

AeroRTD: boosting UAV-based airport target recognition with a lightweight enhanced RT-DETR framework

Xiaolei Qu¹, Jiacheng Zhang², Xichao Wang^{2*}, Shengdian Liu², Xia Wang¹ and Yating Jiang³

¹ School of Automation, Northwestern Polytechnical University, Xi'an 710129, China

² College of Electrical and Energy Engineering, Shanghai Dianji University, Shanghai 201306, China

³ School of Aeronautic Science and Engineering, Beihang University, Beijing 102206, China

* Correspondence: wangxc@sdju.edu.cn (Wang X)

Abstract

When micro unmanned aerial vehicles (UAVs) are deployed for airport target detection, they face significant challenges such as complex backgrounds, frequent occlusion, and large-scale variations, which degrade detection accuracy and real-time performance. To address these issues, this paper proposes AeroRTD, an enhanced detection algorithm based on an improved RT-DETR framework. Specifically, a Parallel Convolution Enhancement (PCE) module is introduced to improve feature representation, while a Multi-Scale Contextual Self-Attention (MSCSA) module and a Channel-Spatial Dual Attention (CSDA) module are designed to enhance multi-scale contextual modeling and robustness to scale variations. In addition, an Adaptive Grid Upsampling (AGU) module is proposed to improve feature fusion efficiency. Experimental results on the AFTD dataset demonstrate that AeroRTD achieves superior performance compared with both the baseline RT-DETR and several state-of-the-art object detection methods. The proposed method reaches an mAP_{0.5} of 0.681 with an inference speed of 23.89 FPS, achieving improved accuracy while maintaining a lightweight model structure (15.76 M parameters) and moderate computational cost (50.9 GFLOPs). These results indicate that AeroRTD provides an effective and efficient solution for UAV-based airport target detection in complex environments.

Keywords: Unmanned aerial vehicle, Target detection, Real-time detection transformer, Parallel convolution enhancement module, Multi-scale contextual self-attention

Citation: Qu X, Zhang J, Wang X, Liu S, Wang X, et al. 2026. AeroRTD: boosting UAV-based airport target recognition with a lightweight enhanced RT-DETR framework. *International Journal of Micro Air Vehicles* 18: e015 <https://doi.org/10.48130/mav-0026-0007>

Introduction

The airport apron—including runways, taxiways, and parking stands—serves as a critical hub within the air transportation system. Its safe and efficient operation has a direct impact on flight punctuality and passenger safety. According to statistics from the International Civil Aviation Organization (ICAO)^[1], approximately 30% of global aviation accidents occur in the apron area each year, with delays in target detection and conflict warning identified as major contributing factors. With the ongoing development of smart airports, automated target detection technologies based on computer vision have emerged as essential tools for enhancing apron surveillance. These technologies enable real-time identification of aircraft, ground vehicles, and personnel, thereby providing fundamental data support for systems such as the Surface Movement Guidance and Control System (SMGCS) and the Automatic Conflict Avoidance System (ACAS). Among them, pure computer vision (CV) methods based on visible light imaging have become core technologies for smart airport construction due to their advantages in non-contact operation, high resolution, and the ability to monitor multiple targets simultaneously^[2]. Nevertheless, the unique characteristics of airport environments—such as wide variations in target scale, frequent dynamic occlusions, and complex lighting conditions—impose stringent requirements on the accuracy and real-time performance of detection algorithms^[3]. Early research in target detection predominantly relied on traditional image processing approaches, including manually designed feature-based recognition techniques^[4,5]. These methods typically employed heuristic rules (e.g., edge detection, texture analysis) for target representation. However, their inherent limitations have become

increasingly evident. First, their feature representation capabilities are confined to specific target categories, resulting in poor generalization across different scenes and low robustness to environmental noise^[6]. Second, the overall pipeline—from feature extraction to classifier design—requires extensive manual parameter tuning. As a result, algorithmic complexity increases significantly with target diversity, making it difficult to achieve the millisecond-level latency required for real-time performance in industrial applications^[7].

In UAV-based airport monitoring, the detection model must simultaneously handle small targets, visually similar ground objects, wide-area backgrounds, and strict real-time requirements. Although CNN-based detectors and transformer-based detectors such as DETR^[8] have achieved strong performance in general object detection tasks, their direct application to airport surface scenarios remains challenging. YOLO-based methods^[9–12] are efficient but may suffer from inaccurate localization when targets are small, densely distributed, or partially occluded. Two-stage detectors such as Faster R-CNN^[13,14] can provide accurate region proposals, but their computational cost is less suitable for real-time UAV deployment. RT-DETR^[15] provides an attractive end-to-end detection framework by combining transformer modeling with real-time inference. However, in airport scenes, its feature representation and cross-scale fusion capability still need to be strengthened to better distinguish aircraft, vehicles, and personnel under complex backgrounds. To address these challenges, this paper proposes AeroRTD, a real-time detection transformer framework tailored for airport scenarios. The main contributions of this work are summarized as follows:

(1) We propose a lightweight Parallel Convolution Enhancement (PCE) module to improve fine-grained feature representation. Compared with conventional convolutional designs, PCE enhances local detail modeling and channel interaction while maintaining computational efficiency, leading to improved detection of small and visually similar targets.

(2) We introduce a Multi-Scale Context Self-Attention (MSCSA) module to handle extreme scale variations. Unlike standard attention mechanisms, MSCSA integrates multi-scale convolution with self-attention to capture both local structures and global dependencies, significantly improving robustness to large-scale disparities.

(3) We design an Adaptive Grid Upsampling (AGU) module and a Channel-Spatial Dual Attention (CSDA) module to enhance cross-scale feature fusion. Compared with traditional upsampling and fusion strategies, the proposed modules achieve more accurate feature alignment and more effective suppression of redundant information, resulting in improved detection performance under complex and dense scenarios.

Related works

Object detection, as one of the core tasks in the field of computer vision, aims to identify the locations (bounding boxes) and categories of objects within images or videos. In recent years, with the rapid advancement of deep learning technologies, object detection algorithms have achieved remarkable breakthroughs, providing robust technical support for a wide range of practical applications. In the context of airport surveillance, multi-object detection technology plays a crucial role by enabling real-time detection and tracking of various targets such as aircraft, ground vehicles, and pedestrians. This capability not only enhances the safety of surface operations by preventing collisions and abnormal behaviors but also optimizes resource scheduling and reduces flight delays. Furthermore, object detection significantly improves airport security by facilitating the rapid identification of suspicious objects. Simultaneously, it provides critical support for intelligent airport management and data-driven decision-making, thereby promoting the digital transformation and efficient operation of modern airports.

Two-stage object detection

Two-stage object detection frameworks, such as Faster R-CNN^[16], have long been acknowledged for their high detection accuracy, particularly in scenarios characterized by complex backgrounds and dense object distributions. By decoupling the detection process into region proposal generation and object classification, these methods enable precise localization and demonstrate robust performance across varying object scales. The introduction of the Region Proposal Network (RPN) has further enhanced the ability of two-stage frameworks to generate high-quality candidate regions, thereby facilitating fine-grained object classification and bounding box refinement. As a result, two-stage detectors have become the benchmark in applications that demand extremely high precision, such as medical imaging and autonomous driving.

However, in scenarios with highly diverse and densely overlapping objects, the reliance on the region proposal mechanism introduces inherent limitations. Two-stage detectors often struggle to capture the fine-grained contextual information necessary for accurate predictions, particularly for small or partially occluded objects. While the hierarchical structure of these models aids in multi-scale feature learning, it may lead to information loss in deeper layers,

diminishing the model's ability to differentiate between visually similar or contextually complex objects. These challenges are particularly pronounced in environments like airport aprons, where objects such as ground service vehicles, baggage carts, and personnel exhibit significant diversity in appearance and are frequently subject to occlusion.

Single-stage target detection

Single-stage object detectors have experienced significant advancements since their inception, primarily driven by the need to simplify the detection process and enhance computational efficiency. The introduction of SSD (Single Shot MultiBox Detector)^[17] marked a breakthrough in the region proposal mechanism of two-stage detectors. SSD uses a single convolutional network to directly predict object categories and bounding boxes, achieving object detection across different scales through multi-scale feature maps. However, its reliance on fixed anchor boxes and limited contextual information restricts its performance, particularly in detecting small objects.

The YOLO (You Only Look Once) series further revolutionized the field of single-stage detection by reinterpreting object detection as a regression problem. YOLOv1^[18] introduced a grid-based prediction mechanism, offering impressive inference speed but at the cost of some accuracy, particularly in detecting dense and small objects. Subsequent versions, such as YOLOv2^[19] and YOLOv3^[20], incorporated key improvements, including anchor boxes, multi-scale detection, and optimizations to the Darknet backbone, which significantly boosted detection accuracy while maintaining real-time performance.

