 3: Add C2fM_MetaFormer module to Experiment 1
- Experiment 4: Add the C2fM_Rep module to Experiment 1.
- Experiment 5: Original YOLOv8m model.
- Experiment 6: Adding C2fM module to Experiment 5.
- Experiment 7: Add C2fM_MetaFormer module to Experiment 5.
- Experiment 8: Add the C2fM_Rep module to Experiment 5.

Under the condition of the same dataset and experimental environment, the models of 8 experiments were trained separately and their optimal models were tested on the test dataset as shown in Tabel 2.

Tabel 2 Ablation experiments

Exp	P/%	R/%	mAP50/%	mAP50-95/%	FPS
1	98.5	95.5	98.9	76.5	208.2
2	98.7	96.5	98.5	77.6	208.2
3	98.6	94.7	98.6	75.1	217.3
4	98.6	94.7	98.6	75.1	238.1
5	98.7	95.7	99.1	76.7	204.1
6	98.8	96.7	98.7	77.8	204.1
7	98.7	94.8	98.6	75.2	212.8
8	98.7	94.8	98.6	75.2	222.2

Experiments 1 to 4 used different configurations of YOLOv8s, while experiments 5 to 8 employed the YOLOv8m configuration. Both experiments were formulated to assess the impact of structural adjustments on model performance. Comparing Experiment 1 and Experiment 2, we can see that the introduction of the C2fM module leads to enhanced performance, indicating that the C2fM module can effectively optimise the detection precision and recall of the framework. Particularly, in Experiment 2, mAP50-95 is slightly improved over Experiment 1, which indicates that the model's performance in handling multi-scale object detection is enhanced. Further observations were made on Experiments 3 and 4, which introduced the C2fM_MetaFormer and C2fM_Rep modules, respectively. The results show that although these improved architectures have a slight decrease in precision and recall, the FPS is significantly improved, especially in Experiment 4, where the FPS increases to 238.1, which is an important advantage for real-time target detection systems that require high frame rate applications.

To further evaluate the effectiveness of network improvement, we compared the improved YOLOv8 algorithm with other mainstream algorithms, so this paper chooses the most commonly used YOLOv5 and the latest YOLOv9 to conduct comparative experiments, and sets up the experiments as follows:

- Experiment 9: Original YOLOv5s model.
- Experiment 10: Original YOLOv5m model.
- Experiment 11: Original YOLOv9s model.
- Experiment 12: Original YOLOv5s model.

The data obtained for the comparison experiment is shown in Tabel 3.

Tabel 3 Comparative Experiments

Exp	P/%	R/%	mAP50/%	mAP50-95/%	FPS
9	99.8	97.5	99.3	79.7	17.8
10	99.8	97.5	99.5	79.5	16.2
11	94.6	95.0	98.2	76.2	28.7
12	97.2	95.0	97.8	77.4	27.1

The YOLOv5s and YOLOv5m models show extremely high accuracy on the mAP50 and mAP50-95, with an accuracy of more than 99%, which demonstrates that the YOLOv5 series remains highly competitive in maintaining high-precision detection. However, these two models have low FPS of 17.8 and 16.2 respectively, which limits their use in application scenarios that require fast response. On the other hand, although the YOLOv9s and YOLOv9m models are slightly lower than YOLOv5 in terms of accuracy, they perform better in terms of FPS, especially for YOLOv9s, which has an FPS of 28.7. This reflects that the YOLOv9 model has been designed to optimise for speed, that makes it better suited for real-time detection tasks. Combining the data in Tabel 2and Tabel 3, we can arrival at a conclusion that the YOLOv8 model, through structural reparameterization and modular design, significantly improves the processing speed while maintaining high accuracy, especially when combined with the MetaFormer architecture, which achieves a better trade-off between the speed and accuracy of the model. This makes YOLOv8 a target detection model with greater advantages in multiple application scenarios.

