An Improved RT-DETR-Based Object Detection Algorithm for UAV Aerial Photography

Yingying Long
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail:longyingying@st.xatu.edu.cn

Liuhua Di
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: di.liuhua@technopro.com.cn

Yanfang Fu
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: fuyanfang@xatu.edu.cn

Shifeng Zhao
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: fuyanfang@xatu.edu.cn

Xiaojun Bai
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: baixiaojun@st.xatu.edu.cn

***Abstract*—Object detection and recognition in drone aerial images hold broad application value, but also present challenges such as large variations in object scales, difficulties in detecting small objects, and occlusions in dense scenes. To address these issues, this paper proposes an improved object detection algorithm based on RT-DETR. First, a Spatial-Channel Collaborative Attention (SCSA) module is introduced into the PResNet backbone network to enhance feature representation and improve detection accuracy. Second, the Content-Aware ReAssembly of Features (CARAFE) upsampling method is adopted in the Hybrid Encoder, which preserves more detailed information of small objects while reducing model complexity, further boosting detection performance. Finally, a modified MFRC3 module incorporating Biphasic Feature Aggregation Module (BFAM) and boundary attention mechanism is proposed to replace the original CSPRepLayer. This enhances multi-scale feature fusion and improves the retention of fine-grained and textural features.Experimental results on the VisDrone2019 datasets show that the improved algorithm achieves an mAP@0.5 of 51.1%, which is 3.1% higher than the baseline RT-DETR model.**

Keywords-Target Detection; RT-DETR; Deep Learnin g; Transformer; Attention Mechanism; Unmanned Aerial Imagery

I. INTRODUCTION

Drones, with their advantages of flexibility, low cost, and high-resolution imaging, have been widely applied in fields such as agricultural monitoring, disaster assessment, and traffic inspection. However, object detection in drone aerial imagery still faces significant challenges: First, due to high shooting altitudes, objects in images typically exhibit multi-scale characteristics, small sizes, and high density, leading to frequent false negatives and false positives in traditional detection models. Additionally, real-time detection tasks impose stringent demands on algorithmic lightweighting, resulting in poor detection capabilities of existing lightweight detection techniques for dense, small-object scenarios. These factors severely limit the practical effectiveness of drone vision systems.

DOI: 10.2478/ijanmc-2025-0036

The evolution of object detection technology traces back to the 1990s, undergoing a major transformation from traditional methods to deep learning approaches. The success of AlexNet in the 2012 ImageNet [1] competition marked the dawn of the deep learning era. In 2014, the introduction of R-CNN [2] pioneered the two-stage detector approach, generating candidate regions through selective search before employing CNNs for feature extraction and classification. The subsequent Fast R-CNN [3] further optimized this workflow. Early detection relied primarily on manually designed feature extraction methods. In recent years, lightweight models like YOLO and DETR-related variants have become prevalent for UAV aerial detection tasks. In 2016, Redmon et al. [4] introduced YOLO (You Only Look Once), pioneering a new paradigm of single-stage detection by transforming the task into a regression problem, significantly boosting detection speed. In 2017, the Transformer [5] architecture achieved breakthroughs in natural language processing and was subsequently adopted in computer vision. In 2020, Facebook's Detection Transformer (DETR) [6] successfully applied the Transformer architecture to object detection for the first time. Utilizing an encoder-decoder structure and ensemble prediction, it achieved true end-to-end detection, eliminating post-processing steps like non-maximum suppression (NMS) required by traditional methods. However, DETR suffers from slow training convergence and poor detection performance for small objects.

Subsequent research proposed various improvements to address these shortcomings. Microsoft's Deformable DETR[7] reduced computational complexity through a deformable attention mechanism and enhanced small object detection capabilities, though its computational cost remained higher than traditional CNN methods. ByteDance's DN-DETR (Denoising DETR) [8] introduced denoising training to accelerate model convergence, but it proved sensitive to noise and relied on high-quality annotated data. Liu et al.'s DAB-DETR (Dynamic Anchor Boxes DETR) [9] employs dynamic anchor boxes to improve detection accuracy, yet its memory consumption exceeds that of traditional CNN detectors, and detection performance is significantly influenced by anchor box initialization. In 2023, Baidu's RT-DETR (Real-Time DETR) [10] achieved significant inference efficiency gains through hybrid encoder design and velocity optimization, while preserving the end-to-end advantages of the DETR series. This made it the first DETR variant capable of meeting real-time detection demands. RT-DETR employs a reparameterization design and efficient feature fusion strategy, achieving an optimal balance between accuracy and speed across multiple benchmarks. However, improvements remain possible in feature extraction efficiency and small object detection.

Given RT-DETR's outstanding performance in object detection and to accommodate limited computational resources on edge devices, this paper proposes a more practical improved model based on the lightweight RT-DETR-r18 baseline. This model addresses the challenges of detecting small, multi-scale objects in complex backgrounds and dense object scenarios within drone imagery. The main contributions of this paper include:

- The Spatial-Channel Synergistic Attention (SCSA) module was introduced, replacing the original BasicBlock in PResNet with the more efficient BasicSCSABlock. By synergistically leveraging both channel and spatial semantic information, this module enhances discriminative feature representation while reducing redundant computations, significantly decreasing model parameters and improving detection accuracy.
- Introducing the CARAFE content-aware upsampling operator to replace the original upsampling method enables adaptive feature reorganization and reconstruction based on input characteristics. This effectively preserves high-frequency details, enhances small-object representation capabilities, and further improves overall model performance;
- Proposes a Multi-Field-of-View Fusion Module (MFRC3) that reconstructs the RepC3 architecture by embedding a

Bidirectional Feature Aggregation Mechanism (BFAM). This module integrates multi-scale contextual information and enhances boundary feature extraction, mitigating background interference while reducing feature loss. Consequently, it significantly improves detection robustness and accuracy in complex scenes.