Recent developments have primarily focused on addressing the intrinsic limitations of single-stage methods. YOLOv4^[21] and YOLOv5 introduced advanced features such as the Cross Stage Partial Network (CSPNet), Spatial Pyramid Pooling (SPP), and improved activation functions, enhancing feature extraction capabilities and robustness in complex environments. Additionally, lightweight versions designed for embedded systems, such as YOLOv5-Nano, have balanced speed and accuracy. YOLOX^[22] further improved detection accuracy and speed by introducing innovative features such as an anchor-free structure, multiple positive samples, decoupled heads, advanced label assignment, and robust data augmentation. However, its strong augmentation strategy may lead to the model's reliance on specific data augmentations, and further optimization is required for better robustness in highly complex scenarios. YOLOv8^[23] has made notable improvements in both detection accuracy and speed by introducing an anchor-free detection strategy, decoupled head design, and optimized network structures (e.g., the C2f module and PA-FPN). It also supports multi-task capabilities, including object detection, instance segmentation, and pose estimation. Its advantages lie in its efficiency, flexibility, and multi-task support, making it easy to deploy. However, there remains room for improvement in detecting small objects.

Overall, single-stage detectors provide a practical balance between detection speed and deployment efficiency, which makes them suitable for real-time visual perception tasks. The recent YOLO-family detectors^[23–25] and related efficient detection frameworks^[26] further improve this balance through anchor-free prediction, optimized training strategies, and streamlined network structures. However, UAV-based airport scenes introduce additional challenges that are not fully addressed by general-purpose single-stage designs, including large viewpoint changes, strong background interference, small service vehicles, and dense object distribution around aircraft.

Therefore, simply adopting existing YOLO-family detectors may not fully satisfy the accuracy and robustness requirements of airport target detection, motivating the design of a more task-oriented lightweight detection framework.

Object detection based on transformer architecture

DETR was initially proposed by Carion et al.^[8], introducing a novel paradigm to the field of object detection by discarding components traditionally relied upon in conventional methods, such as manually designed anchor boxes and Non-Maximum Suppression (NMS). DETR employs a transformer-based encoder-decoder architecture, enabling end-to-end prediction from the input image to the object bounding boxes and class labels. The model utilizes a set-based loss function, implemented via the Hungarian algorithm, to match the predicted boxes with the ground truth, ensuring a one-to-one correspondence. Despite its simplicity in structure and impressive performance, DETR faces two major limitations: (1) its training convergence is slow, typically requiring up to 500 epochs to achieve optimal performance; and (2) the model underperforms in small object detection tasks due to insufficient utilization of multi-scale features. These challenges primarily arise from the high computational complexity of the transformer attention mechanism, where the computational cost increases quadratically with the input resolution.

To address these challenges, researchers have proposed various improved models. Among them, Deformable DETR^[27] introduced a deformable attention mechanism with sparse sampling, which focuses solely on key sampling points around reference points. This approach significantly reduces computational complexity and accelerates the convergence process. Additionally, the model's performance in detecting small objects is enhanced through a multi-scale feature fusion mechanism. Building on this, SMCA-DETR^[28] further incorporates a spatial modulation co-attention mechanism in the decoder. By predicting the center and scale of the object to generate a Gaussian-shaped weight map, it improves the model's spatial selective attention, thereby enhancing training efficiency. Conditional DETR^[29] proposed a conditional cross-attention mechanism. By separating content and location queries, the decoder can more effectively focus on the object region, further mitigating the training convergence issue.

In addition, DN-DETR^[30] introduced adversarial noisy object queries, simulating the uncertainty of object presence during the training process. This approach enhances the model's robustness and accelerates convergence. Anchor DETR^[31] explicitly modeled object queries as anchor points, incorporating more physically meaningful position priors. This improvement enables the model to focus on more relevant regions. Building upon the aforementioned methods, DINO^[32] integrated contrastive learning with a multi-positive and negative noise mechanism, achieving state-of-the-art (SOTA) performance with a significantly reduced number of training epochs, particularly excelling in small object detection tasks.

Building upon these variants, RT-DETR (Real-Time DETR) was proposed to address the computational efficiency challenges of transformer models in real-time applications. The core innovation of RT-DETR lies in its hybrid encoder design, which integrates intra-scale feature interaction with cross-scale feature fusion, effectively reducing the computational burden of the transformer encoder. Additionally, RT-DETR introduces an IoU-aware query selection mechanism, which optimizes the quality of the initial object queries by leveraging Intersection-over-Union (IoU) scores during the

training process. This mechanism not only enhances detection accuracy but also ensures the model's stability during inference, eliminating the need for traditional Non-Maximum Suppression (NMS) post-processing. While maintaining high accuracy, RT-DETR significantly improves inference speed, making it a strong candidate for real-time object detection tasks.

For airport target detection, the advantage of RT-DETR^[15] lies in its end-to-end inference and favorable speed-accuracy trade-off. Compared with earlier transformer-based detectors such as DETR^[8] and its improved variants^[27–32], RT-DETR is more suitable for real-time deployment. In addition, recent fine-grained detection studies, such as PCLDet^[33], further indicate that scale-aware feature modeling and discriminative feature representation are important for detecting small and visually similar objects in aerial scenes. Nevertheless, airport apron images usually contain small targets, scale-varying aircraft, cluttered ground equipment, and partially occluded objects, which place higher demands on local detail extraction and cross-scale feature interaction. This motivates the improvement of RT-DETR from the perspectives of fine-grained feature enhancement, multi-scale contextual modeling, and adaptive feature fusion.

Methods Overview

The overall architecture of the improved RT-DETR model is designed to enhance both the accuracy and efficiency of airport surface object detection, optimizing the feature processing pipeline through innovative module design. Upon inputting the image, it first passes through the backbone network. The PCE Module (Parallel Convolution Enhancement Module) enhances the ability to extract both local and global features through parallel PConv 3×3 convolutions, linear layers, and activation functions, generating multi-level features (P_1 – P_5). In the encoder, the MSCSA (Multi-Scale Context Self-Attention Module) extracts multi-scale features from the P_5 feature map, integrates spatial and channel attention mechanisms, calibrates feature weights, and emphasizes key information. It outputs K (keys) and V (values) of size $C \times P \times P$, which are used in subsequent operations, such as multi-head self-attention, to capture associations between features at different positions.

Subsequently, the traditional upsampling operation in the CCFF Module (Feature Fusion Module) of the original model is replaced with AGU (Adaptive Grid Upsampling). This adjustment optimizes the feature fusion process by enhancing the information processing mechanism. During this optimization, the CSDA (Channel-Spatial Dual Attention Module) is introduced to dynamically capture associations between features across scales, enhance the representation of key features, and further refine the feature fusion process. The final enhanced model, referred to as AeroRTD, is illustrated in Fig. 1.

From a unified perspective, the proposed AeroRTD framework can be interpreted as an effective solution for enhancing multi-scale feature representation and cross-scale feature interaction. Specifically, the PCE module improves fine-grained feature extraction by strengthening local feature representation, while the MSCSA module enhances global contextual modeling through multi-scale attention mechanisms. Furthermore, the AGU and CSDA modules focus on optimizing cross-scale feature alignment and suppressing redundant information during feature fusion.

