mathematics

MDPI

Article

Fast Identification and Detection Algorithm for Maneuverable Unmanned Aircraft Based on Multimodal Data Fusion

Tian Luan [1,2], Shixiong Zhou [1,2], Yicheng Zhang [3,*] and Weijun Pan [1,2,*]

[1] Civil Aviation Flight Technology and Flight Safety Engineering Technology Research Institute of Sichuan Province, Civil Aviation Flight University of China, Deyang 618307, China; luantian@cafuc.edu.cn (T.L.); zsx@cafuc.edu.cn (S.Z.)
[2] College of Air Traffic Management, Civil Aviation Flight University of China, Deyang 618307, China
[3] Institute for Infocom Research (I2R) at the Agency for Science, Technology and Research (A*STAR), Singapore 138632, Singapore
* Correspondence: zhang_yicheng@i2r.a-star.edu.sg (Y.Z.); wjpan@cafuc.edu.cn (W.P.)

Abstract: To address the critical challenges of insufficient monitoring capabilities and vulnerable defense systems against drones in regional airports, this study proposes a multi-source data fusion framework for rapid UAV detection. Building upon the YOLO v11 architecture, we develop an enhanced model incorporating four key innovations: (1) A dual-path RGB-IR fusion architecture that exploits complementary multi-modal data; (2) C3k2-DATB dynamic attention modules for enhanced feature extraction and semantic perception; (3) A bilevel routing attention mechanism with agent queries (BRSA) for precise target localization; (4) A semantic-detail injection (SDI) module coupled with windmill-shaped convolutional detection heads (PCHead) and Wasserstein Distance loss to expand receptive fields and accelerate convergence. Experimental results demonstrate superior performance with 99.3% mAP@50 (17.4% improvement over baseline YOLOv11), while maintaining lightweight characteristics (2.54M parameters, 7.8 GFLOPS). For practical deployment, we further enhance tracking robustness through an improved BoT-SORT algorithm within an interactive multiple model framework, achieving 91.3% MOTA and 93.0% IDF1 under low-light conditions. This integrated solution provides cost-effective, high-precision drone surveillance for resource-constrained airports.

Keywords: UAV; detect and avoid; object tracking; unti-UAV

MSC: 90-10

check for updates

Academic Editor: Xiaosong Du

Received: 22 April 2025
Revised: 21 May 2025
Accepted: 27 May 2025
Published: 30 May 2025

Citation: Luan, T.; Zhou, S.; Zhang, Y.; Pan, W. Fast Identification and Detection Algorithm for Maneuverable Unmanned Aircraft Based on Multimodal Data Fusion. *Mathematics* **2025**, *13*, 1825. https://doi.org/10.3390/math13111825

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As a core force in the development of low-altitude economy, drones have been widely used in a variety of fields such as emergency rescue, geographic mapping and so on, bringing a lot of convenience to the development of society, but at the same time triggering a series of safety hazards, especially in terms of aviation safety. In recent years, the frequent occurrence of illegal intrusion of drones into the airspace of airport terminal areas has become one of the most important factors threatening the safety of civil aviation. For example, in December 2018, Gatwick Airport in the United Kingdom was interfered by a drone, halting operations for more than 30 h and stranding 120,000 passengers due to the proximity of the Christmas holiday [1]. On the night of 11 September 2024, Tianjin Binhai Airport in China was invaded by a drone, causing 29 flights to be delayed, 8 canceled, and 32 landed, with more than 3000 travelers whose journeys were disrupted, and on the following night, the airport was again subjected to drone The following night the

airport was again disturbed by drones [2]. According to statistics, the Federal Aviation Administration (FAA) of the United States receives, on average, over 100 monthly reports of unauthorized drone flights [3].

The International Civil Aviation Organization (ICAO) and national civil aviation administrations have included drone intrusion identification in the category of key preventive security risks. Compared with large international hub airports, small and medium-sized remote airports are often in a more difficult situation when faced with drone black flight interference, mainly due to the remote location of these airports, the complex surrounding environment, the existing security protection facilities are relatively simple, the deployment of sophisticated monitoring systems and maintenance difficulties, resulting in a lack of professional security equipment, and the mobility of personnel, professional and technical personnel are limited, and can not be implemented around-the-clock manual surveillance, these factors together lead to the small and medium-sized remote airports, and the need to protect the airspace from the intrusion of drones. These factors together have led to small and medium-sized remote airports becoming the weak links of drone invasion. Therefore, there is an urgent need to integrate the actual operating environment and strengthen the rapid drone detection capabilities of small and medium-sized remote airports.

Currently, unmanned aerial vehicle (UAV) detection methods primarily include radar detection, acoustic detection [4], radio frequency detection [5], and optical detection [6]. In recent years, with the rapid development of deep learning technologies, optical detection methods based on machine vision have gradually become important tools for UAV identification and detection at small and medium-sized airports due to their advantages of flexible deployment, high intelligence level, relatively low cost, and immunity to electromagnetic interference. Although innovative approaches such as lightweight object detection network architectures [7], attention mechanisms [8], and multi-scale feature fusion techniques [9] have made significant progress in the field of object detection [10], the application of these technologies in rapid UAV identification and detection remains in its early stages and still struggles to simultaneously meet the dual requirements of detection accuracy and real-time performance needed by small and remote airports.

Based on the aforementioned analysis, this study proposes a machine vision-based approach for rapid UAV identification and detection, specifically designed to address the unique requirements and challenges faced by small and remote airports, and conducts validation tracking experiments to demonstrate its effectiveness. The main contributions of the paper are as follows:

(1) This study designs a dual-path fusion image input architecture for RGB visible light and infrared (IR) images, which strengthens information fusion and complementarity under multi-modal and multi-scale conditions to enhance the model's feature extraction capability for high-speed moving targets in complex backgrounds. A C3k2-DATB dynamic attention-enhanced feature extraction module is proposed, which integrates the grouped channel self-attention (G-CSA) and masked window self-attention (M-WSA) mechanisms to realize dynamic reconstruction of multi-dimensional information. This effectively captures the key features of highly maneuverable drones and solves the problem of difficult feature extraction in complex backgrounds.

(2) A bi-level attention routing mechanism (BRSA) based on proxy queries is designed, introducing the concept of "agent queries". By using deformable points as semantic-sensitive proxy queries, two-level routing attention allocation is achieved to accurately locate key regions and perform fine-grained feature sampling. This approach reduces computational complexity while enhancing the model's perception of small, high-speed moving drones.

(3) A semantic and detail injection (SDI) module is adopted to optimize the feature fusion process. Through adaptive feature mapping, interactive feature enhancement, and a gated fusion mechanism, the simple "feature concatenation" is upgraded to "meaningful interaction and selective injection", enabling mutual enhancement between high-level semantic information and low-level detail information. This effectively addresses the issues of multi-level information conflicts and feature dilution caused by traditional simple concatenation.

(4) A detection head (PCHead) improved by windmill-shaped convolution is introduced. By using asymmetric padding to create horizontal and vertical convolution kernels for different regions, this method significantly expands the model's receptive field with minimal parameter increase, enhancing the target detection capability under complex backgrounds and low signal-to-noise ratio conditions. Additionally, the Wasserstein Distance loss function based on optimal transport theory is introduced, which establishes a scale-insensitive and gradient-smooth loss metric to significantly improve the regression accuracy and convergence speed of bounding boxes.

(5) To validate the system's practicality, we propose an improved BoT-SORT algorithm based on the Interactive Multiple Model (IMM) framework, which integrates complementary motion models including constant velocity, constant acceleration, and turning models. Through inter-model probability interaction and weight fusion, the algorithm achieves adaptive tracking of complex UAV motion patterns, effectively resolving tracking drift issues for highly maneuverable targets. Combined with a hierarchical association strategy incorporating IoU and ReID features, the system achieves stable tracking under various lighting conditions.

2. Related Works

2.1. Main Technical Means of Drone Detection

UAV detection technology mainly includes radar detection, radio frequency detection, acoustic detection and optical detection, etc., of which radar detection is the most traditional means of airborne target detection, which is mainly categorized into two types: active radar and passive radar. Active radar detects targets by transmitting electromagnetic waves and receiving echoes, which has all-weather and long-range detection capability [11]. However, conventional radar systems face significant challenges in detecting small UAVs: first, the radar scattering cross-sectional area (RCS) of small UAVs is small, usually only 0.01–0.1 square meters, which is much lower than that of conventional aircraft; second, UAVs fly at low altitudes and slow speeds, and are easily masked by ground clutter [12]. Although millimeter wave radar performs better in small target detection, its coverage is limited and costly [13].

Radio frequency (RF) detection technology utilizes the communication signals between the UAV and the remote controller for identification and localization. Nguyen et al. [14] developed an RF detection system that can accurately identify common commercial UAVs and extract their model number characteristics. RF detection has the advantage of being simple to deploy, affordable, and able to work in line-of-sight-obstructed situations [15]. However, the technique also has obvious limitations, such as the inability to detect UAVs flying autonomously (not relying on remote control signals) and being susceptible to interference in complex electromagnetic environments [16].

Acoustic detection uses microphone arrays to capture sound features generated by UAVs for identification. The deep learning-based acoustic detection method proposed by Svanström et al. [17] achieves high identification rates in ideal environments. The acoustic detection system is flexible to deploy, has a low cost, and has no special requirements for UAV types. However, experiments by Hammer et al. [18] show that in noisy environments

(especially airport environments), the effective distance of acoustic detection is significantly reduced, and it is susceptible to wind and background noise, leading to high false alarm rates.

Optical detection techniques are based on images captured by visible or infrared cameras, combined with image processing and computer vision algorithms to recognize UAV targets. Wang et al. [19] showed that optical detection methods based on deep learning can achieve detection accuracy in open environments. Compared with other techniques, optical detection has the following advantages: first, it does not rely on signals emitted by drones and can detect all types of drones; second, it is flexible in deployment and can utilize existing surveillance facilities; and third, it can provide intuitive visual evidence. However, optical detection also faces challenges: it is highly affected by lighting conditions and weather conditions, and it is difficult to recognize small targets in complex backgrounds. Especially in small and medium-sized airport environments, optical detection systems need to overcome difficulties such as the small size of UAVs, complex backgrounds, and changes in lighting.