II. RT-DETR Object Detection Model

RT-DETR combines the strengths of CNN and Transformer models, making it a highly practical real-time detector. The overall architecture of this algorithm is shown in Figure 1.

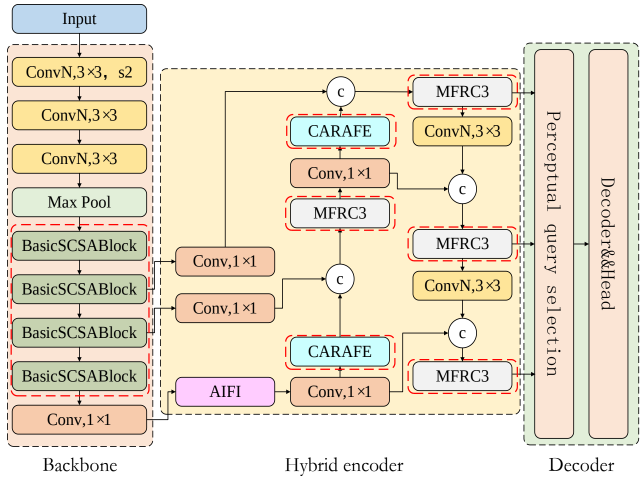


Figure 1. RT-DETR structure diagram

The core architecture of RT-DETR consists of three key modules: the backbone network, the encoder, and the decoder with query selection. The backbone network is responsible for extracting multi-scale features and outputs three feature maps for subsequent processing. The encoder adopts an innovative dual-module design, first utilizing self-attention mechanisms to focus on high-level feature interactions and avoid redundant calculations of low-level features, and finally employing a lightweight CNN to achieve cross-scale feature fusion. The decoder employs an uncertainty minimization strategy to optimize query selection, jointly evaluating classification and localization confidence to screen high-quality initial queries, thereby effectively enhancing detection performance.

III. Algorithm improvements

For the task of object detection in drone imagery, we propose an improved detection algorithm based on RT-DETR, specifically optimized for this scenario. . Specifically, we introduce the novel BasicSCSABlock module, which incorporates a Spatial-Channel Attention (SCSA) mechanism within the Basicblock module of the PResNet backbone. This enhances feature representation capabilities while accelerating inference speed. Second, the Hybrid Encoder employs the Content-Aware ReAssembly of Features (CARAFE) upsampling method to preserve more small object details, thereby improving detection accuracy for small targets. Finally, the feature fusion module within the Feature Pyramid Network is enhanced to create the MFRC3 module based on a boundary attention mechanism, optimizing the multi-scale feature fusion process.The overall framework of the improved algorithm is shown in Figure 2.

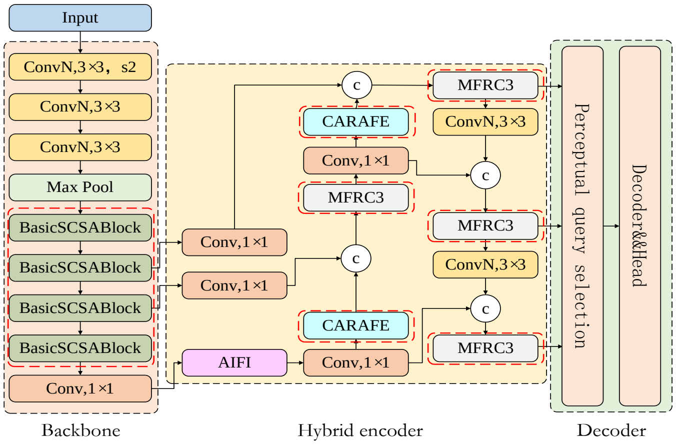


Figure 2. Improved RT-DETR algorithm

A. Improvements to the Backbone Network

The SCSA [11] attention module consists of two core components: Shared Multi-Semantic Space Attention (SMSA) and Progressive Channel Self-Attention (PCSA). The former effectively provides spatial prior information for channel attention through the integration and progressive compression of multi-semantic space information, while the latter utilizes the spatial information output by SMSA to further optimize channel features through a self-attention mechanism. Together, they form a collaborative mechanism that guides channels based on spatial information.The module structure is shown in Figure (a) below:

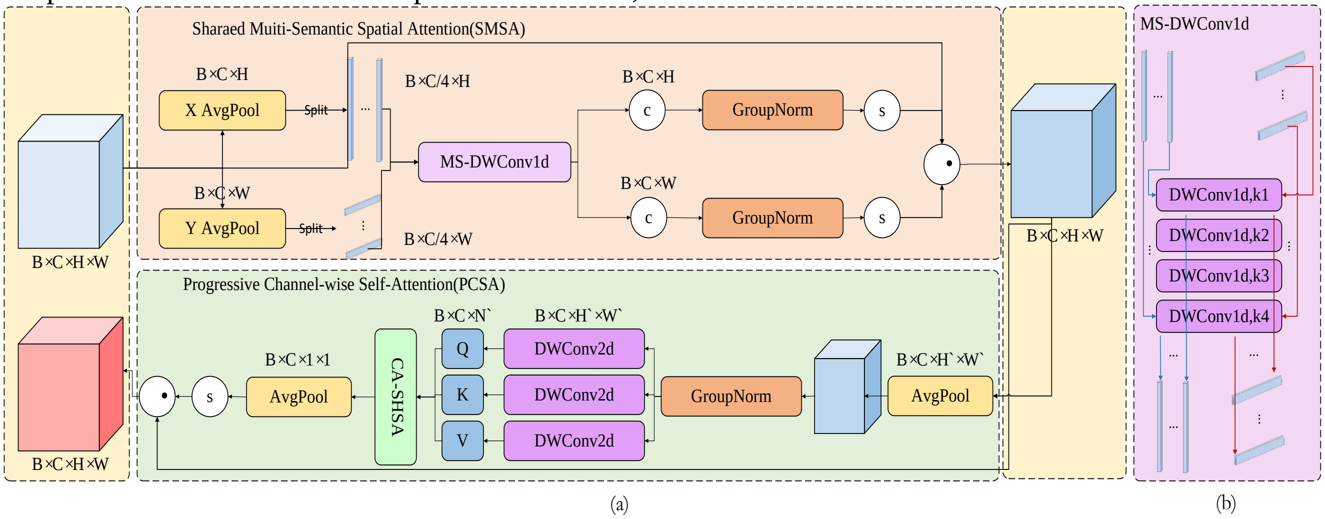


Figure 3. SCSA overall structure diagram. (a) Shows the structure diagram of SCSA, and (b) shows the structure diagram of MS-DWConv1d

First, feature map X∈R(B×C×H×W) input the SMSA module, perform global average pooling along the H and W dimensions, respectively, to obtain Xh∈R(B×C×W) and Xw∈R(B×C×H),By decoupling the dimensions of the feature map, we reduce computational complexity while retaining unidirectional spatial structure information. Subsequently, using a lightweight convolution strategy, we divide the features into four sub-features and apply deep separable 1D convolutions with kernel sizes of 3/5/7/9. This allows us to capture spatial features at different semantic levels using multi-scale convolutions while reducing the computational load on the model. The MS-DWConv1d structure diagram is shown Figure 3(b).

Considering that BN's normalized statistics come from mixed data of all sub-features within the same batch, which may cause statistical information between different semantic features to interfere with each other, SCSA uses four sets of Group Normalization to maintain semantic isolation between sub-features. GN avoids batch statistical noise and better distinguishes semantic differences between sub-features, while independent normalization prevents attention dilution.

To reduce the computational cost of subsequent self-attention, the discriminative spatial prior provided by SMSA is retained, and the feature map is compressed. Compared to a global compression strategy, a progressive compression strategy reduces information loss, so the feature map is progressively pooled and compressed, reducing its size from 7×7 to 1×1.

Subsequently, to leverage the Transformer's powerful relationship modeling capabilities, and based on experimental evidence that the single-head attention mechanism is faster than the multi-head attention mechanism combined with channel mixing design, the single-head self-attention mechanism is adopted to calculate the similarity between channels. The single-head design enhances channel interaction and mitigates the semantic differences introduced by SMSA. The

CA-SHSA module structure is shown in the figure below.

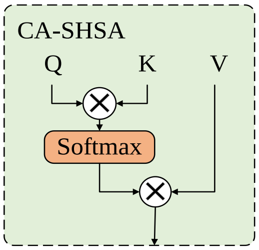


Figure 4. CA-SHSA Module Structure

Use spatial prior (SMSA) to guide channel recalibration, highlight semantic differences between sub-features, compress the feature map space, multiply it by the input, and obtain the final output.

B. Improvements to the Upsampling Module

In drone-based object detection tasks, algorithms must process complex scenes from the drone's perspective, These scene features vary in scale, are densely distributed, and have complex backgrounds. RT-DETR, as a feature upscaling operation in the Transformer-based Hybrid Encoder, has a significant impact on the final detection performance. Traditional upscaling methods (bilinear interpolation/transposed convolution) have fixed patterns, are insensitive to content, and are limited in locality, making it difficult to adapt to target features and global context. The module structure is shown in the figure below.