Overall, these modules work in a complementary manner to address key challenges in object detection, including scale variation, occlusion, and dense object distribution. This unified design perspective highlights that AeroRTD is not merely a combination of independent modules, but a coherent framework for improving

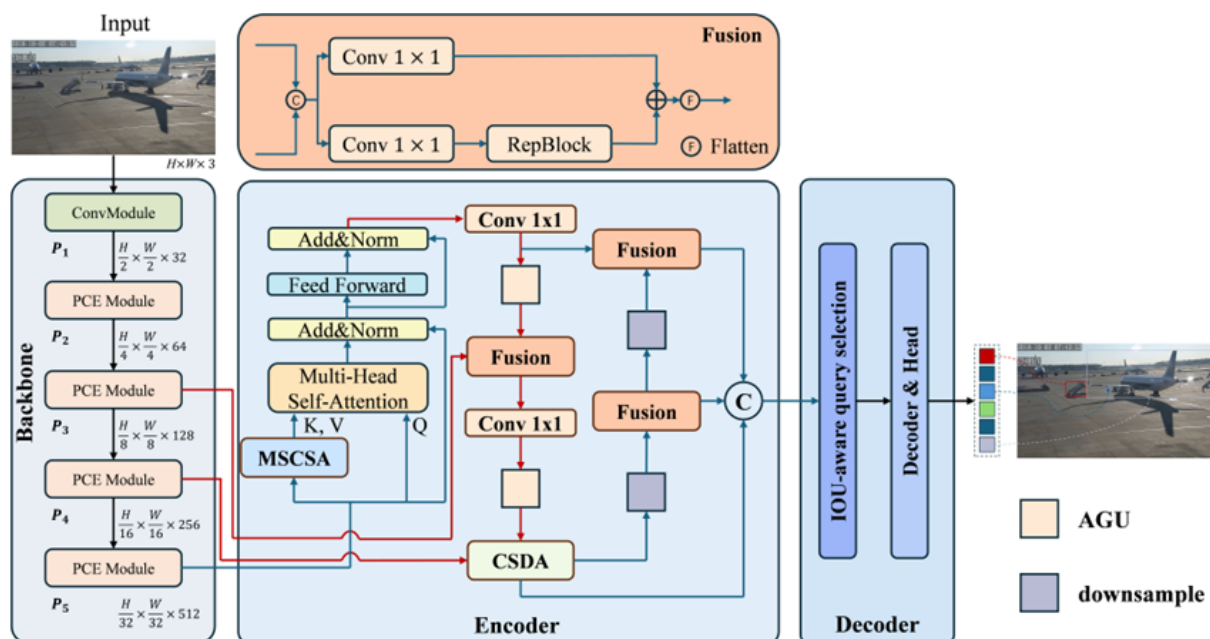


Fig. 1 Overall flow of the proposed algorithm.

feature representation and interaction in complex detection scenarios.

To further clarify the functional roles of each proposed module and avoid potential ambiguity, a structured comparison is provided in Table 1, summarizing the design motivation and unique contribution of each component in the AeroRTD framework.

Parallel Convolution Enhancement module

To improve the feature expression capability of the model for object detection in complex airport environments, this paper introduces the PCE module (Parallel Convolution Enhancement module) within the backbone network, inspired by recent lightweight convolution and feature enhancement strategies^[34]. The core concept behind this module is parallel feature extraction combined with multi-level information fusion to strengthen the model's feature extraction capacity. Specifically, it is designed to enhance the modeling of local features while maintaining computational efficiency and a lightweight structure. This approach aims to improve the model's recognition accuracy, particularly for fine-grained objects.

As illustrated in Fig. 2, the PCE module (Parallel Convolution Enhancement module) consists of two 3×3 partial convolutions (PConv) and several linear transformations, enabling stable training and feature fusion within the deep network via residual connections and DropPath technology. Initially, the input features undergo PConv 3×3 , which performs convolution operations on a subset of channels, thus significantly reducing the computational load while preserving local spatial information. The output is then fused with another branch containing linear operations before passing through

a second PConv 3×3 and activation stages to further refine the fine-grained features. During this process, the activation weight of each channel is adaptively adjusted based on the local convolution features, enabling more targeted channel attention. The fused result is projected into the feature space with the same dimensionality as the input via a linear projection, with some connections randomly discarded using DropPath to mitigate overfitting. Finally, the enhanced output features are obtained by adding the result to the original input through a residual connection. The PCE module not only captures local details and spatial context but also improves recognition performance for small objects and complex backgrounds, all while maintaining a high inference speed. It is particularly well-suited for real-time detection tasks, such as identifying aircraft and ground service vehicles in airport scenarios.

To effectively fuse features and introduce a dynamic mechanism, the PCE module is designed, and its calculation process can be described in detail through the following formulas. Let the input feature be X_{in} . The PCE module first introduces a dynamic mask M through partial convolution to screen the effective regions, and its operation is defined as:

$$X^{(1)} = PConv(X^{(0)}, M) = \frac{\text{Conv}(X^{(0)} \odot M)}{\sum_{i,j} M_{i,j} + \varepsilon} \sum_{i,j} 1 + b_1 \quad (1)$$

where M is the dynamic mask, $\odot$ represents the element-wise multiplication, and b_1 is the bias term. This operation screens the effective features by adjusting the weight distribution.

Subsequently, the feature $X^{(1)}$ goes through two independent linear transformations to achieve the interaction and adjustment of the feature dimensions. The specific calculation is as follows:

Table 1. Functional comparison of proposed modules.

Module	Targeted problem	Key idea	Position	Unique role
PCE	Small objects/fine-grained features	Parallel partial conv + linear fusion	Backbone	Enhance local details
MSCSA	Scale variation	Multi-scale conv + self-attention	Encoder	Model cross-scale context
AGU	Misaligned upsampling	Dynamic sampling grid	Fusion stage	Achieve precise alignment
CSDA	Redundant features	Channel + spatial attention	Fusion stage	Suppress redundant information

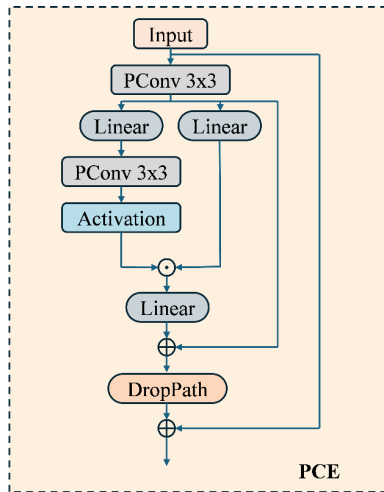


Fig. 2 PCE module.

$$U = W_2 X^{(1)} + b_2, \quad V = W_1 X^{(1)} + b_1 \quad (2)$$

where W_1 and W_2 are the weight matrices of the linear layers, respectively, and b_1 and b_2 are the corresponding bias terms.

Next, the feature V further undergoes a second partial convolution operation, using a new dynamic mask M' for fine-grained screening, and applying the ReLU activation function to introduce non-linear expression ability. The calculation process is as follows:

$$X^{(2)} = \text{ReLU}(\text{PConv}(V, M')) = \text{ReLU}\left(\frac{\text{Conv}(V \odot M')}{\sum_{i,j} M'_{i,j} + \varepsilon} \sum_{i,j} l\right) \quad (3)$$

To fuse the local convolution features and the linearly mapped features, the PCE module performs element-wise multiplication on U and $X^{(2)}$ for fusion, obtaining a composite feature representation:

$$X^{(3)} = U \odot X^{(2)} \quad (4)$$

The fused feature $X^{(3)}$ then goes through a linear mapping layer to further optimize the feature representation:

$$X^{(4)} = W_3 X^{(3)} + b_3 \quad (5)$$

To enhance the generalization ability of the model and suppress overfitting, the PCE module introduces the DropPath random dropout mechanism. Specifically, first, a residual connection is made with the output of the partial convolution to obtain:

$$\hat{Y} = X^{(4)} + X^{(1)} \quad (6)$$

$$Y' = \begin{cases} 0, & p \\ \hat{Y}, & 1-p \end{cases} \quad (7)$$

Finally, the module makes a second skip connection between the output processed by DropPath and the original input X_{in} to obtain the final output feature:

$$Y_{out} = Y' + X^{(0)} \quad (8)$$

Multi-scale context self-attention

As shown in Fig. 3, the MSCSA (Multi-Scale Context Self-Attention) module is designed to enhance the self-attention mechanism, inspired by transformer-based detection frameworks such as DETR^[8] and its variants^[15]. For the input feature of size $C \times H \times W$, a 1×1 convolution is first used to adjust the channel dimension. Subsequently, the feature is divided into three paths, and multi-scale local features are captured through 3×3 , 5×5 , and 7×7 depthwise separable convolutions (dwConv), respectively. This approach significantly reduces the number of parameters and computational cost required for standard convolution operations while maintaining high model expressiveness. Capturing multi-scale local features through convolution kernels of different scales helps improve the model's adaptability to targets of different sizes. Then, the feature maps obtained from the above three paths are added together and then pooled. The pooled output is processed by a 7×7 depthwise separable convolution and a Sigmoid function, and then fused with the processed global features, followed by a 1×1 convolution. Meanwhile, global features are extracted through average pooling (AvgPool). These features are sequentially processed by a 1×1 convolution, a ReLU activation function, and another 1×1 convolution. Finally, K (keys) and V (values) are generated through adaptive pooling. They allow the model to dynamically adjust its focus of attention based on the relationships between different parts of the input, thereby enhancing the model's ability to understand and correlate features of targets at different scales. This design enables the module to provide rich multi-scale context information for the subsequent attention mechanism, effectively enhancing the model's ability to understand and correlate features of targets at different scales.