Combining the characteristics of each technology and the actual needs of small and medium-sized remote airports, machine vision-based optical detection technology has obvious economic and practical value [20]. By optimizing the algorithm design and solving the problems of small target recognition and environmental adaptability, optical detection can provide an economically feasible UAV monitoring solution for small and medium-sized airports.

2.2. Machine Vision Based Target Detection Method

Since the breakthrough success of AlexNet in the ImageNet competition in 2012, deep learning methods have rapidly made significant progress in various fields of computer vision, and target detection has entered a new stage of development as a result [21]. Unlike traditional methods that rely on hand-designed features, deep learning methods are able to automatically learn hierarchical feature representations and significantly improve detection performance [22]. Currently, deep learning target detection algorithms are mainly categorized into two main groups: two-stage detection algorithms and single-stage detection algorithms.

Two-stage detection algorithms first generate candidate regions, and then classify and positionally refine these regions. Two-stage detectors, particularly those in the R-CNN family (e.g., Faster R-CNN, Cascade R-CNN), have been evaluated for drone detection in several recent studies. For instance, Zhao. et al. [23] performed a broad benchmarking of two-stage detectors (Faster- RCNN, Cascade-RCNN with ResNet18/50 and VGG16 backbones) on the DUT Anti-UAV dataset, which comprises annotated images and videos of small UAVs in Performance was measured using metrics such as mean Average Precision (mAP), precision-recall curves, and inference Performance was measured using metrics such as mean Average Precision (mAP), precision-recall curves, and inference speed (frames per second, FPS). Cascade-RCNN with a ResNet50 backbone achieved the highest mAP among tested methods but at the cost of reduced inference speed compared to other approaches. Similarly, some scholars [24,25] included Faster-RCNN in comparative studies, assessing its efficacy in challenging settings, such as complex backgrounds and adverse weather. complex backgrounds and adverse weather.

The single-stage detection algorithm directly predicts the bounding box location and category, omits the region candidate generation step, and focuses on detection speed and real-time performance.

Papers [26,27] systematically compared YOLO-family models and SSD against two-stage methods, consistently finding that while single-stage detectors sometimes trail the

best two-stage detectors in mAP, they offer much higher FPS and are often more practical for real-time, edge-based detection. Notably, Hakani et al. [28] demonstrated the latest YOLOv9 model on the NVIDIA Jetson Nano, reporting a mAP of 95.7% with real-time detection speed, which represents a significant improvement over real-time detection speed. speed, which represented a significant improvement over the previous YOLOv8. Model optimization techniques such as pruning and transfer learning were also explored [25,29–31], further boosting the suitability of single-stage architectures for operational drone detection. For instance, Liu et al. [31] pruned YOLOv4 to yield a lightweight model capable of 69 FPS at 90.5% mAP, and several works employed training strategies and data augmentation schemes targeted at small, hard-to-detect drone detectors. schemes targeted at small, hard-to-detect drones [29–31].

However, deep learning methods still face challenges when applied at small and medium-sized remote airports. Zamri, et al. [32] pointed out that some untargeted optimized deep learning methods suffer from lack of accuracy when dealing with small-target UAVs at long distances. Gökçe et al. [33] found that the detection performance is seriously affected in complex backgrounds and poor weather conditions (e.g., cloudy and foggy days). In addition, Wang et al. [34] emphasized that the high computational volume of deep learning models tends to lead to difficulties in deploying them on edge devices with limited resources, which indirectly reflects the difficulty points in practical applications at remote airports.

Taken together, deep learning target detection methods have obvious advantages over traditional methods, but still need targeted optimization when dealing with the special needs of small and medium-sized remote airports. These optimization directions mainly include: enhancing the recognition ability of small targets, improving robustness in complex environments, and reducing model complexity to achieve edge deployment.

2.3. Multi-Object Tracking Algorithm

Target tracking is an important component of a visual surveillance system, whose core task is to continuously identify and localize the trajectory of the detected target's spatial position over time. In the UAV monitoring system of small and medium-sized remote airports, efficient and reliable target tracking algorithms can not only reduce the computational overhead and improve the real-time performance of the system, but also provide key data support for UAV behavior analysis and threat assessment.

Traditional target tracking algorithms rely on hand-crafted features and probabilistic motion models. Methods such as correlation filters (e.g., KCF, CSRT), background subtraction, and Kalman or particle filtering have been core components of earlier computer vision pipelines for object tracking. Pruned or lightweight implementations, including adaptation of single-object trackers like KCF with enhancements for UAV deployment, have enabled some level of real-time tracking on embedded hardware [35]. However, these approaches face substantial challenges when applied to UAV scenarios: the small size of targets, significant motion blur, drastic viewpoint changes, object occlusions, and frequently changing backgrounds all degrade performance. Furthermore, classical techniques struggle to sustain tracking through long-term occlusions, scale changes, or erratic motion and are often limited to single-object scenarios. These limitations have driven the field toward automated feature extraction and temporal data integration.

Deep learning has transformed target detection and tracking due to its capacity for robust, automated feature extraction and joint optimization for both detection and appearance representation. The dominant paradigms in recent years are end-to-end architectures unifying object detection and re-identification (ReID) embedding within a single network—commonly known as joint detection-embedding (JDE) or one-shot frame-

works [36,37]. These deep-learning models have demonstrated substantial improvements in accuracy and robustness under the adverse conditions typical for UAVs, such as small-object scale, dynamic backgrounds, and rapid viewpoint changes [38,39]. Temporal feature fusion modules [40,41], attention mechanisms [42], and explicit modeling of object and camera motion [43] further enhance association accuracy and continuity in challenging environments.

In comparison to traditional methods, deep learning-based approaches are more adaptable: they can learn to recognize objects under a wide array of visual conditions and can leverage multi-frame temporal context to anticipate occlusions or reappearances. Lightweight network architectures and model optimizations (e.g., YOLO-tiny, model pruning) have recently enabled real-time inference on embedded and resource-constrained hardware, expanding the viability of deep learning solutions for onboard UAV tracking [44,45].

Multi-target tracking (MOT) extends the challenge from following a single object to maintaining robust, persistent identities for multiple objects as they move, enter, or exit the field of view—often under severe occlusion and complex scene dynamics. Recent literature has shifted strongly toward tracking-by-detection frameworks that combine high-accuracy object detectors with advanced data association pipelines [46,47]. These systems typically use joint detection and ReID feature extraction to minimize identity switches and are now incorporating sophisticated modules to address UAV-specific problems, such as temporal feature boosting, cross-view geometric alignment [48], and explicit modeling of both object and platform motion. Performance is rigorously evaluated on UAV-specific benchmarks such as VisDrone and UAVDT, using metrics including MOTA, IDF1, and HOTA [49].

Comprehensively analyzing the current development trend of target tracking technology, the main challenges of UAV tracking for small and medium-sized remote airports focus on (1) how to reduce the computational complexity of the algorithms while maintaining high accuracy; (2) how to improve the robustness of the algorithms in complex backgrounds and bad weather conditions; and (3) how to effectively deal with tracking difficulties brought by high-speed movements and rapid attitude changes of UAVs.

3. Methods

Considering the diverse requirements for non-cooperative UAV detection, this paper proposes a rapid UAV identification and detection algorithm based on multi-modal data fusion, with the framework illustrated in the figure below. We present a fast small UAV detection algorithm based on an improved YOLOv11 network, enhancing the overall detection accuracy and efficiency through modifications to the feature extraction module and the introduction of attention mechanisms. To verify the system's practicality and account for UAV motion characteristics, we propose an improved BoT-SORT algorithm based on interactive multiple model filtering, enabling rapid identification and tracking of non-cooperative UAVs. Main structure is shown in Figure 1.

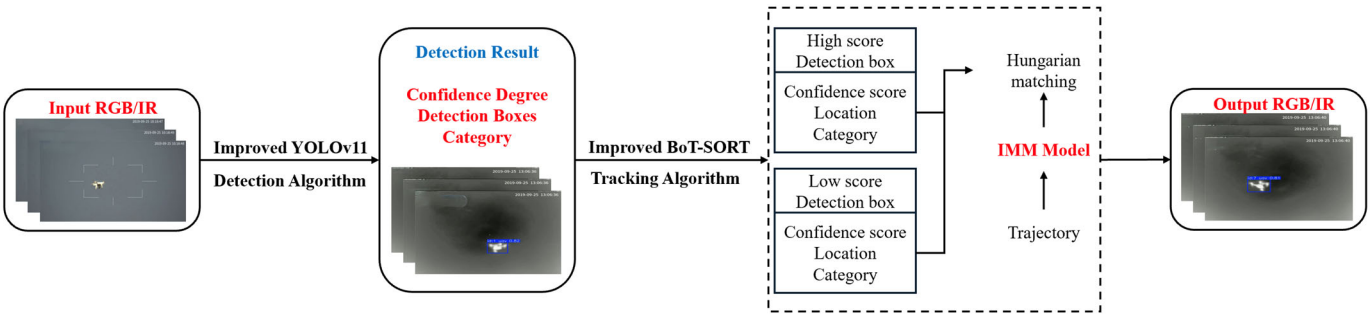


Figure 1. Framework of UAV Detection and Tracking Algorithm.

3.1. Target Detection Algorithms

YOLO (You Only Look Once), as a landmark algorithm in the field of image recognition, is based on its unique single-stage detection paradigm, which is able to simultaneously predict the location and category information of all targets in an image through a single network forward propagation, realizing an effective balance between detection accuracy and inference speed. YOLOv11 main structure is shown in Figure 2 [50].