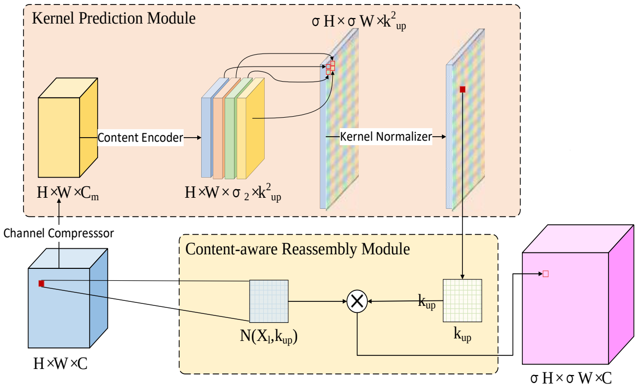


Figure 5. CARAFE Module Structure Diagram

Wang [12] et al. proposed CARAFE (Content-Aware ReAssembly of Features) as an advanced feature restructuring operator that can dynamically adjust the feature restructuring process in a content-aware manner. CARAFE consists of a kernel prediction module (Kernel Prediction Module) and a content-aware restructuring module (Content-aware Reassembly Module). The kernel prediction module dynamically generates content-aware upsampling kernels, while the content-aware restructuring module uses the predicted kernels for feature restructuring. This paper replaces the upsampling operation in the Hybrid Encoder with CARAFE. CARAFE achieves content-aware feature restructuring with a large receptive field by predicting position-specific restructuring kernels.

First, in the nuclear prediction module, to reduce the subsequent computational load, the feature map is compressed from C to Cm channels using 1×1 convolution. To predict the upsampling weights based on local content, the upsampling kernel kup is generated using kencoder×kencoder convolution, and then softmax normalization is applied to each upsampling kernel. The calculation formula is as follows:

$$W_{raw} = Conv_{k_{encoder}}\left(X_{comp}\right) \quad (1)$$

$$W_{l'}\left(n,m\right) = \frac{\exp\left(W_{raw}(n,m)\right)}{\sum_{i,j=-r}^{r}\exp\left(W_{raw}(i,j)\right)}, r = k_{up} / 2 \quad (2)$$

Among them, $W_{raw} \in R^{\sigma^2 k_{up}^2 ? H ? W}$, each position outputs a σ2k2up vector, corresponding to the σ2 target positions of the kup×kup kernel.

Next, in the content-aware reconstruction module, the predicted kernel is used to perform a weighted sum of the local regions of the input features. First, the output position l'(i',j') needs to be mapped to the center position l(i,j) of the input.

Then extract the kup×kup neighborhood centered on l from the input.

$$N\left(X_l, k_{up}\right) = \left\{X_{(i+n,j+m)} \middle| \ n,m \in [-r,r]\right\} \quad (3)$$

Weight the features within the neighborhood and sum them to obtain the module output value, as shown in the following formula:

$$X_l^{''} = \sum_n \sum_m W_l^{'}(n, m) \cdot X_{(i+n, j+m)} \qquad (4)$$

C. Improvements Based on the RepC3 Module

UAV aerial photography images typically have complex backgrounds, and in scenarios with dense targets and shooting angles, there are often many occlusion phenomena, which place higher demands on the expressive capabilities of feature extraction. In the FPN and Pan modules of the hybrid_encoder, the traditional RepC3 module enhances feature extraction capabilities by repeatedly stacking convolutional layers. However, when handling multi-scale targets and complex scenes, its receptive field and feature fusion capabilities may be insufficient.

RepC3 [13] achieves efficient gradient flow and inference acceleration through the CSP structure and RepVGGBlock. However, fixed 3×3 convolutions struggle to cover extreme scale changes, leading to small target detection failures and blurred edges of large targets; simple residual addition cannot dynamically weight cross-temporal and spatial features, resulting in insufficient target distinguishability in complex backgrounds; and static convolutions lack adaptive filtering mechanisms, making them susceptible to lighting changes and shadow interference. The module structure is shown in the figure below.

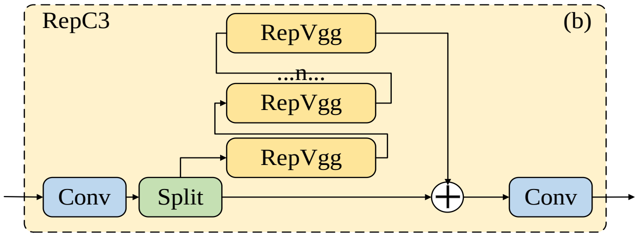


Figure 6. Structural diagram of RepC3

Compared to RepVGGBlock, which only combines 3×3 and 1×1 convolutions and may lose some details of the input features, RepConv implicitly implements identity connections through the BN layer, preserving the original input information and alleviating the vanishing gradient problem. RepConv (Re-parametrized Convolution) is a convolutional module with different structures during training and inference stages, primarily designed to enhance model efficiency. During training, multiple branches are processed in parallel, while during inference, parameter reparameterization techniques merge multiple branches into a single 3x3 convolution, maintaining the same computational effectiveness while reducing computational overhead.

The original operation of simply adding elements one by one across two feature map branches is a static, indiscriminate fusion that cannot distinguish the importance of different spatial locations or channels. When the feature distributions across the two branches differ significantly, direct addition may lead to feature cancellation or blurring. BFAM (Bitemporal Feature Aggregation Module) [14], originally used for change detection tasks in remote sensing images such as B2CNet, is an efficient feature fusion structure that combines multi-scale depth-separable convolutions and the SimAM parameter-free attention mechanism. It extracts multi-scale features by parallel use of convolution branches with different dilation rates and adaptively enhances key features using the attention mechanism. Finally, residual connections are used to fuse the input and weighted features, significantly enhancing the model's feature fusion capabilities in multi-scale object detection tasks while maintaining low computational overhead. The module structure is shown in the figure below.