The MSCSA module achieves efficient modeling of complex scenarios through the synergistic effect of multi-scale convolution, spatial attention, channel attention, and self-attention mechanisms. Its core calculation process is as follows:

First, the input features $X \in R^{C \times H \times W}$ are subjected to three different scales of depthwise convolution to extract the multi-scale features in a parallel manner. The three convolution kernel sizes are 3×3 , 5×5 , and 7×7 , and the resulting feature maps are fused in the channel dimension, denoted as:

$$D = \sum_{k \in \{3,5,7\}} \text{DWConv}_{k \times k}(X) \quad (9)$$

where $\text{DWConv}_{k \times k}(\cdot)$ denotes a depth-separable convolution operation with $k \times k$ kernels, and D is the fused multiscale feature map.

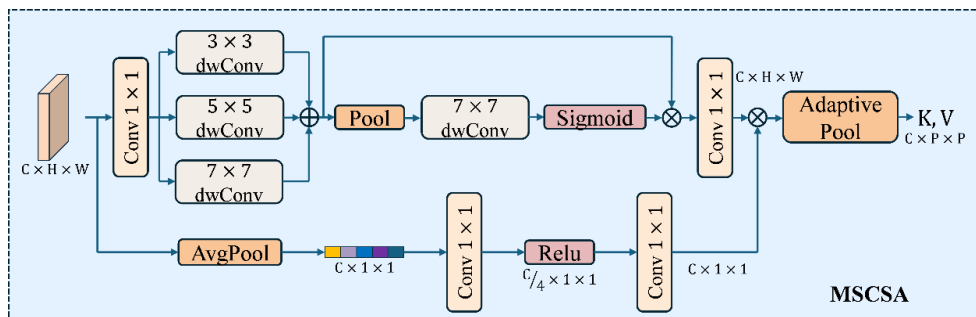


Fig. 3 MSCSA module.

In order to further model the feature responses at the spatial level, MSCSA introduces a spatial attention mechanism. Specifically, a pooling operation is first applied to D to compress the channel information, and then spatial context information is extracted using a 7×7 depth-separable convolution and the spatial attention map $S \in [0, 1]^{1 \times H \times W}$ is generated by a Sigmoid activation function with the following equation:

$$S = \sigma(\text{DWConv}_{7 \times 7}(\text{Pool}(D))) \quad (10)$$

where $\sigma(\cdot)$ denotes the Sigmoid function and $\text{Pool}(\cdot)$ is a pooling operation (e.g., maximum pooling or average pooling) for keeping the feature size constant. The spatial attention map S is able to adaptively enhance important regions while suppressing background interference.

Meanwhile, to capture the inter-channel dependencies, the module introduces a channel attention mechanism. Firstly, the input features X are globally average-pooled to compress the spatial dimensions, followed by a two-layer 1×1 convolution to achieve the compression and recovery of the channel dimensions, and the Sigmoid function to generate the channel attention weights $F \in [0, 1]^{C \times 1 \times 1}$. The specific process is denoted as:

$$F = \sigma(W_2 \text{ReLU}(W_1 \text{AvgPool}(X))) \quad (11)$$

where $\text{AvgPool}(\cdot)$ denotes the global average pooling operation, $W_1 \in \mathbb{R}^{\frac{C}{4} \times C}$ and $W_2 \in \mathbb{R}^{C \times \frac{C}{4}}$ are the learnable weight matrices used for channel compression and expansion, respectively, and ReLU is the activation function.

Next, spatial attention and channel attention are weighted and summed to the input feature X , respectively, to obtain the fused feature. The information is then further fused by a 1×1 convolution to form the final enhanced feature representation X' with the following formula:

$$X' = \text{Conv}_{1 \times 1}(X \odot S + X \odot F) \quad (12)$$

where $\odot$ denotes element-by-element multiplication (broadcasting mechanism). This fusion effectively combines the importance assessment of the spatial and channel dimensions, allowing the model to better focus on key features.

Finally, the augmented feature X' is channel-transformed by another layer of 1×1 convolution, and Key and Value features for subsequent attention mechanisms (e.g., self-attention or interaction modules) are generated by an Adaptive Pooling operation:

$$K, V = \text{AdaptivePool}(\text{Conv}_{1 \times 1}(X')) \quad (13)$$

where K and V are generated by adaptive pooling for subsequent self-attention computation.

AGU

In the feature fusion stage of RT-DETR^[15], low-resolution semantic features need to be aligned with higher-resolution spatial features. In UAV airport images, this process is particularly important because small vehicles, personnel, and partially occluded aircraft structures may occupy only limited image regions. Conventional upsampling operations restore feature resolution using fixed interpolation rules, which may weaken boundary details and introduce inaccurate feature alignment in dense airport scenes. To improve the recovery of spatial details while maintaining computational efficiency, this study introduces Adaptive Grid Upsampling (AGU) into the cross-scale feature fusion process. Inspired by dynamic point-sampling strategies^[35], AGU learns adaptive offsets for sampling locations and adjusts the sampling range according to feature content, thereby improving feature alignment for small and

scale-varying targets. The overall structure of the AGU module is shown in Fig. 4.

Feature encoding: Learn offsets through convolution to provide adaptive sampling point position adjustment relative to the standard grid. Since the offset of each position is dynamically learned, it can adapt to the information requirements of different regions, especially the edge regions. Apply a 2D convolution to the low-resolution feature map $X \in \mathbb{R}^{H \times W \times C}$ to learn the preliminary feature offset information:

$$\Delta P = f_{\theta}(X) \in \mathbb{R}^{H \times W \times 2gs^2} \quad (14)$$

where $f_{\theta}(\cdot)$ is a convolutional neural network (CNN), and its parameters are learned through training; ΔP represents the predicted dynamic offset, with a dimension of $H \times W \times 2gs^2$, where g is the number of groups and s is the upsampling scale factor; each pixel predicts $2gs^2$ offset values (representing the two-dimensional coordinate offsets $\Delta x, \Delta y$ of gs^2 sampling points).

Dynamic offset prediction: Use a grouping strategy to calculate the offset information of different regions and generate new sampling positions through normalized coordinates. To make the sampling point distribution adapt to the input features, we use standard normalized coordinates:

$$P_{\text{grid}} = \left(\frac{2i}{W} - 1, \frac{2j}{H} - 1 \right), \quad i = 0, \dots, W-1, \quad j = 0, \dots, H-1 \quad (15)$$

$$P_{\text{sample}} = P_{\text{grid}} + \Delta P \quad (16)$$

where P_{grid} is the standardized grid coordinate, with a range between $[-1, 1]$; and P_{sample} is the sampling position after adding the dynamic offset. Mapping the normalized coordinates to the range of $[-1, 1]$ allows the offsets to be adjusted on a unified scale. By dynamically predicting the offset ΔP , the distribution of sampling points can be changed, enhancing the feature capture ability of different regions.

Adaptive scale regulation: A dynamic range factor S_{dyn} is introduced, which can adaptively adjust the scale of the offset according to the feature content. This helps to reduce the sampling overlap problem and enhance the information capture ability in edge regions:

$$S_{\text{dyn}} = \sigma(W_s * X) \quad (17)$$

$$\Delta P' = S_{\text{dyn}} \cdot \Delta P \quad (18)$$

where W_s is an additional convolution weight used to learn the range factor; $\sigma(\cdot)$ is the Sigmoid activation function, which ensures that S_{dyn}

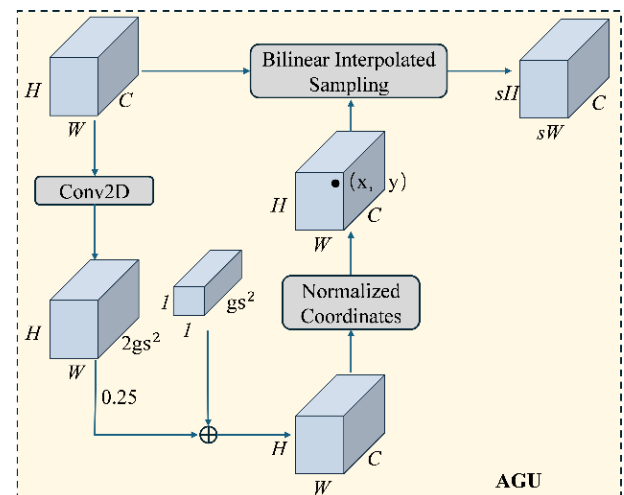


Fig. 4 AGU module.

is within the range of (0, 1). Scaling by S_{dyn} can adjust the magnitude of the offset to avoid over-concentration or over-dispersion of sampling points.