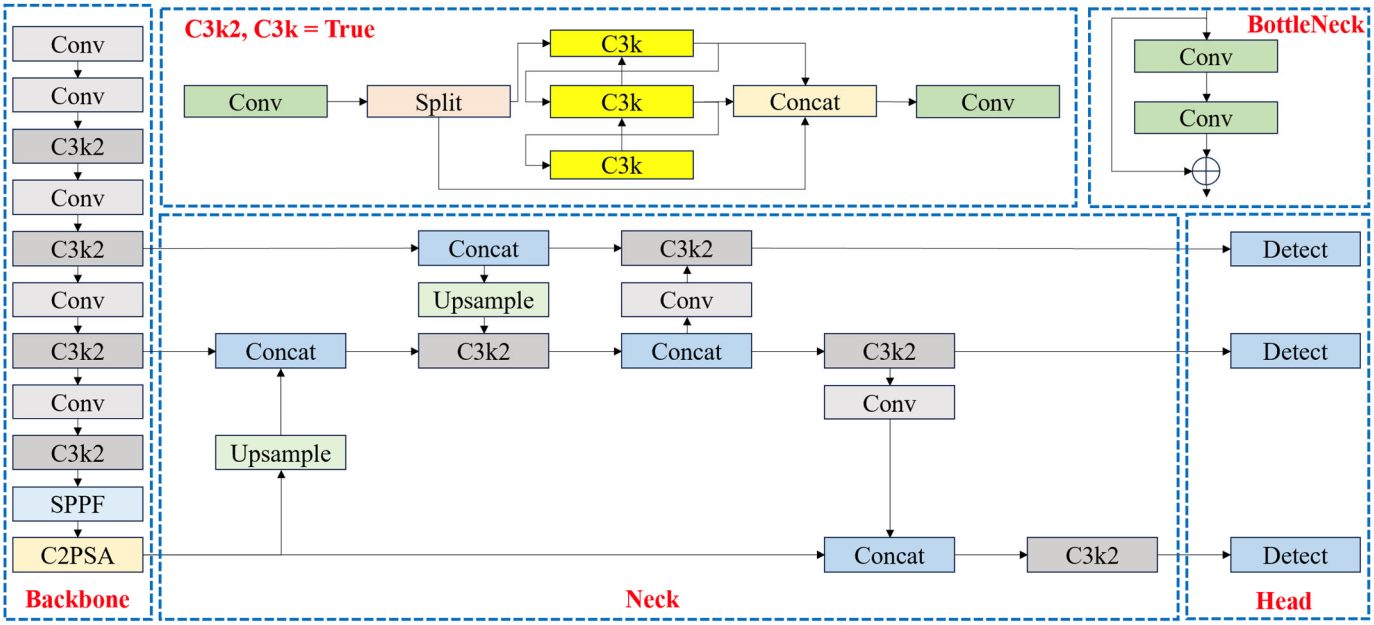


Figure 2. Structure of Original YOLOv11.

Although YOLOv11 shows excellent detection performance in the field of general-purpose target recognition, it still has certain technical bottlenecks in the face of the special requirements of the anti-UAV detection mission. Among them, at the feature expression level, the model is biased towards capturing global low-frequency information while ignoring the high-frequency details required for small UAV targets, and the lack of an effective cross-channel information interaction mechanism restricts the discriminative ability of the features. In the network structure design, its feature map resolution and down-sampling strategy lead to the gradual loss of key spatial detail information in the deep network, making it difficult to accurately localize tiny targets in the complex background. In feature fusion, the simple Concat operation not only increases the computational burden, but also leads to unbalanced mixing of semantic-spatial information and mutual interference between features of different depth layers, and fails to effectively integrate complementary information. Together, these limitations constrain the effectiveness of YOLOv11 in anti-UAV detection missions.

In view of this, this paper proposes a fast identification and detection algorithm for small UAV targets based on an improved YOLOv11 network, whose network structure is shown in Figure 3. The specific improvements are shown in Table 1 below:

Table 1. Key Improvements.

Module	Original Method	Improved Method
Fusion Architecture	Single-modal input with simple feature concatenation	Dual-path hierarchical fusion (RIFusion + ADD modules)

Table 1. *Cont.*

Module	Original Method	Improved Method
Feature Extraction	C3 fixed convolution kernels	Multi-scale deformable convolution ($3 \times 3/5 \times 5$ combinations)
Attention Mechanism	None	Agent queries with deformable point-based two-level routing
Feature Fusion	Direct concatenation	Spatial-adaptive gated bidirectional complementation
Detection Head	Conventional symmetric convolution	Pinwheel-shaped asymmetric convolution
Loss Function	Standard IoU Loss	Wasserstein Distance Loss with gradient guidance

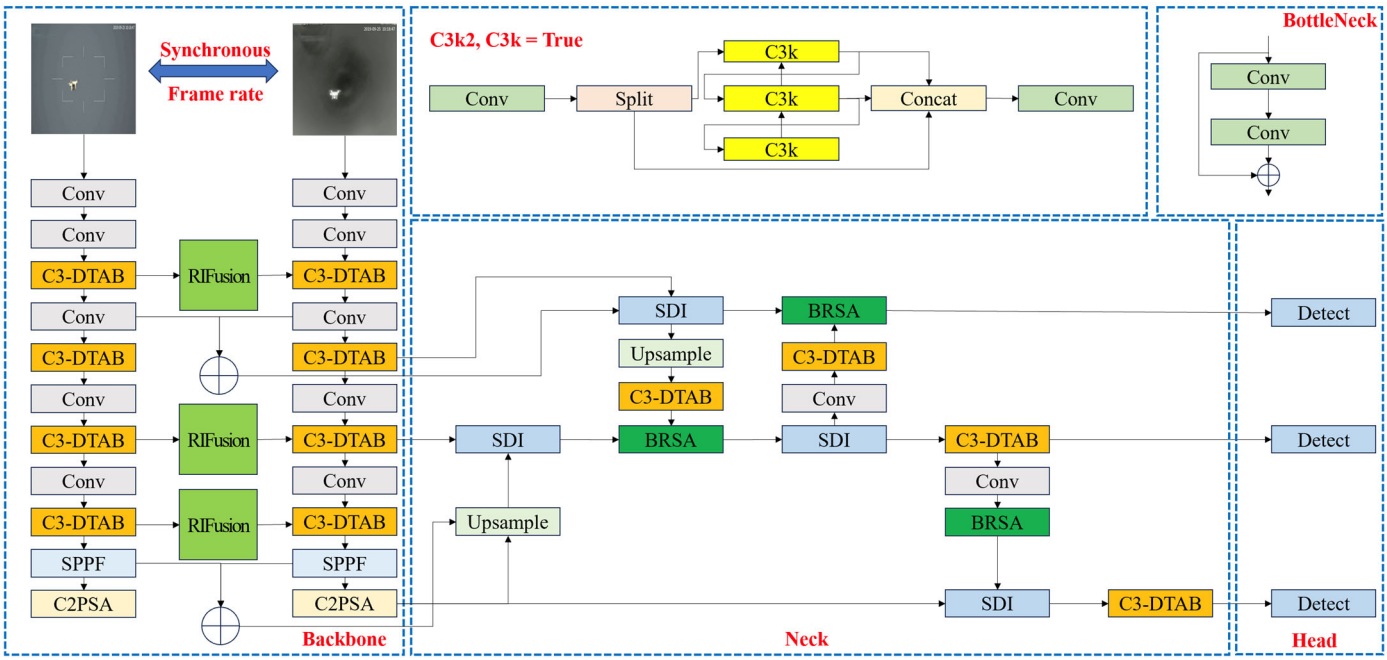


Figure 3. Structure of Improved YOLOv11.

3.1.1. C3k2-DTAB Module

The C3k2 module, as the core innovative component of YOLOv11, is a unique feature extraction mechanism formed by deeply improving the traditional C3 architecture and integrating variable convolution kernel and feature segmentation-splicing strategy. Its structure is as shown in Figure 4.

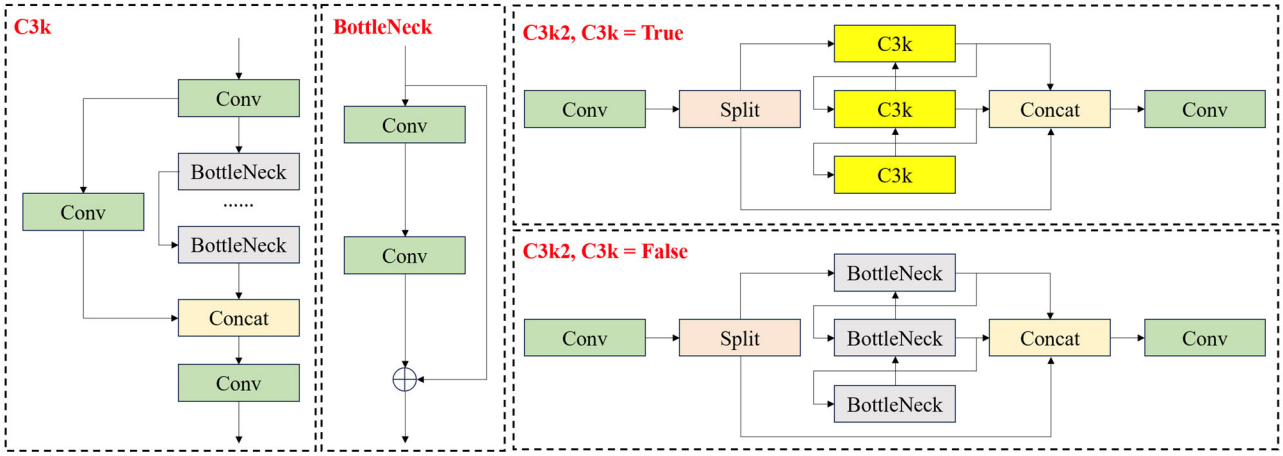


Figure 4. Structure of C3k2.