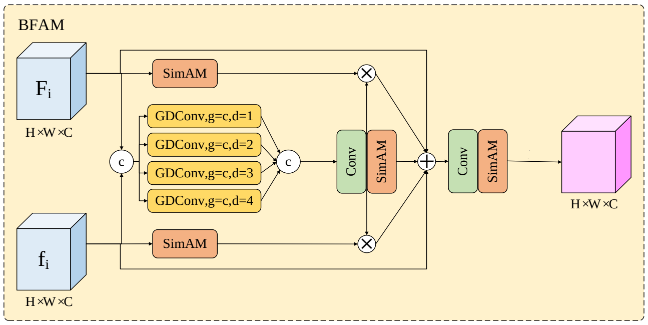


Figure 7. BFAM Structure Diagram

In summary, we introduce the BFAM and RepConv modules, rethink the RepC3 module, and form the MFRC3 (Multi-scale Feature-enhanced RepC3) module. For the RepC3 part, we retain the repeated convolution structure of RepC3, replace the RepVggBlock module with the RepConv module to extract local features, and then connect the BFAM module to the output of

RepC3 to perform multi-scale feature aggregation. The MFRC3 structure is shown in the figure below.

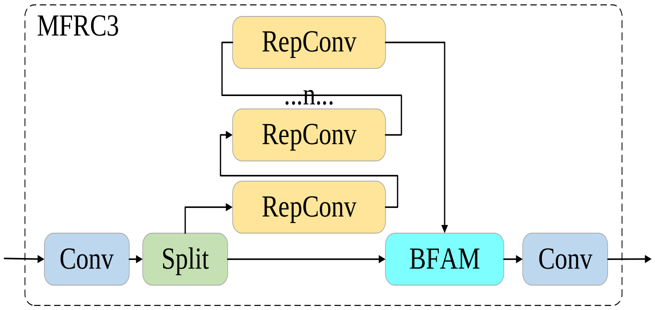


Figure 8. MFRC3 Module Structure Diagram

The specific calculation process for the MFRC3 module is as follows: First, the input feature maps are passed through two 1×1 convolutions, cv1 and cv2, to generate features from two paths. One path retains the original features, while the other is sent to the RepConv block for processing. Next, the features from both paths are input into the BFAM module, where four different dilation rates (dilation=1,2,3,4) are used in parallel to extract features via dilated convolutions. Dilated convolutions enable the capture of broader contextual information without increasing the number of parameters, thereby expanding the receptive field. Each set of convolutions uses group convolutions (group=c, where c is the number of channels) to reduce the number of parameters. To fuse multi-scale features and suppress redundant information, the multi-scale features are concatenated and then reduced in dimension via a 1×1 convolution.

Among them, d is the expansion rate, g is the number of groups, and different d values are used to capture spatial information at different scales. The SimAM attention mechanism is then used to highlight important features in the feature map. First, the energy function is calculated, and then the energy matrix composed of all energies is reweighted. The calculation formula is as follows:

$$e_t^* = \frac{4\left(\hat{\sigma}^2 + \lambda\right)}{(t-\hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda} \tag{5}$$

$$\tilde{f}_{down} = \mathrm{sigmoid}\left(\frac{1}{E}\right) \square\, f_{down} \tag{6}$$

Among these, t represents the neuron, $\hat{\mu}$ and $\hat{\sigma}^2$ denote the mean and variance of all neurons except t, respectively. λ is a tuning parameter (default 1e-4). E performs feature reweighting on the energy matrix composed of all energies.

The reduced-dimension features are multiplied by the original features processed through the SimAM module, and residual connections are used to retain detailed information. After convolution and SimAM processing, the final output is obtained. The multiplication operation emphasizes the similarity between features, while residual connections prevent information loss in deep networks. This enhances the texture details and spatial relationships in the target region.

IV. Experiments And Analysis Of Results

A. Datesets

VisDrone [15] is a large-scale object detection benchmark datesets specifically designed for drone aerial photography scenarios, making it particularly suitable for evaluating the performance of drone object detection algorithms in research. The datesets was collected by the MLDM Lab at Tianjin University and contains over 10,000 high-resolution images, annotated with 2.6 million instances, covering 10 typical drone-view targets such as pedestrians, vehicles, and bicycles. Its notable features include a large number of small targets, dense occlusions, and complex backgrounds, which closely align with the challenges faced by real-time detection algorithms like RT-DETR. The data was collected at various altitudes, lighting conditions, and scenes, providing an ideal testing platform for improving algorithm performance in terms of scale adaptability, small object detection, and occlusion resistance.