4 Conclusion

In this study, we successfully designed and implemented a lightweight YOLOv8-based UAV tracking vision algorithm, called YOLO_Rep, aiming to enhance better balance between detection accuracy and processing rate in UAV target detection. By employing structural reparameterization techniques and introducing a new modular design, our algorithm significantly enhances the model's processing rate while preserving high detection accuracy, making it

particularly suitable for real-time operation in resource-constrained environments.

The main contribution of algorithmic improvement:

1.Lightweight design: We have efficiently diminished the computational burden of the model by introducing MobileNetV3 and the MetaFormer architecture of lightweight Vision Transformers (ViTs). This design allows YOLO_Rep to significantly reduce model complexity and runtime without sacrificing accuracy.

2.Multi-branch structure: By optimising the block structure in the backbone network to a multi-branch structure, our model is able to achieve parallel execution of kernel computations in the training phase, which further accelerates the training rate of the model. This structure is simplified into a more efficient convolutional operation during the inference process by the structure reparameterization technique, which greatly improves the inference speed of the network.

3.Experimental validation: We designed ablation experiments and comparative experiments that proved the model has the advantages of enhanced detection accuracy and speed. This represents a significant improvement over the original algorithm and offers greater advantages in UAV tracking and detection algorithms.

5 Acknowledgments

This work was supported by National Key R&D Program of China (Grant No. 2018AAA0103100).

6 References

[1] Alotaibi E T, Alqefari S S, Koubaa A. Lsar: Multi-uav collaboration for search and rescue missions[J]. IEEE Access, 2019, 7: 55817-55832.
[2] Zhou G, Reichle S. UAV-based multi-sensor data fusion processing[J]. International journal of image and data fusion, 2010, 1(3): 283-291.
[3] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
[4] Al-Qubaydhi N, Alenezi A, Alanazi T, et al. Detection of unauthorized unmanned aerial vehicles using YOLOv5 and transfer learning[J]. Electronics, 2022, 11(17): 2669.
[5] Liu M, Wang X, Zhou A, et al. Uav-yolo: Small object detection on unmanned aerial vehicle perspective[J]. Sensors, 2020, 20(8): 2238.
[6] Li Z, Namiki A, Suzuki S, et al. Application of low-altitude UAV remote sensing image object detection based on improved YOLOv5[J]. Applied Sciences, 2022, 12(16): 8314.
[7] Al-Qubaydhi N, Alenezi A, Alanazi T, et al. Detection of unauthorized unmanned aerial vehicles using YOLOv5 and transfer learning[J]. Electronics, 2022, 11(17): 2669.
[8] Zeng Y, Zhang T, He W, et al. Yolov7-uav: An unmanned aerial vehicle image object detection algorithm based on improved yolov7[J]. Electronics, 2023, 12(14): 3141.
[9] Zhai X, Huang Z, Li T, et al. YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection[J]. Electronics, 2023, 12(17): 3664.
[10] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arxiv preprint arxiv:1704.04861, 2017.
[11] Mehta S, Rastegari M. Separable self-attention for mobile vision transformers[J]. arxiv preprint arxiv:2206.02680, 2022.
[12] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1314- 1324.
[13] Qin D, Leichner C, Delakis M, et al. MobileNetV4-Universal Models for the Mobile Ecosystem[J]. arxiv preprint arxiv:2404.10518, 2024.
[14] Li Y, Hu J, Wen Y, et al. Rethinking vision transformers for mobilenet size and speed[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 16889-16900.
[15] Yu W, Luo M, Zhou P, et al. Metaformer is actually what you need for vision[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 10819-10829.
[16] Wang A, Chen H, Lin Z, et al. Repvit: Revisiting mobile cnn from vit perspective[J]. arxiv preprint arxiv:2307.09283, 2023.
[17] Ding X, Zhang X, Ma N, et al. Repvgg: Making vgg-style convnets great again[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13733-13742.