Although this module has excellent performance in many tasks, it still shows some deficiencies when facing the high-speed small UAV detection and tracking task scenarios:

1. The C3k2 module uses a gating mechanism to dynamically adjust the structure, and when the gating parameter is set to False, the module will be downgraded to the C2f structure. Although this adaptive mechanism can effectively balance computational resources and performance in simple scenes, when dealing with small target detection in complex backgrounds (e.g., UAV targets), the extracted low-frequency information is difficult to adequately characterize the local details of the target's features, which leads to the model's inability to efficiently construct the long-distance dependency relationship between the features, which directly affects the model's performance in complex visual environments.
2. In the design of the C3k2 module, there is the problem of insufficient interaction of channel information. The feature mappings of different channels are processed relatively independently, lacking an effective cross-channel information interaction mechanism. This design results in the module not being able to fully utilize the correlation information between channels, which limits the richness and accuracy of feature expression. Especially when dealing with targets with complex feature distributions, the mutual enhancement between the channel features cannot be fully activated, which reduces the discriminative ability of feature expression.
3. The information transfer between the C3k2 module and the subsequent target detection components is only through simple feed-forward connections, lacking in-depth feature fusion and interaction mechanisms. This loose integration of modules leads to the fact that the feature information extracted from the upstream cannot be fully utilized by the downstream detection module, which forms a "bottleneck" of information transfer, resulting in insufficient information fusion of the model and limiting the overall performance of the model.

Aiming at the inherent limitations of the C3k2 module in complex background high-speed UAV recognition and detection, a dual-attention-enhanced feature extraction framework based on improved DTAB (Dilated Transformer Attention Blocks) is introduced, which incorporates the Grouped Channel Self-attention, G-CSA, and Masked Window Self-Attention, M-WSA, mechanisms by grouping feature channels and establishing dynamic weights [51]. Its structure is as shown in Figure 5. Attention (G-CSA) and Masked Window Self-Attention (M-WSA), which effectively enhances the information interaction between channels by grouping the feature channels and establishing dynamic weighting relationships, realizing the adaptive enhancement of key features and the suppression of background noise. Meanwhile, drawing on the concept of inflationary convolution, more flexible long-range feature association is realized through attention calculation, which significantly expands the effective sensory field of the model and enables the network to pay attention to local details and global context at the same time. Through the synergistic effect of the two mechanisms in the channel dimension and spatial dimension, the model's ability in high-speed target detection, tiny feature capture and complex background interference suppression is jointly improved, thus realizing the fast recognition and detection of high-speed moving UAVs in complex scenes.

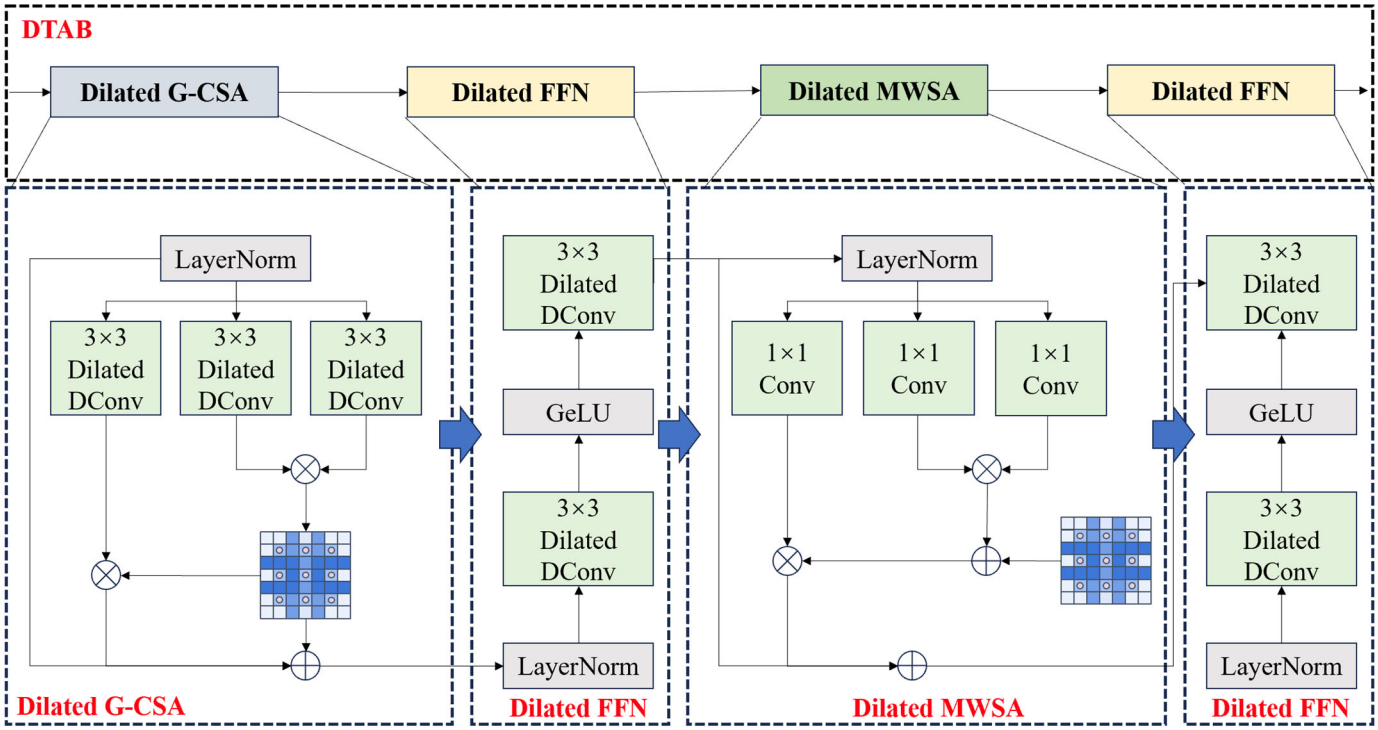


Figure 5. Structure of DTAB (Dilated Transformer Attention Blocks).

The specific computational flow of the improved model is shown below:

Suppose the input feature is $X \in \mathbb{R}^{H \times W \times C}$, where H, W, and C are the height, width, and number of channels of the feature map, respectively. The input features are first normalized by applying the LayerNorm layer to the input features, and the output features are denoted as X_{LN}. The normalization formula is shown below:

$$X_{LN} = \text{LayerNorm}(X) \tag{1}$$

The channel dimension is partitioned into G groups, each containing $C_g = \frac{C}{G}$ individual channels, and the grouped features are denoted as:

$$X = X_{g1}, X_{g2}, \ldots, X_{gG} \tag{2}$$

where, $X_{gk} \in \mathbb{R}^{H \times W \times C_g}$.

Next, global average pooling is performed on the features within each group to obtain global information for each channel. For group k, the global average pooling result is:

$$s_{gk} = \text{GlobalAvgPool}\left(X_{gk}\right) \in \mathbb{R}^{C_g} \tag{3}$$

The global average pooling results are nonlinearly transformed through the full link layer (FC) and the activation function ReLU to obtain the channel attention weights $\phi\left(X_{gk}\right)$:

$$\phi\left(X_{gk}\right) = \text{FC}\left(\text{ReLU}\left(\text{FC}\left(s_{g_k}\right)\right)\right) \in \mathbb{R}^{C_g} \tag{4}$$

The channel attention weights are normalized by applying the softmax function:

$$\phi_{\text{norm}}\left(X_{gk}\right) = \text{softmax}\left(\phi\left(X_{gk}\right)\right) \in \mathbb{R}^{C_g} \tag{5}$$

The recalibrated features are obtained by multiplying the channel attention weights with the features in the corresponding group on a channel-by-channel basis:

$$X_{gk}^{\text{out}} = X_{gk} \cdot \phi_{\text{norm}}\left(X_{gk}\right) \tag{6}$$

where, denotes element-by-element multiplication operation. The features of all groups are spliced to get the output features of G-CSA:

$$X_{\text{G-CSA}} = \text{Concat}\left(X_{g1}^{\text{out}}, X_{g2}^{\text{out}}, \ldots, X_{gG}^{\text{out}}\right) \tag{7}$$

When the feature map is input to the mask window self-attention mechanism (M-WSA), it is first divided into multiple windows of size 7×7. Assuming that the height and width of the feature map are $H = W = L$, the number of divided windows is $\frac{L^2}{M^2}$. Within each window, a linear transformation (e.g., a fully connected layer) is applied to the input features, and the input features are projected into the query (Q), key (K), and value (V) spaces to obtain the Q, K, and V matrices, respectively:

$$Q = W_Q \cdot X_{LN} \in \mathbb{R}^{M^2 \times d} \tag{8}$$

$$K = W_K \cdot X_{LN} \in \mathbb{R}^{M^2 \times d} \tag{9}$$

$$V = W_V \cdot X_{LN} \in \mathbb{R}^{M^2 \times d} \tag{10}$$

where, W_Q, W_K, W_V are the learnable weight matrix and d is the projected dimension.

Subsequently, a fixed-attention mask matrix $M \in \mathbb{R}^{M^2 \times M^2}$, whose elements are defined as:

$$M(i,j) = \begin{cases} 0 & \text{if } (i\%2 == 0 \text{ and } j\%2 == 0) \\ -\infty & \text{otherwise} \end{cases} \tag{11}$$

Incorporating the mask matrix into the attention fraction matrix:

$$S = QK^T/\sqrt{d} + M \tag{12}$$

The masked attention score matrix is normalized by applying softmax function to obtain the attention weight matrix:

$$W = \text{softmax}(S) \in \mathbb{R}^{M^2 \times M^2} \tag{13}$$

Finally, the G-CSA and M-WSA processed features are nonlinearly transformed through a feed-forward network (FFN) by two convolutional layers and an activation function (GeLU), respectively:

The first convolutional layer:

$$X_{\text{FFN1}} = \text{Conv}(\text{GeLU}(\text{Conv}(X_{\text{G-CSA}} + X_{\text{M-WSA}}))) \tag{14}$$

The second convolutional layer:

$$X_{\text{FFN2}} = \text{Conv}(X_{\text{FFN1}}) \tag{15}$$

The output features of the FFN are fused with the input features to obtain the final output features of the DTAB:

$$X_{\text{out}} = X_{\text{FFN2}} + X_{LN} \tag{16}$$

Through the above improvements, the limitations of insufficient inter-channel information interaction are overcome, and the accurate modeling of inter-feature channel dependencies is realized, which enhances the model's ability to distinguish between small targets and background. At the same time, long-distance feature associations are constructed to extend the range of sensory field, which solves the problem of the model's insufficiency in capturing global contextual information. The structure of C3k2-DTAB is shown in Figure 6.