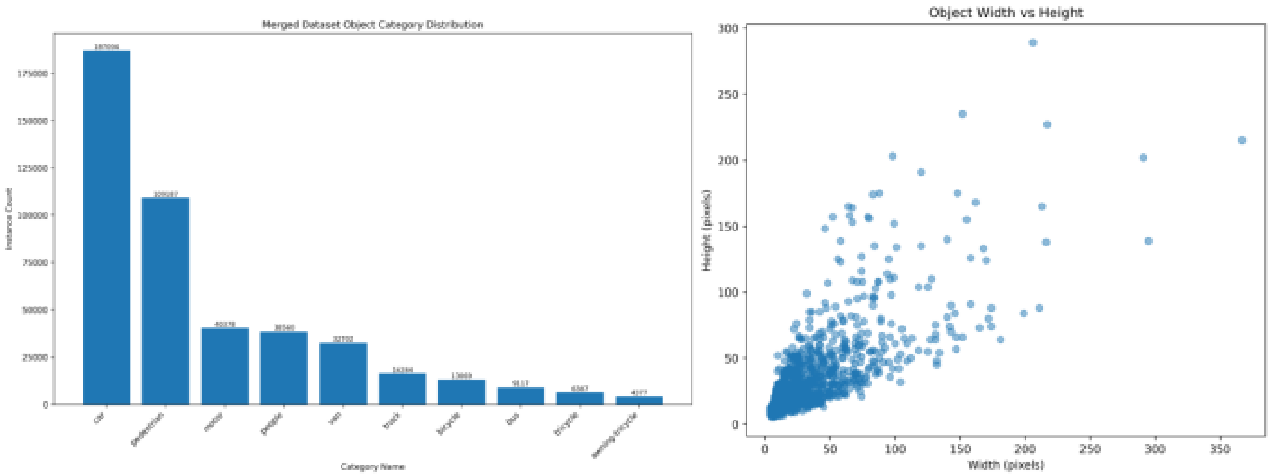


Figure 9. Target category and size distribution chart

B. Experimental setup

The experiment was evaluated on the VisDrone2019 and a datesets, as shown in Table 1. The experimental environment was Ubuntu 20.04, using NIVDIA Corporation GP102 [TITAN X], CPU using E4-2609 v4@1.70GHz×16, PyTorch version 2.0.1, CUDA version 12.1, and Python version 3.8.10.

As shown in Table 2, the table displays the hyperparameter information used in the experiment. To process high-resolution input images with limited GPU memory resources, batch_size is set to 4. The model training requires sufficient training epochs to learn complex feature representations, especially for small objects and occlusion scenarios, so epoch is set to 200. The optimizer selected is AdamW, whose adaptive learning rate characteristics and decoupled weight decay are more suitable for handling transformer architectures and class imbalance issues. The setting of num_workers=4 is to match the typical 4-core CPU configuration, ensuring that data preprocessing and loading do not become training bottlenecks; while the selection of a 640×640 input size is an experimentally validated balance point that retains sufficient small object information while maintaining relatively fast inference speeds.

TABLE I. EXPERIMENTAL HYPERPARAMETERS

Hyperparameters	Configurations
batch_size	4
epoch	200
optimizer	AdamW
num_workers	4
image_size	640×640

C. Ablation experiment

To analyze the effectiveness of each improvement strategy in enhancing model performance, we systematically conducted ablation experiments on the Visdrone datasets using RT-DETR as the baseline model. The experiments sequentially introduced the BasicSCSABlock module, CARAFE module, MFRC3 module, and their various combinations—including BasicSCSABlock with CARAFE, CARAFE with MFRC3, BasicSCSABlock with MFRC3, and all three modules together. Performance changes were compared in detail across seven evaluation metrics, with corresponding validation set results shown in Table 2.2. The results demonstrate that each module effectively enhances model performance: BasicSCSABlock increases mAP@0.5:0.95 by 0.1% while reducing parameter count; CARAFE's upsampling operation boosts APl from 54.8% to 57.1%, significantly strengthening large-scale feature reconstruction capabilities; The MFRC3 module elevated mAP@0.5:0.95 from 28.5% to

30.4% while also improving APs and APm, demonstrating the effectiveness of its multi-scale feature fusion. Ultimately, the model integrating all modules achieved an overall performance improvement of 3.1% over the baseline, with mAP@0.5:0.95 reaching 53.7%, fully validating the effectiveness and synergistic nature of the proposed enhancement modules.

TABLE II. THE EFFECTS OF EACH IMPROVEMENT MODULE

BasicSCSABlock	CARAFE	MFRC3	Param/MB	APs/%	APm/%	APl/%	mAP@0.5/%	mAP@0.5:0.95/%	FPS
-	-	-	19.2	19.2	39.3	54.8	48.0	28.5	81.4
√	-	-	18.1	19.3	39.4	54.8	48.1	28.6	80.5
-	√	-	18.2	19.3	39.5	57.1	48.5	28.9	70.8
-	-	√	34.2	20.3	42.0	56.6	50.5	30.4	49.5
√	√	-	18.2	18.9	38.9	53.2	47.6	28.3	74.7
-	√	√	34.3	20.5	41.7	53.3	51.0	30.1	45.6
√	-	√	27.9	21.0	42.3	52.8	50.0	29.5	63.2
√	√	√	28.0	21.0	42.4	58.3	51.1	31.0	53.7

D. Visualization of experiments and analysis of results

The figure below shows a comparison of the improved model. The mAP curve clearly, demonstrates whether the performance of the model before and after improvement has been enhanced in the detection task. As can be observed from the figure below, the improved model in this paper has achieved significant results in mAP@0.5:0.95, with a 3.1% improvement compared to the model before improvement, and superior performance at various thresholds in the object detection task.