Grid-based bilinear interpolation sampling: Bilinear interpolation can be used to smoothly obtain feature values and avoid information loss caused by sampling at non-integer positions. Combining with dynamic offsets enables the feature map to be more accurately aligned to appropriate sampling point positions. Since the sampling points P_{sample} may fall between the pixel grids, bilinear interpolation is used to calculate the final features:

$$Y(x, y) = \sum_{m=0}^1 \sum_{n=0}^1 w_{mn} X(x_m, y_n) \quad (19)$$

$$w_{mn} = (1 - |x - x_m|)(1 - |y - y_n|) \quad (20)$$

where (x_m, y_n) represents the four nearest neighboring pixel points, w_{mn} is the weight calculated based on the distance, and $Y(x, y)$ is the new feature map obtained through the final interpolation.

Channel-Spatial Dual Attention module

Traditional fusion modules are mainly used for cross-scale feature fusion to enhance the detector's perception ability of targets at different scales. However, ordinary feature fusion methods usually adopt convolution or weighted summation, making it difficult to capture global information in both the channel and spatial dimensions simultaneously. This paper proposes the Channel-Spatial Dynamic Attention (CSDA) module to replace the Fusion module in RT-DETR, aiming to enhance the modeling ability of multi-scale features and optimize computational efficiency at the same time. The CSDA (Channel-Spatial Dynamic Attention) module aims to optimize the feature fusion process in RT-DETR to improve the performance of object detection. Its core idea is to combine channel attention and spatial attention, inspired by attention mechanisms widely used in computer vision^[36], thereby enhancing the ability to extract key information. First, the module extracts channel-level features through global average pooling and calculates channel weights using a non-linear transformation to highlight useful information and suppress redundant interference. Then, a spatial attention mechanism is used to calculate the correlation between feature points, enabling the model to dynamically adjust the weights of different regions and enhance the spatial perception ability of targets. Finally, the feature maps after fusing channel and spatial attention are combined to achieve a more accurate target representation. This module effectively alleviates the over-dependence of traditional fusion methods on high-resolution features and reduces computational redundancy, enabling the detector to more efficiently detect small and dense targets in complex airport scenarios. The specific network structure is shown in Fig. 5. The core idea is as follows:

Calculate the importance of channel-level features through global average pooling (GAP), enabling the network to adaptively adjust the weights of each channel to enhance the focus on key features. For the input feature map $X \in \mathbb{R}^{H \times W \times C}$, calculate the importance of each channel through global average pooling:

$$X_{gap} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W X_{c,i,j} \quad (21)$$

Then, the channel weights S_c are calculated through a fully connected layer:

$$S_c = \sigma(W_2 \delta(W_1 X_{gap})) \quad (22)$$

where $\delta(\cdot)$ is the ReLU activation function; and $\sigma(\cdot)$ is the Sigmoid normalization function.

Rearrange the input feature X into a matrix form and calculate the spatial attention matrix. Then, normalize it using the Softmax function to ensure the stability of the attention weights:

$$A = \text{softmax}(X^T X) \quad (23)$$

Then, calculate the weighted features:

$$X_{att} = AX \quad (24)$$

where A is the spatial attention matrix that represents the correlation between feature points.

Finally, fuse the channel attention and spatial attention to obtain the final output:

$$Y = X_{att} \cdot S_c \quad (25)$$

where $Y \in \mathbb{R}^{C \times H \times W}$ is the final feature map. This feature contains both global channel information and enhances the key regions at the spatial level.

Experiment

Datasets and preprocessing

In this study, the AFTD (Airport Surface Foreground Target Detection) dataset is selected as the experimental verification object. AFTD is an application-oriented dataset for monitoring moving targets on the airport surface^[37]. Currently, the AFTD dataset consists of 10,050 image data, which includes 25,968 aircraft instances, 24,388 vehicle instances, and 5,745 person instances. It covers various challenges in airport surface movement target detection, such as extreme multi-scale, multi-level occlusion, and viewpoint changes. A total of 6,868 images are selected for the training set, 1,963 for the validation set, and 982 for the test set. There are three types of objects and various image sizes, and the distribution of the dataset is shown in Fig. 6.

Experimental validation

Setting

The experimental platform processor for this paper is the Intel(R) Core(TM) i7-14700KF 20-Core processor; the GPU uses the RTX 4090 with a system video memory size of 24 GB; the operating system is Ubuntu 20.04, and the programming language is Python 3.8 with PyTorch 1.13.0 architecture. As shown in Table 2, the main training parameters include 300 epochs, a batch size of 16, an input image size of 640, and the AdamW optimizer with an initial learning rate of 0.0001.

In this paper, we use the commonly used detection performance metrics (mean average precision mAP@50, mAP@50-95, precision P

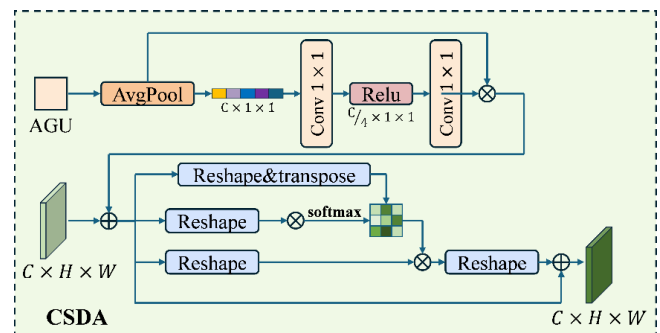


Fig. 5 CSDA module.

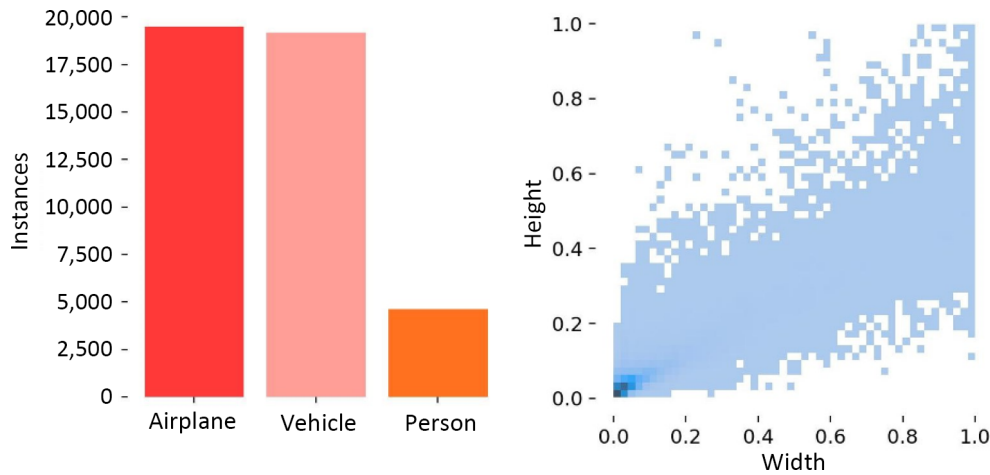


Fig. 6 Target category distribution and size diagram.

Table 2. Setting table of experimental environment parameters.

Parameters	Settings
Epochs	300
Batch	16
Imgsz	640
Workers	4
Optimizer	AdamW ^[38]
Lr0	0.0001

[Precision], recall R [Recall]) and model efficiency metrics (number of model parameters Paras, inference speed FPS) in the field of target detection to comprehensively validate the detection effect of the model. Among them, $mAP@50$ represents the average precision mean value when the intersection and merger ratio threshold is 0.5, $mAP@50-95$ reflects the average detection precision under different intersection and merger ratio thresholds from 0.5 to 0.95, and FPS quantifies the real-time inference capability of the model by processing frames per second.

Comparison experiment

This experiment evaluates the performance of various target detection algorithms on the airport field target detection dataset with respect to precision (P), recall (R), mean average precision

(mAP) of different IoU thresholds ($mAP_{0.5}$, $mAP_{0.5:0.95}$), the number of model parameters (Param), the number of floating-point operations per second (GFLOPs), and the number of frames per second (FPS). metrics. The comparison algorithms cover the classical YOLO family (YOLOv8^[23], YOLOv9^[24], YOLOv10s^[25], YOLOv11^[39], DETR^[8], RT-DETR^[15]), DETR based on the transformer architecture, RT-DETR, and the AeroRTD algorithm proposed in this paper.