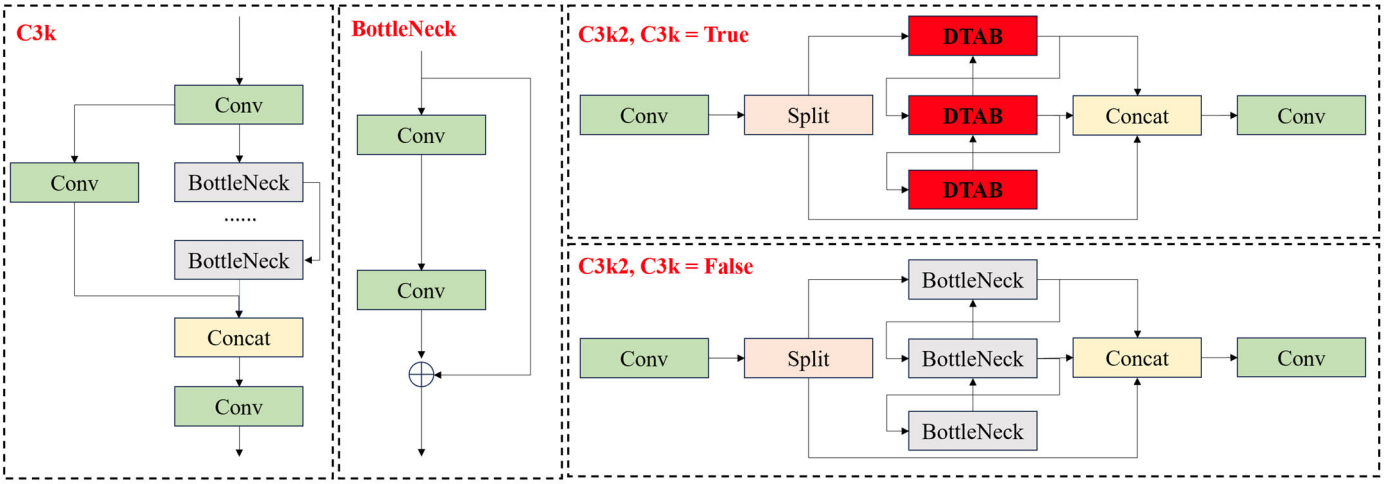


Figure 6. Structure of C3k2-DTAB.

3.1.2. Bi-Level Routing & Spatial Attention (BRSA)

Restricted by the inherent network structure and feature map resolution, the standard YOLOv11 network has certain limitations when facing small-sized and highly detailed targets (e.g., drones) in complex scenes. This is mainly reflected in the insufficient feature expression capability of the model, the single feature map is difficult to capture the rich detail information of small targets, and it tends to favor high-frequency categories in single-sample training scenarios, e.g., targets such as similar flocks of birds are incorrectly identified as drones, leading to classification bias problems. In addition, the model attention allocation mechanism lacks dynamic adaptability and cannot flexibly adjust the attention distribution according to the semantic content of the input image, which reduces the robustness and generalization ability of the model in complex environments.

To cope with the above challenges, some researchers have introduced the Convolutional Block Attention Module (CBAM) as a strategy to enhance the feature representation of the model. CBAM enhances the model's ability to perceive the critical region to a certain extent by combining the channel attention and the spatial attention. The structure of the CBAM attention mechanism is shown in Figure 7.

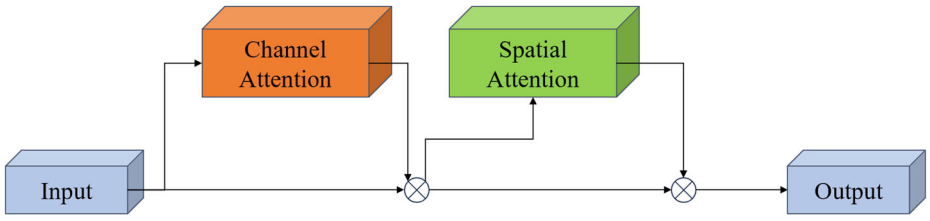


Figure 7. Structure of CBAM.

The calculation process is shown below:

In the channel dimension computation, the input feature map $F \in \mathbb{R}^{C\times H\times W}$ (where C, H, and W denote the channel, height, and width, respectively.) is firstly An average pooling operation is performed along the spatial dimension to obtain the average pooled features $F_{\text{avg}} \in \mathbb{R}^{C\times 1\times 1}$. The calculation formula is:

$$F_{\text{avg}}(c) = \frac{1}{H\times W}\sum_{h=1}^{H}\sum_{w=1}^{W} F(c,h,w),\ c \in [1,C] \tag{17}$$

Next, Favg and Fmax are each fed into a shared multilayer perceptual machine (MLP) for processing. The structure of an MLP typically consists of a hidden layer, where the number of neurons in the hidden layer can be set to (r is a scaling ratio used to minimize the number of parameters). The parameters of the MLP include the weight matrices $W_0 \in \mathbb{R}^{C/r\times C}$ and $W_1 \in \mathbb{R}^{C\times C/r}$. Subsequently, the input feature vectors $x \in \mathbb{R}^{C\times 1\times 1}$ is varied linearly through W_0, then the ReLU activation function is applied, and then linearly through W_1. The output of the MLP can be expressed as:

$$\text{MLP}(x) = \sigma(W_1 \cdot \text{ReLU}(W_0 \cdot x)) \tag{18}$$

where σ is the ReLU activation function.

The average pooled features and maximum pooled features after MLP processing are summed at the element level to obtain the channel attention features $M_{\text{c_raw}} \in \mathbb{R}^{C\times 1\times 1}$:

$$M_{\text{c_raw}} = \text{MLP}(F_{\text{avg}}) + \text{MLP}(F_{\text{max}}) \tag{19}$$

The channel attention map is obtained by mapping $M_{\text{c_raw}} \in \mathbb{R}^{C_{x|x|}}$ to the interval [0, 1] via the sigmoid activation function:

$$M_{\text{c}} = \sigma(M_{\text{c_raw}}) \tag{20}$$

where, $\sigma(x) = \frac{1}{1+e^{-x}}$.

Finally, the channel attention map Mc is multiplied element-by-element with the input feature map F to obtain the refined feature map in the channel dimension $F' \in \mathbb{R}^{C\times H\times W}$:

$$F'(c,h,w) = M_c(c)\cdot F(c,h,w),\ c \in [1,C], h \in [1,H], w \in [1,W] \tag{21}$$

In the spatial level computation, the feature map F' after the channel dimension refinement is firstly subjected to the average pooling operation along the channel axis to obtain the average pooled features $F_{\text{s_avg}} \in \mathbb{R}^{1\times H\times W}$. The calculation formula is:

$$F_{\text{s_avg}}(h,w) = \frac{1}{C}\sum_{c=1}^{C} F'(c,h,w),\ h \in [1,H], w \in [1,W] \tag{22}$$

Similarly, a maximum pooling operation is performed on F' along the channel axis to obtain the maximally pooled features $F_{s_\text{max}} \in \mathbb{R}^{1\times H\times W}$. The computational formula is:

$$F_{\text{s_max}}(h,w) = \max_{c\in[1,C]} F'(c,h,w),\ h \in [1,H], w \in [1,W] \tag{23}$$

Splice F_{s_avg} and $F_{\text{s_max}}$ in channel dimension to obtain the spliced feature $[F_{\text{s_avg}}; F_{\text{s_max}}] \in \mathbb{R}^{2\times H\times W}$. Apply a convolutional layer (usually using a 7×7 convolutional kernel) to the spliced feature to generate the spatial attention feature $M_{\text{s}_{\text{r}}\text{aw}} \in \mathbb{R}^{1\times H\times W}$. Assuming that the weight of the convolutional layer is Conv:

$$M_{\text{s_raw}} = \text{Conv}([F_{\text{s_avg};}; F_{\text{s_max}}]) \tag{24}$$

The spatial attention map $M_s \in \mathbb{R}^{1\times H\times W}$ is obtained by mapping $M_{s_{raw}} \in \mathbb{R}^{1\times H\times W}$ to the interval [0, 1] via the Sigmoid activation function:

$$M_s = \sigma(M_{s_raw}) \tag{25}$$

The spatial attention map $M_s \in \mathbb{R}^{1\times H\times W}$ is multiplied element-by-element with the channel dimension-refined feature map F' to obtain the final feature map $F'' \in \mathbb{R}^{C\times H\times W}$ refined in the spatial dimension:

$$F''(c,h,w) = M_s(h,w)\cdot F'(c,h,w),\ c\in[1,C], h\in[1,H], w\in[1,W] \tag{26}$$

The above computation can improve the model's ability to perceive key features to a certain extent, but there are still some defects. First, CBAM's channel attention mechanism overly relies on average pooling and maximum pooling operations for feature compression, and this simplified processing inevitably leads to the loss of semantic information, especially for complex semantic contents distributed in different spatial locations; second, CBAM globally pools the entire feature map when calculating the channel attention to generate a single attention vector to be applied to all spatial locations, ignoring the the semantic variability of different regions, and is unable to realize fine-grained attention allocation; third, the design of CBAM fails to fully consider the semantic correlation between feature points, making it difficult to effectively capture the small differences between the target and the background, which is crucial for distinguishing between visually similar but semantically different regions (e.g., drones vs. birds). These limitations together lead to the unsatisfactory performance of CBAM in small target detection tasks under complex backgrounds, especially when the target is small, the background is complex and similar interferences are present, and it fails to provide sufficient detection accuracy and robustness.