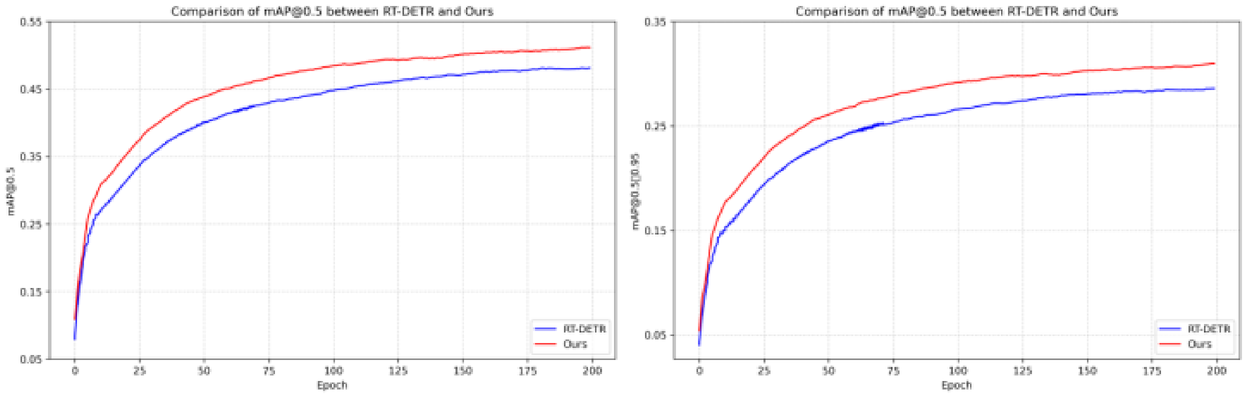
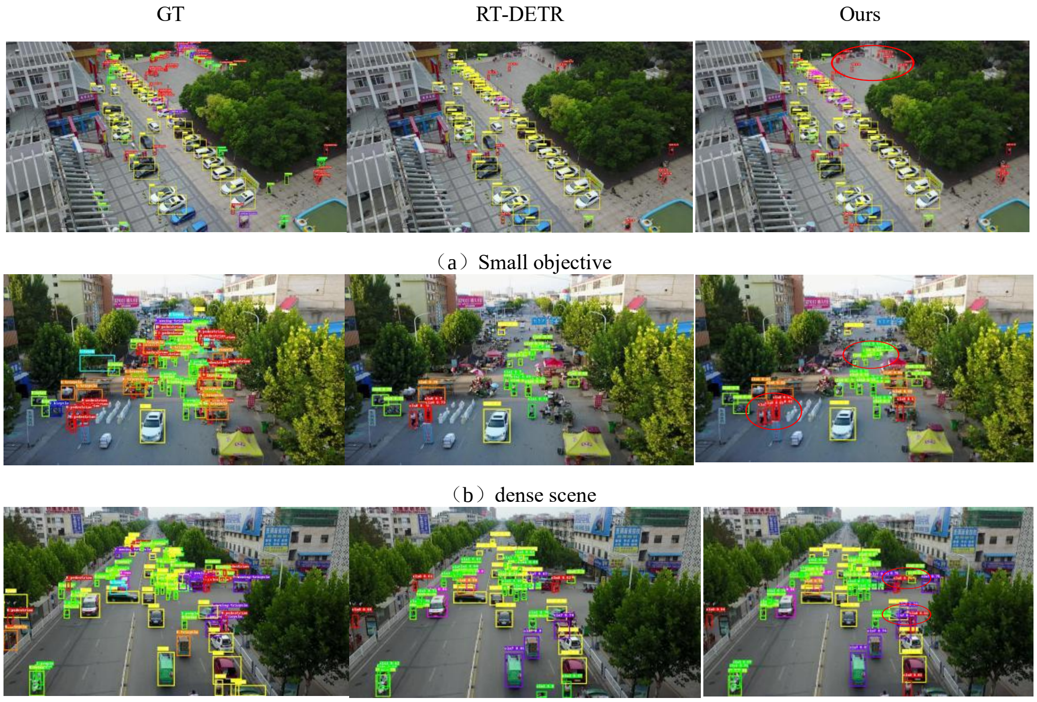


Figure 10. mAP curve comparison chart

Through visual operations, the detection performance of the model can be intuitively observed. Three representative challenging scenarios from the Visdrone2019 datesets were selected for demonstration.

As shown in the figure below, part (a) represents the detection scenario for small targets. The improved model in this paper significantly outperforms the baseline model, particularly in the detection of small, distant pedestrians. The comparison results in Figure (b) demonstrate that the improved detection model performs better in dense target detection tasks, effectively handling the detection of dense targets. Figure (c) shows the performance of the two models in multi-scale detection scenarios, with the improved target detection model maintaining stable performance.



（a）Small objective

（b）dense scene

（c）multiscale

Figure 11. Visualization of experimental results comparison

E. Comparative experiment

The table below presents the comparison experiments between the improved RT-DETR algorithm and other classical algorithms on the VisDrone datesets.