As shown in Fig. 7, in the initial training phase (Epoch 0–50), the $mAP_{0.5}$ values of all algorithms grow rapidly, which indicates that they can effectively capture the basic features of the airport field targets; the YOLO series algorithms have similar growth curves due to the similarity of their architectures and initialization settings, while the DETR algorithms are based on the transformer architecture, with a unique training mechanism, which makes the adaptation of data distribution more complicated and more fluctuating in the initial phase. DETR is based on the transformer architecture; the training mechanism is unique, the initial adaptation to data distribution is complicated, and the fluctuation of $mAP_{0.5}$ is large. The trend of $mAP_{0.5:0.95}$ indicator is similar. In the middle of training (Epoch 50–200), the growth rate of $mAP_{0.5}$ slows down. The AeroRTD algorithm in this paper has a significant lead in $mAP_{0.5}$ by virtue of its unique structure and optimisation strategy designed for airport scenes, compared to RT-DETR. YOLOv11 has a slight advantage in the YOLO series, and the growth of DETR is flat and fluctuates continuously. In terms of $mAP_{0.5:0.95}$, AeroRTD grows rapidly,

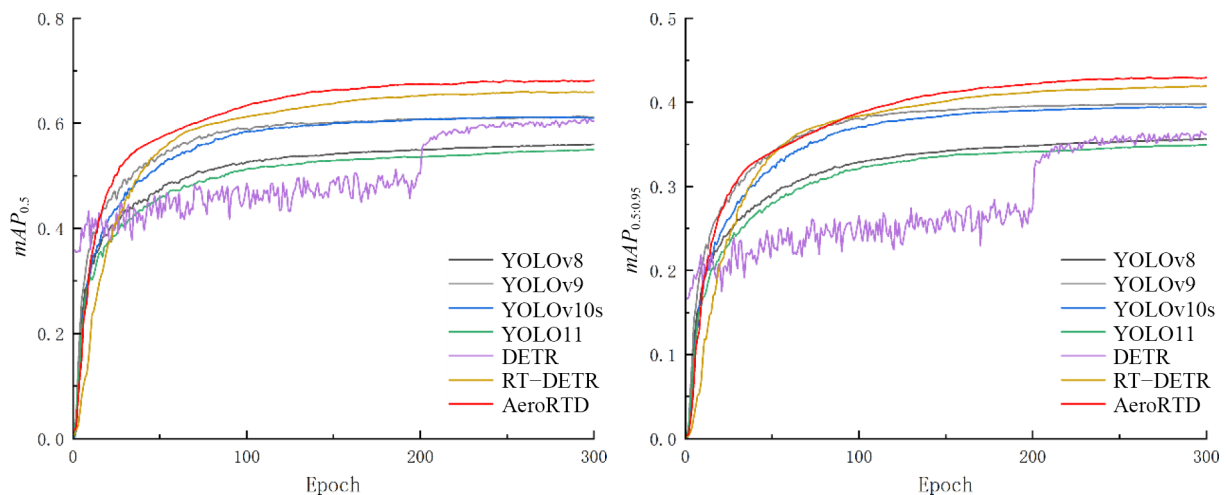


Fig. 7 Comparative experimental chart.

highlighting its excellent performance in complex scenarios and strict IoU threshold detection, followed by YOLOv11, which struggles with DETR. At the late stage of training (Epoch 200–300), most algorithms' $mAP_{0.5}$ and $mAP_{0.5:0.95}$ gradually stabilise, while AeroRTD maintains the lead, showing good stability and generalisation ability.

As shown in Table 3, AeroRTD has the best precision (0.758) and recall (0.635), which indicates that it can identify the target more accurately and miss fewer targets; RT-DETR has the second-best precision (0.754) and recall (0.612), while YOLOv9 and YOLOv10s also have good performance in terms of precision and recall. In terms of the number of parameters, DETR is as high as 36,739,785, which is a complex model, while AeroRTD is 15,762,572, which is relatively lightweight and has excellent performance. GFLOPs reflect the amount of operations, with DETR at 100.94, which is a heavy burden of operations, and AeroRTD at 50.9, which is moderate in terms of the amount of operations. In terms of inference speed, AeroRTD (ours) leads with 23.89 FPS, balancing detection, accuracy, and speed; RT-DETR is 22.59 FPS, which is also good for real-time performance; in the YOLO series, YOLOv8 is 13.14 FPS, YOLOv9 is 10.77 FPS, YOLOv10s is 11.96 FPS, YOLOv11 is 12.96 FPS, DETR is 11.85 FPS, and the inference speed is relatively slow.

To further evaluate the generalization capability of the proposed AeroRTD, we conduct additional experiments on the public VisDrone2019 dataset. This dataset contains a large number of small

and densely distributed objects captured in UAV scenarios, which is significantly different from the AFTD dataset.

For the VisDrone2019 experiments, we used the object detection subset and followed the official data split. The training set was used for model optimization, the validation set was used for performance evaluation during training, and the test set was used for final comparison. Unless otherwise specified, all compared methods were trained and evaluated under the same experimental environment as the AFTD experiments, using an input size of 640, a batch size of 16,300 training epochs, the AdamW optimizer, and an initial learning rate of 0.0001. The experiments were conducted on the same workstation equipped with an Intel Core i7-14700KF CPU and an NVIDIA RTX 4090 GPU with 24 GB of memory, ensuring a fair comparison under identical hardware and training settings.

As shown in Table 4, AeroRTD achieves competitive performance compared with other state-of-the-art methods, demonstrating its strong generalization ability across different datasets and application scenarios.

In order to visually compare the regions of interest of YOLOv8, RT-DETR, and the AeroRTD proposed in this paper on target detection, we visualised the heatmaps of their responses on the same test image. The heatmaps were all generated using GradCAMPlusPlus with consistent parameter settings to ensure fairness of comparison. As shown in Fig. 8, YOLOv8's activation hotspots are more dispersed, especially at both the background and target edges, which tends to introduce noise and lead to an increased risk of false detection. RT-DETR is more responsive in the main body area, but still generates large hotspots at the tail or the bottom of the aircraft, which suggests that it is not yet accurate enough to localise the

Table 3. Comparison parameter table.

Modules	<i>P</i>	<i>R</i>	$mAP_{0.5}$	$mAP_{0.5:0.95}$	Params	GFLOPs	FPS
YOLOv8 ^[23]	0.701	0.518	0.56	0.357	3,006,233	8.1	13.14
YOLOv9 ^[24]	0.743	0.567	0.613	0.399	7,168,249	26.7	10.77
YOLOv10s ^[25]	0.734	0.567	0.611	0.394	8,037,282	24.5	11.96
YOLOv11 ^[39]	0.659	0.521	0.55	0.35	2,582,737	6.3	12.96
DETR ^[8]	–	–	0.604	0.362	36,739,785	100.94	11.85
RT-DETR ^[15]	0.754	0.612	0.659	0.419	19,875,612	56.9	22.59
AeroRTD (ours)	0.758	0.635	0.681	0.43	15,762,572	50.9	23.89

Bold values represent the best-performing results for the corresponding metrics.

Table 4. Performance comparison on VisDrone2019 dataset.

Modules	$mAP_{0.5}$ (val)	$mAP_{0.5:0.95}$ (val)	$mAP_{0.5}$ (test)	$mAP_{0.5:0.95}$ (test)
YOLOv8s ^[23]	0.411	0.248	0.333	0.193
YOLOv11s ^[39]	0.408	0.245	0.328	0.192
RT-DETR ^[15]	0.474	0.281	0.369	0.229
AeroRTD	0.479	0.292	0.373	0.232

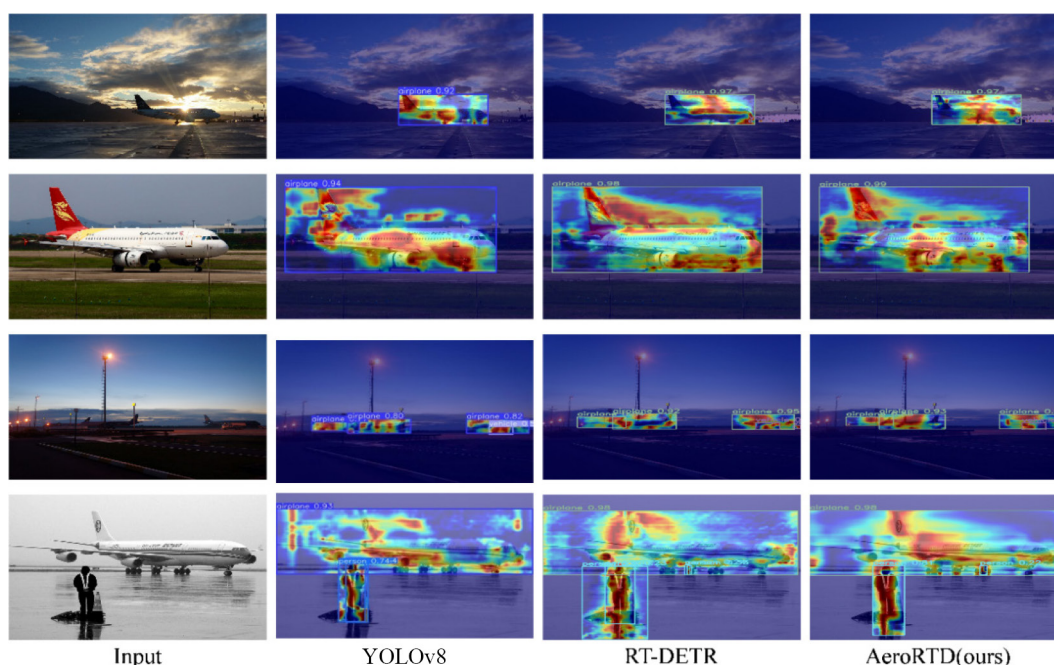


Fig. 8 Visualization results.