In view of this, this paper proposes a Bi-Level Routing & Spatial Attention mechanism, which adopts the CBAM attention mechanism as the baseline module, introduces the concept of "agent queries", and utilizes the deformable points as the semantically-aware query agents to adaptively localize to the key regions in the feature map, and learns the offset to select the region with the current target. The learned offset selects the key-value pairs that are most relevant to the current task, which ensures that the attention mechanism can accurately focus on the regions that are highly relevant to the target detection task and effectively avoids the defect of the lack of semantic associations of key-value pairs in the traditional sparse attention mechanism, whose structure is shown in Figure 8.

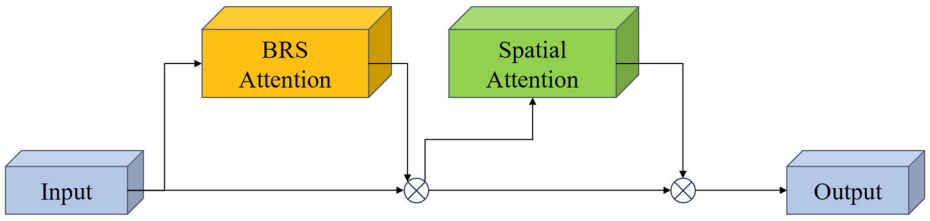


Figure 8. Structure of BRSA.

The BRA [52] optimizes the feature map representation mainly through two cascade routing processes, assuming an input feature map $F \in \mathbb{R}^{H\times W\times C}$, where H, W, and C denote the height, width, and number of channels, respectively. A uniformly distributed reference point $p \in \mathbb{R}^{H_G\times W_G\times 2}$ is generated by downsampling the input feature map, where $H_G = H/r$, $W_G = W/r$, r is the downsampling ratio. Its structure is shown in Figure 9.

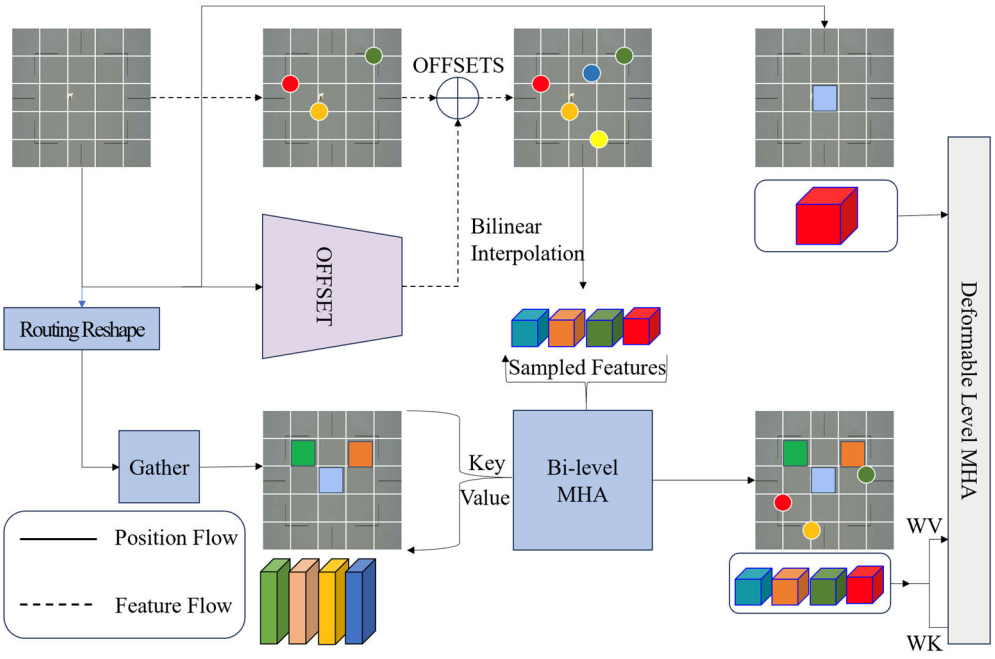


Figure 9. Structure of BRA.

Next, the input feature map is mapped to the feature query space to obtain the query features $q \in \mathbb{R}^{H_G \times W_G \times C}$:

$$q = F \cdot W_q \tag{27}$$

where, $W_q \in \mathbb{R}^{C \times C}$ is the projection matrix.

The query features are fed into the bias network θ_{offset} to generate offsets $\Delta p \in \mathbb{R}^{H_G \times W_G \times 2}$:

$$\Delta p = \theta_{\text{offset}}(q) \tag{28}$$

$$\overline{x}(i,j) = \sum_{rx,ry} g(p_x + \Delta p_x, rx) g(p_y + \Delta p_y, ry) x(rx, ry) \overline{x}_r \in \mathbb{R}^{S^2 \times \frac{H_G W_G}{S^2} \times C} \hat{v} = x_r W_v \hat{q}_r = \text{avg_pool}(\hat{q}) \tag{29}$$

A bias network is usually a small neural network that learns how to generate appropriate offsets based on query features.

The feature map is sampled based on the offsets to get the deformed features $\overline{x} \in \mathbb{R}^{H_G \times W_G \times C}$:

For each reference point $p(i,j)$ and its offset $\Delta p(i,j)$, compute the deformed coordinates $(p_x + \Delta p_x, p_y + \Delta p_y)$. Then find the four integer coordinate points (x_1, y_1), $(x_1 + 1, y_1)$, $(x_1, y_1 + 1)$, $(x_1 + 1, y_1 + 1)$ around the deformed coordinates and compute the bilinear interpolation weights for these four points:

$$g(a,b) = \max(0, 1 - |a - b|) \tag{30}$$

The value of the deformation feature is the weighted sum of these four points:

$$\overline{x}(i,j) = \sum_{rx,ry} g(p_x + \Delta p_x, rx) g(p_y + \Delta p_y, ry) x(rx, ry) \tag{31}$$

The deformed feature map and the original feature map are divided into non-overlapping regions of size S × S. The divided features can be expressed as:

$$\overline{x}_r \in \mathbb{R}^{S^2 \times \frac{H_G W_G}{S^2} \times C} \tag{32}$$

$$\overline{x}_r \in \mathbb{R}^{S^2 \times \frac{H_G W_G}{S^2} \times C} \tag{33}$$

Linear projection of region features to obtain region queries, keys and values:

$$\hat{q} = \overline{x}_r W_q \tag{34}$$

$$\hat{k} = x_r W_k \tag{35}$$

$$\hat{v} = x_r W_v \tag{36}$$

The features of each region are average pooled to obtain the region query and region key:

$$\hat{q}_r = \text{avg_pool}(\hat{q}) \tag{37}$$

$$\hat{k}_r = \text{avg_pool}(\hat{k}) \tag{38}$$

$$\text{avg_pool}(X) = \frac{1}{S^2} \sum_{i=1}^{S} \sum_{j=1}^{S} X(i,j) \tag{39}$$

The affinity matrix was constructed by matrix multiplication of region queries and region keys:

$$A_r = \hat{q}_r \hat{k}_r^{\top} \tag{40}$$

Affinity matrix $A_r \in \mathbb{R}^{\frac{H_G W_G}{s^2} \times \frac{H_G W_G}{s^2}}$ Indicates the similarity between different regions.

A pruning operation is performed on the affinity matrix, retaining only the first k connections in each region to obtain the routing index matrix:

$$I_r = \text{topk}(A_r) \tag{41}$$

The routing index matrix $I_r \in \mathbb{N}^{\frac{H_G G W_G}{s^2} \times k}$, denotes the indexes of the first k related areas for each area.

Based on the routing index matrix, the keys and values of the relevant areas are collected:

$$\hat{k}_g = \text{gather}\left(\hat{k}, I_r\right) \tag{42}$$

$$\hat{v}_g = \text{gather}(\hat{v}, I_r) \tag{43}$$

Attention calculations are performed on the collected keys and values:

$$\hat{O} = \hat{x} + W_o \left(\text{Attention}\left(\hat{q}, \hat{k}_g, \hat{v}_g\right) + \text{LCE}(\hat{v}) \right) \tag{44}$$

where $W_{o'} \in \mathbb{R}^{C \times C}$ is the output projection matrix and LCE denotes local context embedding with 5×5 depth convolution.

The feature X is then reshaped to $X_r \in \mathbb{R}^{H_G \times W_G \times C}$, and the reshaped feature is linearly projected to obtain the keys and values:

$$k = X_r W_k \tag{45}$$

$$v = X_r W_v \tag{46}$$

Finally, compute the relative position embedding $\phi(\hat{B}; R)$, where is the relative position paranoia and R is the relative position. And a second round of attention computation is performed using the query, key, value and relative position embedding:

$$z_m = W_{\bar{o}}\left(\sigma\left(\frac{q_m k(m)^\top}{\sqrt{d}} + \phi(\hat{B}; R)\right)v_m\right) \tag{47}$$

The outputs of all the attention heads are spliced and projected to get the final output:

$$z = \text{Concat}\left(z^{(1)}, \ldots, z^{(M)}\right)W_o \tag{48}$$

where $W_o \in \mathbb{R}^{C\times C}$ is the final output projection matrix and M is the number of attention heads.