The selected classical object detection models primarily include, YOLOv5x, YOLOv8n, YOLOv8s, YOLOv10n [16], YOLOv10s, Deformable DETR [7], DINO [17], and RT-DETR [10]. From the experimental results, it can be seen that the two-stage detector Faster R-CNN achieves high accuracy through region proposals, but has high computational costs; the single-stage YOLO series excels in lightweight and real-time performance, but lightweight versions such as YOLOv10n see a significant drop in accuracy; the DETR series, including Deformable DETR and DINO, are based on Transformer architecture and improve accuracy through global modeling, but have high computational complexity. Among these, RT-DETR optimizes real-time performance through a hybrid encoder and query denoising, while this work further improves upon it, achieving higher detection accuracy with only 60% of the parameters of DINO. Compared to the YOLO series, its advantages are evident in its adaptability to small targets and dense scenes in the typical challenges of the Visdrone datesets. This model achieves state-of-the-art (SOTA) accuracy of 51.1% on the VisDrone datesets (mAP@0.5) at the cost of moderately increasing the number of parameters and computational complexity, while maintaining a practical speed of 53.7 FPS.

TABLE III. COMPARISON WITH MAINSTREAM ALGORITHMS

model	Param/MB	GFLOPs	FPS	mAP@0.5%	mAP@0.5：0.95%
YOLO v5x	86.2	203.8	34	42.5	25.2
YOLO v8n	3.0	**8.1**	119	33.5	17.8
YOLO v8s	11.2	28.7	85	37.3	21.6
YOLO v10n	**2.69**	8.2	111	29.9	17.1
YOLO v10s	8.04	24.5	**143**	36.4	21.4
Deformable DETR	40	196	29	42.2	27.1
DINO	47	279	24	46.2	29.4
RT-DETR	19.2	30.2	81.4	48.0	28.5
Ours	28.0	56.0	**53.7**	**51.1**	**31.0**

V. CONCLUSIONS

This paper addresses challenges in drone aerial imagery, such as variable target scales and complex backgrounds, by proposing an improved object detection algorithm based on RT-DETR. By incorporating a Spatial-Channel Collaborative Attention (SCSA) module, a Content-Aware ReAssembly of Features (CARAFE) upsampling method, and an enhanced MFRC3 module, the model achieves significantly improved detection performance in complex scenes. The SCSA module effectively enhances feature expression efficiency through its spatial-channel collaborative attention mechanism, accelerating inference while reducing redundant computations. CARAFE upscaling preserves more small-object detail information via content-aware feature reorganization, substantially improving detection accuracy for small targets. The MFRC3 module combines dual-phase feature aggregation with boundary attention mechanisms to optimize multi-scale feature fusion, enhancing detection capabilities for complex backgrounds and occluded objects. Experiments on the VisDrone2019 datasets demonstrate that the improved algorithm achieves 51.1% mAP@0.5, surpassing the baseline RT-DETR by 3.1 percentage points while reducing model parameters. Cross-domain validation results further confirm the algorithm's robust performance. Future work will focus on further optimizing the model architecture to enhance inference speed and exploring additional lightweight designs to adapt to practical drone applications constrained by hardware resources.

REFERENCES

[1] Dalal N, Triggs B .Histograms of Oriented Gradients for Human Detection[C]//IEEE Computer Society Conference on Computer Vision & Pattern Recognition.IEEE, 2005.DOI:10.1109/CVPR.2005.177.

[2] Girshick R , Donahue J , Darrell T ,et al.Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[J].IEEE Computer Society, 2014.DOI:10.1109/CVPR.2014.81.

[3] Girshick R .Fast R-CNN [J].Computer Science, 2015.DOI:10.1109/ICCV.2015.169.

[4] Redmon J , Divvala S , Girshick R ,et al.You Only Look Once: Unified, Real-Time Object Detection[C]//Computer Vision & Pattern Recognition.IEEE, 2016.DOI:10.1109/CVPR.2016.91.

[5] Vaswani A , Shazeer N , Parmar N ,et al.Attention Is All You Need[J].arXiv, 2017.DOI:10.48550/arXiv.1706.03762.

[6] Carion N, Massa F, Synnaeve G, et al.End-to-End Object Detection with Transformers [M]. 2020.

[7] Zhu X, Su W, Lu L, et al.Deformable DETR: Deformable Transformers for End-to-End Object Detection [J]. 2020. DOI: 10.48550/arXiv.2010.04159.

[8] Li F, Zhang H, Zhang N L .DN-DETR: Accelerate DETR Training by Introducing Query DeNoising [J].IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(4):2239-2251.

[9] Liu S, Li F, Zhang H, et al.DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR [J]. 2022. DOI:10.48550/arXiv.2201.12329.

[10] Liu S, et al. RT-DETR: Real-Time Detection Transformer. NeurIPS 2023.

[11] Wang J, et al. SCSA: Exploring Spatial-Channel Synergy. CVPR 2022.

[12] Zhang Y, et al. CARAFE: Content-Aware Feature Reassembly. ICCV 2019.

[13] Ding X, Zhang X. Ma N, et al. RepVGG: Making VGG-style ConvNets Great Again [J]. 2021. DOI:10.1109/CVPR46437.2021.01352.

[14] Li X, et al. B2CNet: Boundary-to-Center Refinement. TGRS 2023.

[15] Zhu P, Wen L, Bian X, et al.Vision Meets Drones: A Challenge [J].Springer, Cham, 2018.DOI:10.1007/978-3-030-11021-5_27.

[16] Wang A, Chen H, Liu L, et al.YOLOv10: Real-Time End-to-End Object Detection [J]. 2024.

[17] Zhang H , Li F , Liu S ,et al.DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection [J].arXiv e-prints, 2022.DOI:10.48550/arXiv.2203.03605.