details. The thermograms of AeroRTD (ours), on the other hand, show the most focused activation: almost all the high-intensity response is focused on the key structures of the target (the centre of the fuselage and the nose area), while the response to the background and non-target areas is effectively suppressed. In the third row of samples (small aircraft at night), the activation signals of YOLOv8 and RT-DETR are weak and scattered for small targets, whereas AeroRTD gives a clear, focused, and higher-confidence (see labelling in the figure) response, even in low light and with a small target size. From the confidence labels on the thermograms, AeroRTD's detection confidence is slightly higher or equal to other methods in all cases (e.g., 0.92 to 0.97 in the first row and 0.98 to 0.99 in the second row), and its thermograms are much more interpretable. AeroRTD can provide a more focused response in a variety of complex scenarios than YOLOv8 and RT-DETR. AeroRTD, compared with YOLOv8 and RT-DETR, can provide more focused and accurate activation heat maps in various complex scenarios, accompanied by higher detection confidence, which proves the advantages of this method in terms of localisation accuracy and robustness.

Although the AFTD dataset is a publicly available benchmark for airport surface target detection, it does not provide an official scale-based categorization (e.g., small, medium, and large objects) as defined in datasets such as COCO. Therefore, a strictly standardized scale-wise quantitative evaluation cannot be directly conducted in this study.

Nevertheless, the AFTD dataset inherently contains significant scale variations. For example, aircraft may occupy a large portion of the image, while ground personnel and service vehicles often appear as small or densely distributed targets. This characteristic makes it particularly suitable for evaluating the robustness of detection models under scale variation.

From the experimental results and visualization analysis, AeroRTD demonstrates strong performance in handling such challenges. In particular, the proposed method shows clear advantages in detecting small and densely distributed targets, as evidenced by the improved mAP performance and the more focused activation regions in the visualization results. This improvement can be attributed to the PCE module, which enhances fine-grained feature representation, and the MSCSA and AGU modules, which effectively model multi-scale contextual information and improve cross-scale feature alignment.

These observations indicate that, although explicit scale-wise metrics are not available, the proposed AeroRTD framework exhibits robust performance across a wide range of target scales in complex airport scenarios.

Ablation results

In order to systematically verify the function of each core component in the proposed model, this study designs an ablation experiment system. By selectively removing the key modules of the model, the experiment compares the performance of the model with different component configurations while ensuring that the

network architecture, training hyperparameters (learning rate, iteration rounds, etc.), and dataset partitioning rules are identical. The ablation results of different module combinations are shown in Table 5, where A, B, and C denote the PCE module, MSCSA module, and the AGU + CSDA fusion strategy, respectively.

Compared to the RT-DETR benchmark model, adding components gradually improves the precision and recall of the model. After adding component PCE, the precision improves from 0.754 to 0.757, and the recall improves from 0.612 to 0.624, indicating that A can enhance the model's ability to identify and detect targets. Further adding component B, the degree improves to 0.76, confirming the positive effect of B in enhancing the targeting of feature extraction. After adding all components, the recall rate increases significantly to 0.635, and the precision is maintained at 0.758, indicating that component C can help the model to identify more targets and maintain accuracy.

On the $mAP_{0.5}$ metric, the baseline model is 0.659, which continues to improve with the addition of components and reaches 0.681 with the addition of all components. On the $mAP_{0.5:0.95}$ metric, the baseline model is 0.419, which rises with the addition of component A, decreases slightly with the addition of both A and B, and reaches 0.43 with the addition of all components, indicating that component C is crucial to improving the detection accuracy under the strict IoU threshold and enhances the model's accuracy in complex scenarios. It shows that component C is crucial for improving the detection accuracy under strict IoU thresholds, which enhances the model's ability to control the target scale and localisation accuracy in complex scenarios.

In terms of the number of model parameters, the number of GFLOPs decreases significantly after adding component A, and increases slightly after adding component B and all components, but the overall level stays low. In terms of GFLOPs, the computational burden decreases after adding component A and increases slightly after adding component B and all components, which is still within a reasonable range. In terms of inference speed, FPS is increased after adding component A and maintained at 23.89 after adding all components, maintaining high real-time performance.

In summary, the core components A, B, and C can effectively improve the model performance, not only significantly improving the detection precision and recall rate but also reasonably controlling the model complexity and maintaining a high inference speed, which verifies the effectiveness and necessity of these components in model optimisation.

To further analyze the contribution of each module, we examine the incremental performance gains brought by each component. It can be observed that the introduction of the PCE module leads to a noticeable improvement, indicating its effectiveness in enhancing fine-grained feature representation. The MSCSA module further improves performance by strengthening multi-scale contextual modeling, which is particularly beneficial for handling scale variation.

In contrast, the AGU and CSDA modules contribute more moderate but consistent improvements, mainly by refining feature alignment and enhancing feature fusion efficiency. These results suggest

Table 5. Table of ablation experiments.

Modules	<i>P</i>	<i>R</i>	$mAP_{0.5}$	$mAP_{0.5:0.95}$	Params	GFLOPs	FPS
RT-DETR (baseline)	0.754	0.612	0.659	0.419	19,875,612	56.9	22.59
Baseline + A	0.757	0.624	0.669	0.429	15,405,804	46.2	25.47
Baseline + A + B	0.76	0.624	0.675	0.424	15,726,988	50.7	23.03
Baseline + A + B + C	0.758	0.635	0.681	0.43	15,762,572	50.9	23.89

that while all modules are beneficial, their contributions are not equally significant, and they play complementary roles in improving the overall detection performance.

Conclusions

This paper proposes AeroRTD, an enhanced real-time detection framework for UAV-based airport target detection under complex conditions. Built upon RT-DETR, the proposed method improves feature representation, contextual modeling, and cross-scale fusion, enabling robust detection of small, densely distributed, and occluded targets. Experimental results on the AFTD dataset demonstrate that AeroRTD achieves superior detection performance while maintaining a lightweight architecture and efficient inference speed. These results validate the effectiveness and practicality of the proposed approach for real-world airport monitoring applications. Future work will focus on incorporating multi-modal sensing (e.g., infrared and radar) and semantic prior information to further enhance robustness under challenging conditions such as low illumination and adverse weather.

Author contributions

The authors confirm their contributions to the paper as follows: study conception and design: Qu X, Wang Xichao; data collection: Zhang J, Jiang Y; analysis and interpretation of results: Zhang J, Liu S; draft manuscript preparation: Wang Xia. All authors reviewed the results and approved the final version of the manuscript.

Data availability

The VisDrone2019 dataset used in this study is publicly available at: <https://github.com/VisDrone/VisDrone-Dataset>. The dataset used in this study (AFTD) is publicly available. The data can be accessed from the corresponding published source. No additional data were generated during the current study.

Acknowledgments

This work was supported by the Artificial Intelligence Promotes Paradigm Reform in Scientific Research and Empowers Discipline Advancement Plan: Research on System Engineering Modeling Methods Based on Generative AI under Grant No. 25AZ015.

Conflict of interest

The authors declare that they have no conflict of interest.