3.1.3. Semantics Detail Fusion (SDI)

The standard YOLOv11 model mainly splices feature maps from different layers with the help of the Concat operation to fuse multi-scale feature information. When the spatial dimensions of feature maps are inconsistent, frequent up and down sampling transformations are required, which can easily increase the computational overhead and introduce information distortion. Moreover, the semantic information and spatial details carried by different levels of feature maps differ significantly, which can easily lead to the dilution of important features if undifferentiated splicing is performed. Most critically, the inconsistency in spatial localization between shallow detailed features and deep semantic features will form feature conflicts and interfere with the model decision-making process. To address the above problems, this study introduces the Semantics and Detail Infusion (SDI) [53] module to realize efficient feature fusion, which elevates simple feature splicing to meaningful interaction and selective injection through a three-phase processing mechanism of adaptive feature mapping, interactive feature enhancement, and gated fusion mechanism. The module firstly applies 1×1 convolution operation to the input high-level semantic features and low-level detailed features to map them to a common feature space, which unifies the feature representations and at the same time assigns preliminary weights to the features at different levels by means of learnable parameters, laying a foundation for the subsequent fusion. In addition, unlike simple feature splicing, the SDI module designs a bidirectional feature interaction mechanism. High-level semantic features inject category information and target structure knowledge into low-level features by means of attention guidance. At the same time, the low-level detail features supplement fine information such as texture and edges into the high-level feature representation through a similar mechanism. This two-way interaction ensures complementary feature fusion and effectively reduces information redundancy and feature conflicts. The core innovation of the SDI module is the introduction of a gating unit that controls the fusion ratio of semantic and detail information at each spatial location by learning a set of dynamic weights. This location-adaptive fusion strategy enables the model to prioritize the retention of semantic information in the target region while focusing on detail information in the boundary region, thus achieving optimal feature combination. Its structure is shown in Figure 10.

For the input image $I \in \mathbb{R}^{H\times W\times C}$, the encoder generates M levels of feature maps, denoted as $\{f_0^1, f_0^2, \ldots, f_0^M\}$. For the feature map f_0^i of the i-th level, the spatial and channel attention mechanisms are first applied to process it in order to fuse the local spatial information with the global channel information:

$$f_1^i = \phi_c^i\left(\phi_s^i\left(f_0^i\right)\right) \tag{49}$$

where ϕ_s^i and ϕ_c^i denote the spatial attention and channel attention functions at layer i, respectively. Here the spatial attention mechanism generates an attention map with the same spatial dimensions as the feature map, emphasizing important spatial regions, while the channel attention mechanism generates a channel weight vector, highlighting important feature channels.

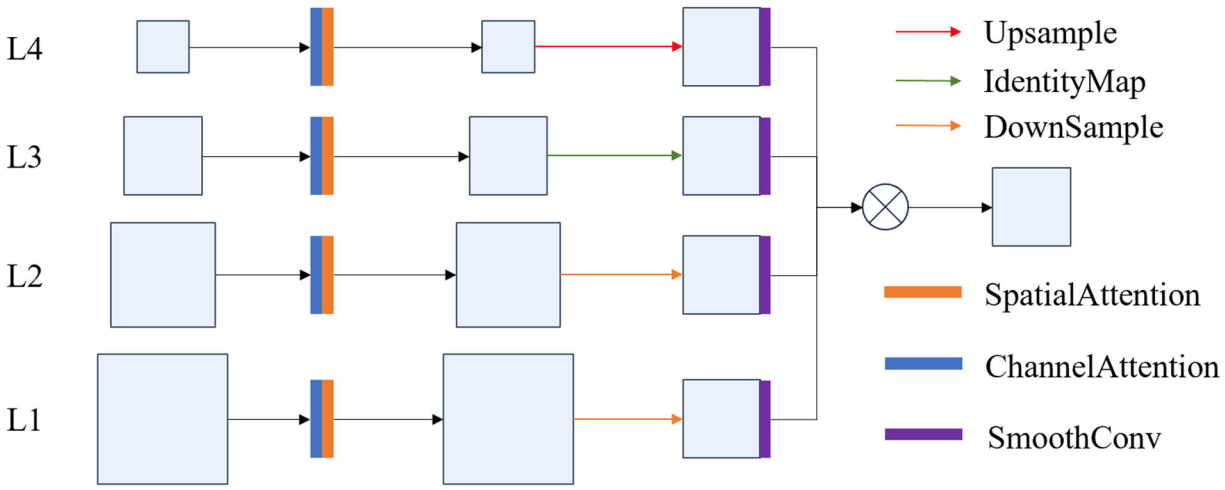


Figure 10. Structure of SDI.

Subsequently, the number of channels of the feature map is reduced by a 1×1 convolution operation:

$$f_2^i = \mathrm{Conv}_{1\times1}\left(f_1^i\right) \tag{50}$$

where, $f_2^i \in \mathbb{R}^{H_i \times W_i \times c}$, c is the number of target channels.

At layer i of the decoder, f_2^i is used as the target reference feature map. For each jth layer of the feature map f_2^j, its size is adjusted to match the resolution of f_2^i:

If $j < i$, i.e., the feature map comes from a shallower layer, use adaptive average pooling to reduce its size:

$$f_3^{i,j} = \mathrm{AdaptiveAvgPool}(\mathrm{f_2^j}, (\mathrm{H_i}, \mathrm{W_i})) \tag{51}$$

If $j = i$, then the feature map dimensions are the same and no adjustment is required:

$$f_3^{i,j} = f_2^j \tag{52}$$

If $j > i$, i.e., the feature map comes from a deeper layer, use bilinear interpolation to enlarge its size:

$$f_3^{i,j} = BilinearInterpolate\left(f_2^j, (H_i, W_i)\right) \tag{53}$$

The resized feature maps $f_3^{i,j}$ are smoothed by applying 3×3 convolution to reduce noise and enhance feature coherence:

$$f_4^{i,j} = \theta_{i,j}\left(f_3^{i,j}\right) \tag{54}$$

where $\theta_{i,j}$ denotes the parameters of the smoothing convolution.

All resized and smoothed feature maps are fused by Hadamard product (element-by-element multiplication) to enhance the feature maps of layer i. The feature maps of layer i are fused by Hadamard product (element-by-element multiplication):

$$f_5^i = H\left(\left[f_4^{i,1}, f_4^{i,2}, \ldots, f_4^{i,M}\right]\right) \tag{55}$$

where $H(\cdot)$ denotes the Hadamard product operation.

Through the above operations, we can unify different levels of features into a common feature space, realize the mutual enhancement of high-level semantic information and low-level detail information, and dynamically regulate the feature fusion ratio at different spatial locations, reduce the computational cost brought by frequent up and down sampling, avoid the dilution of important features caused by non-differentiated splicing, as well as eliminate the conflicting interference between shallow and deep features, and then solve the standard YOLOv11 network multi-scale feature fusion problem.

3.1.4. PCHead

The standard convolution operation used in target detection algorithms such as the standard YOLOv11 is prone to ignore the spatial features of the distribution of infrared small target image elements. Its structure is as shown in Figure 11. Therefore, this paper proposes an improved detection head based on windmill-shaped convolution, which creates horizontal and vertical convolution kernels in different regions through asymmetric filling, effectively enhances the underlying feature extraction capability, and realizes the enhancement of the model's feature extraction capability and significantly expands the model's sensory field under the premise of introducing only a very small parameter increase. The structure of PConv [54] is shown in Figure 12 and Improved head is shown in Figure 13.

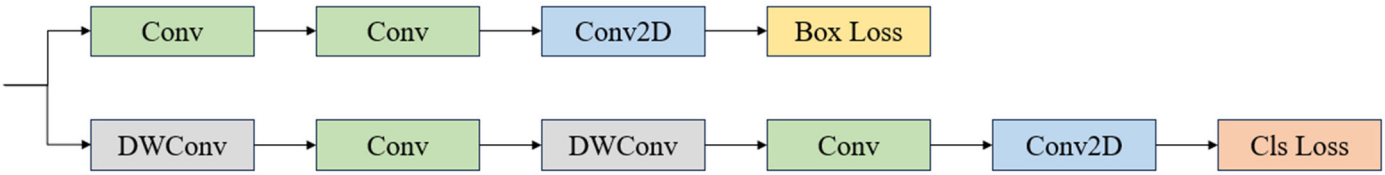


Figure 11. Structure of Original Detect Head.

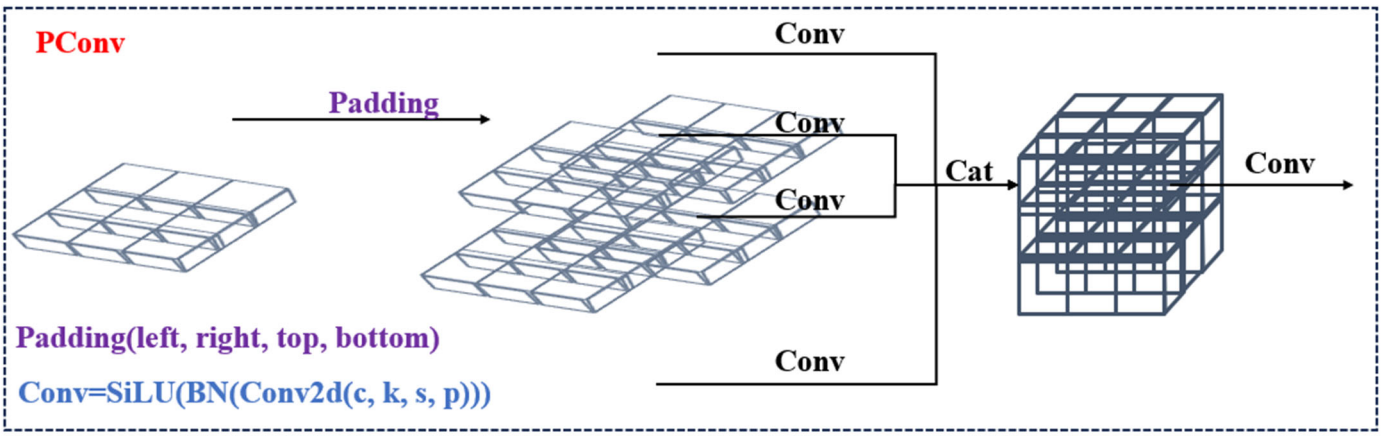


Figure 12. Structure of PConv.

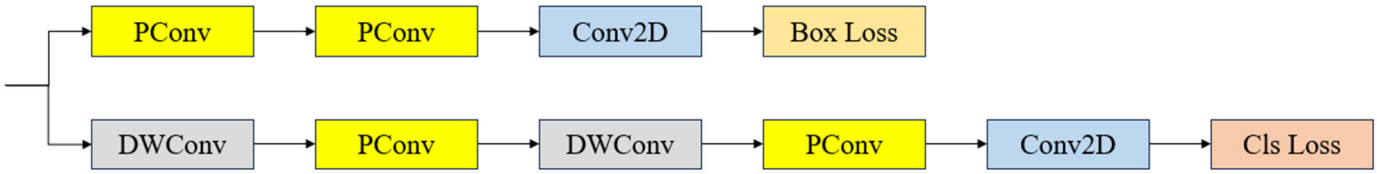


Figure 13. Structure of PCHead.