Dates

Received 30 January 2026; Revised 3 April 2026; Accepted 16 June 2026; Published online 29 September 2026

References

- [1] Batalov AA. 2021. Global aviation safety oversight system: challenges and opportunities for ICAO and its member states. *Air and Space Law* 46:99–118
- [2] Mwesiumo D, Halpern N, Bråthen S, Budd T, Suau-Sanchez P. 2023. Perceived benefits as a driver and necessary condition for the willingness of air passengers to provide personal data for non-mandatory digital services at airports. *Transportation Research Part A: Policy and Practice* 171:103659
- [3] Yi P, Li J, Zhang Y, Chen Y. 2023. Efficient hierarchical signature scheme with batch verification function suitable for ADS-B system. *IEEE Transactions on Aerospace and Electronic Systems* 59(2):1292–1299
- [4] Wang YQ. 2014. An analysis of the Viola-Jones face detection algorithm. *Image Processing on Line* 4:128–148
- [5] Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D. 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(9):1627–1645
- [6] Wang S, Hu X, Sun J, Liu J. 2023. Hyperspectral anomaly detection using ensemble and robust collaborative representation. *Information Sciences* 624:748–760
- [7] Huang Q, Wang D, Lu Z, Zhou S, Li J, et al. 2023. A novel image-to-knowledge inference approach for automatically diagnosing tumors. *Expert Systems with Applications* 229:120450
- [8] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, et al. 2020. End-to-end object detection with transformers. *Computer Vision – ECCV 2020: European conference on computer vision*. Cham: Springer International Publishing. pp. 213–229 doi: [10.1007/978-3-030-58452-8_13](https://doi.org/10.1007/978-3-030-58452-8_13)
- [9] Chen Y, Xu H, Zhang X, Gao P, Xu Z, et al. 2023. An object detection method for bayberry trees based on an improved YOLO algorithm. *International Journal of Digital Earth* 16(1):781–805
- [10] Guan L, Jia L, Xie Z, Yin C. 2022. A lightweight framework for obstacle detection in the railway image based on fast region proposal and improved YOLO-tiny network. *IEEE Transactions on Instrumentation and Measurement* 71:5009116
- [11] Yao H, Liu Y, Li X, You Z, Feng Y, et al. 2022. A detection method for pavement cracks combining object detection and attention mechanism. *IEEE Transactions on Intelligent Transportation Systems* 23(11):22179–22189
- [12] Kang L, Lu Z, Meng L, Gao Z. 2024. YOLO-FA: type-1 fuzzy attention based YOLO detector for vehicle detection. *Expert Systems with Applications* 237:121209
- [13] Roh MC, Lee JY. 2017. Refining faster-RCNN for accurate object detection. *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*. May 8–12, 2017, Nagoya, Japan. USA: IEEE. pp. 514–517 doi: [10.23919/mva.2017.7986913](https://doi.org/10.23919/mva.2017.7986913)
- [14] Ren Y, Zhu C, Xiao S. 2018. Object detection based on fast/faster RCNN employing fully convolutional architectures. *Mathematical Problems in Engineering* 2018(1):3598316
- [15] Zhao Y, Lv W, Xu S, Wei J, Wang G, et al. 2024. Detrs beat yolos on real-time object detection. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. Jun 17th–21st, 2024, Seattle WA, USA. USA: IEEE. pp. 16965–16974 doi: [10.1109/CVPR52733.2024.01605](https://doi.org/10.1109/CVPR52733.2024.01605)
- [16] Ren S, He K, Girshick R, Sun J. 2017. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6):1137–1149
- [17] Liu W, et al. 2016. SSD: single shot multibox detector. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I* 14. Cham: Springer. pp. 21–37 doi: [10.1007/978-3-319-46448-0_2](https://doi.org/10.1007/978-3-319-46448-0_2)
- [18] Redmon J, Divvala S, Girshick R, Farhadi A. 2016. You only look once: unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 27–30, 2016, Las Vegas, NV, USA. USA: IEEE. pp. 779–788 doi: [10.1109/cvpr.2016.91](https://doi.org/10.1109/cvpr.2016.91)
- [19] Redmon J, Farhadi A. 2017. YOLO9000: better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 21–26, 2017, Honolulu, HI, USA. USA: IEEE. pp. 6517–6525 doi: [10.1109/cvpr.2017.690](https://doi.org/10.1109/cvpr.2017.690)
- [20] Redmon J, Farhadi A. 2018. YOLOv3: an incremental improvement. *arXiv Preprint*
- [21] Bochkovskiy A, Wang CY, Liao HM. 2020. YOLOv4: optimal speed and accuracy of object detection. *arXiv Preprint:2004.10934*
- [22] Ge Z, Liu S, Wang F, Li Z, Sun J. 2021. YOLOX: exceeding YOLO series in 2021. *arXiv Preprint:2107.08430*
- [23] Varghese R, Sambath M. 2024. YOLOv8: a novel object detection algorithm with enhanced performance and robustness. *2024 International*

- Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*. April 18–19, 2024, Chennai, India. USA: IEEE. pp. 1–6 doi: [10.1109/ADICS58448.2024.10533619](https://doi.org/10.1109/ADICS58448.2024.10533619)
- [24] Wang CY, Yeh IH, Mark Liao HY. 2024. YOLOv9: learning what you want to learn using programmable gradient information. In *Computer Vision – ECCV 2024*. Cham, Switzerland: Springer Nature. pp. 1–21 doi: [10.1007/978-3-031-72751-1_1](https://doi.org/10.1007/978-3-031-72751-1_1)
- [25] Wang A, Chen H, Liu L, Chen K, Lin Z, et al. 2024. YOLOv10: real-time end-to-end object detection. *Advances in Neural Information Processing Systems (NeurIPS 2024)*. vol. 37. pp. 107984–108011 doi: [10.52202/079017-3429](https://doi.org/10.52202/079017-3429)
- [26] Xu S, Wang X, Lv W, Chang Q, Cui C, et al. 2022. PP-YOLOE: an evolved version of YOLO. *arXiv Preprint*
- [27] Zhu X, Su W, Lu L, Li B, Wang X, et al. 2020. Deformable DETR: deformable transformers for end-to-end object detection. *arXiv Preprint:2010.04159*
- [28] Gao P, Zheng M, Wang X, Dai J, Li H. 2022. Fast convergence of DETR with spatially modulated co-attention. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. October 10–17, 2021, Montreal, Canada. USA: IEEE. pp. 3601–3610 doi: [10.1109/ICCV48922.2021.00360](https://doi.org/10.1109/ICCV48922.2021.00360)
- [29] Meng D, et al. 2021. Conditional DETR for fast training convergence. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 10–17 October 2021, Montreal, QC, Canada. USA: IEEE. pp. 3651–3660 doi: [10.1109/ICCV48922.2021.00363](https://doi.org/10.1109/ICCV48922.2021.00363)
- [30] Li F, Zhang H, Liu S, Guo J, Ni LM, et al. 2022. DN-DETR: accelerate DETR training by introducing query denoising. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46(4):2239–2251
- [31] Wang Y, Zhang X, Yang T, Sun J. 2022. Anchor DETR: query design for transformer-based detector. *Proceedings of the AAAI Conference on Artificial Intelligence* 36(3):2567–2575
- [32] Zhang H, Li F, Liu S, Zhang L, Su H, et al. 2022. DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection. *arXiv Preprint:2203.03605*
- [33] Ouyang L, Guo G, Fang L, Ghamisi P, Yue J. 2023. PCLDet: prototypical contrastive learning for fine-grained object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 61:5613911
- [34] Koonce B. 2021. EfficientNet. In *Convolutional Neural Networks with Swift for Tensorflow*. Berkeley, CA: Apress. pp. 109–123 doi: [10.1007/978-1-4842-6168-2_10](https://doi.org/10.1007/978-1-4842-6168-2_10)
- [35] Liu W, Lu H, Fu H, Cao Z. 2024. Learning to upsample by learning to sample. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 1–6 October 2023, Paris, France. USA: IEEE. pp. 6004–6014 doi: [10.1109/ICCV51070.2023.00554](https://doi.org/10.1109/ICCV51070.2023.00554)
- [36] Woo S, Park J, Lee JY, Kweon IS. 2018. CBAM: convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Cham: Springer. pp 3–19 doi: [10.1007/978-3-030-01234-2_1](https://doi.org/10.1007/978-3-030-01234-2_1)
- [37] Chen P, Zhang X, Lin W, Huang Y, Yang C, et al. 2024. AFTD: a foreground target detection dataset for airport scenario. <https://openreview.net/forum?id=UpPXWd9SBk>
- [38] Zhang Z. 2019. Improved Adam optimizer for deep neural networks. *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. June 4–6, 2018, Banff, AB, Canada. USA: IEEE. pp. 1–2 doi: [10.1109/IWQoS.2018.8624183](https://doi.org/10.1109/IWQoS.2018.8624183)
- [39] Khanam R, Hussain M. 2024. YOLOv11: an overview of the key architectural enhancements. *arXiv Preprint:2410.17725*



Copyright: © 2026 by the author(s). Published by Maximum Academic Press, Fayetteville, GA. This article is an open access article distributed under Creative Commons Attribution License (CC BY 4.0), visit <https://creativecommons.org/licenses/by/4.0/>.