Its main computational process is shown below:

Suppose the input tensor is $T(h_1, w_1, c_1)$, where h_1, w_1 and c_1 denote the height, width and number of channels of the input tensor, respectively. The first layer of PConv performs a parallel convolution operation, which is computed as follows:

$$T_1(h', w', c') = \text{SiLU}(\text{BN}(T(h_1, w_1, c_1) \otimes W_1(1, 3, c'))) \tag{56}$$

$$T_2(h', w', c') = \text{SiLU}(\text{BN}(T(h_1, w_1, c_1) \otimes W_2(3, 1, c'))) \tag{57}$$

$$T_3(h', w', c') = \text{SiLU}(\text{BN}(T(h_1, w_1, c_1) \otimes W_3(1, 3, c'))) \tag{58}$$

$$T_4(h', w', c') = \text{SiLU}(\text{BN}(T(h_1, w_1, c_1) \otimes W_4(3, 1, c'))) \tag{59}$$

where $\otimes$ denotes the convolution operation, $W_1(1, 3, c')$ denotes a 1×3 convolution kernel, and the output channel is c', and $P(1, 0, 0, 3)$ denotes the fill parameter, which denotes the number of filled pixels on the left, right, top, and bottom, respectively.

After the first layer of convolution, the height, width and number of channels of the output feature map are related to the input feature map as follows:

$$h' = \frac{h_1}{s} + 1 \tag{60}$$

$$w' = \frac{w_1}{s} + 1 \tag{61}$$

$$c' = \frac{c_2}{4} \tag{62}$$

where c_2 is the number of channels of the final output feature map of the PConv module and s is the step size of the convolution.

The results of the first convolutional layer are spliced (Cat) to obtain:

$$X'(h', w', 4c') = \text{Cat}(X_1(h', w', c'), \ldots, X_4(h', w', c')) \tag{63}$$

The spliced tensor is normalized by a 2×2 convolution kernel $W(2, 2, c_2)$, and the height and width of the output feature map are adjusted to the preset values h_2 and w_2:

$$h_2 = h' - 1 = \frac{h_1}{s} \tag{64}$$

$$w_2 = w' - 1 = \frac{w_1}{s} \tag{65}$$

Final Output $Y(h_2, w_2, c_2)$:

$$Y(h_2, w_2, c_2) = \text{SiLU}(\text{BN}(X'(h', w', 4c') \otimes W(2, 2, c_2))) \tag{66}$$

For standard convolution, the number of parameters is calculated as:

$$ConvParams = c_2 \times c_1 \times k \tag{67}$$

where k is the size of the convolution kernel.

For PConv, the parameter quantity is calculated as follows:

When the number of output channels c_2 is equal to the number of input channels c_1, the parametric quantity of a standard convolution (e.g., 3×3 convolution) is 9c12. and the parametric quantity for PConv is calculated as:

$$PConvParams = 4 \times \left(\frac{c_2}{4} \times c_1 \times 3 \times 1\right) + 4c_2c_1 = 7c_2c_1 = 7c_1^2 \tag{68}$$

This shows that PConv significantly expands the receptive field with a small number of added parameters.

In summary, PCHead creates horizontal and vertical convolutional kernels with different regions by asymmetric padding, which enables the model to capture the features of small IR targets more efficiently in the presence of complex backgrounds and low signal-to-noise ratios. And a larger receptive field can be acquired at a shallower level of the network, thus capturing the global information of small targets more efficiently. While maintaining the performance, only a very small amount of parameter increase is introduced, which improves the model's ability to detect small targets without significantly increasing the model's complexity.

3.1.5. Wasserstein Distance Loss

The original loss functions are variants based on IoU (Intersection over Union), which calculates the overlap between the predicted and real frames to guide model learning during training. However, when these loss functions are used in combination with the anchor detector, since the IoU class of metrics is extremely sensitive to small target localization deviation, small target pixels are sparse, and subtle localization errors can lead to a sharp drop in IoU, which affects the quality of the allocation of positive and negative samples, making it difficult for the network to converge, and can easily lead to insufficient number of positive samples, weakening the model's ability to detect small targets. In addition, when there is no overlap or only partial overlap between the prediction frame and the real frame, the IoU loss cannot provide an effective gradient, which interrupts the model optimization process and has a significant impact on small targets in particular, and the traditional IoU loss does not fully consider the change of target scale, which makes it difficult to balance the detection performance among targets of different scales, and affects the model generalization ability.

To solve the above problems, this paper introduces the Wasserstein Distance Loss (WD-Loss) loss function [54]. This function is based on the optimal transmission theory, the target bounding box is regarded as a probability distribution on the two-dimensional plane, and the prediction quality is evaluated by calculating the minimum "moving cost" between the distributions. The specific calculation is shown below:

For the predicted bounding box P and the real bounding box G, extract their center coordinates, widths and heights, respectively, assuming that the center coordinates of the predicted bounding box P are $\left(c_x^p, c_y^p\right)$, the width is w_p, and the height is h_p. The center coordinate of the real bounding box G is $\left(c_x^g, c_y^g\right)$, the width is w_g and the height is h_g. The predicted bounding box P and the real bounding box G are modeled as two-dimensional Gaussian manifolds $\mathcal{N}_P$ and $\mathcal{N}_G$, respectively, and their probability density functions are:

$$f_P\left(x\middle|\mu_P, \sum_P\right) = \frac{\exp\left(-\frac{1}{2}(x-\mu_P)^T \sum_P^{-1}(x-\mu_P)\right)}{2\pi|\sum_P|^{1/2}} \tag{69}$$

$$f_G\left(x\middle|\mu_G,\sum_G\right)=\frac{\exp\left(-\frac{1}{2}(x-\mu_G)^T\sum_b^{-1}(x-\mu_G)\right)}{2\pi|\sum_G|^{1/2}} \tag{70}$$

The mean vector and covariance of the two Gaussian distributions are:

$$\mu_P=\begin{bmatrix} c_x^p \\ c_y^p \end{bmatrix} \tag{71}$$

$$\sum_P=\begin{bmatrix} \frac{w_p^2}{4} & 0 \\ 0 & \frac{h_p^2}{4} \end{bmatrix} \tag{72}$$

$$\mu_G=\begin{bmatrix} c_x^g \\ c_y^g \end{bmatrix} \tag{73}$$

$$\sum_G=\begin{bmatrix} \frac{w_g^2}{4} & 0 \\ 0 & \frac{h_g^2}{4} \end{bmatrix} \tag{74}$$

Next, the second-order Wasserstein distance W_2 between two Gaussian distributions $\mathcal{N}_P$ and $\mathcal{N}_G$ is computed:

$$W_2(\mathcal{N}_P,\mathcal{N}_G)=\|\mu_P-\mu_G\|_2^2+\mathrm{Tr}\left(\sum_P+\sum_G-2\left(\sum_P^{1/2}\sum_G\sum_P^{1/2}\right)^{1/2}\right) \tag{75}$$

where $\|\cdot\|_2$ denotes the L2 norm, $\mathrm{Tr}(\cdot)$ denotes the trace of the matrix, and $\sum_P^{1/2}$ denotes the square root of the matrix $\sum_P$.

To normalize the Wasserstein distance to the interval [0, 1], the normalized Wasserstein distance (NWD) is defined as:

$$\mathrm{NWD}(\mathcal{N}_P,\mathcal{N}_G)=\exp\left(-\frac{W_2^2(\mathcal{N}_P,\mathcal{N}_G)}{C}\right) \tag{76}$$

where C is a constant, usually set empirically based on the average absolute size of the dataset.

Ultimately, the Wasserstein Distance Loss is defined as:

$$L_{WD}=1-\mathrm{NWD}(\mathcal{N}_P,\mathcal{N}_G) \tag{77}$$

As in:

$$L_{WD}=1-\exp\left(-\frac{W_2^2(\mathcal{N}_P,\mathcal{N}_G)}{C}\right) \tag{78}$$

Through the above improvements, the model's response to the positional deviation of targets at different scales is made more balanced, and the response to the positional deviation of targets at different scales is made more balanced, which ensures the stability of small-scale target learning. In addition, meaningful gradient information is provided even when the prediction frame is completely separated from the real frame, accelerating convergence and enhancing the ability to detect complex scenes, and balancing the training contribution of multi-scale targets through an adaptive mechanism to improve the detection accuracy of the model, especially for small-scale targets.

3.2. The Proposed Method of UAV Tracking

Small and remote airports typically utilize stitched surveillance footage with limited field of view, necessitating rapid tracking when highly maneuverable UAVs exit the moni-

tored image boundaries to confirm their departure from airport protection zones. To further validate the system's practicality, this paper proposes an improved BoT-SORT algorithm based on the Interacting Multiple Model (IMM) framework, which predicts the target state by fusing multiple complementary motion models (e.g., uniform velocity motion model, uniform acceleration motion model, turning motion model, etc.), calculates the interaction probability of the target transferring from one model state to another, and will calculate the target transferring from one model state to another. model state, calculating the interaction probability of the target transferring from one model state to another, and weighting and fusing the prediction results of each model to obtain a comprehensive prediction state, and then updating this comprehensive prediction state with observation data, so as to realize adaptive tracking of complex UAV motion patterns. The specific calculation process is as follows:

First, the current frame is predicted using a target detector (Improved-YOLOv11 in this paper) to obtain a set of detection frames $D_k = \{d_1, d_2, \ldots, d_n\}$, where each detection frame d_i includes the location information of the bounding box and a confidence score.

Assuming that a multi-UAV system with motion model uncertainty evolves into a Jump Markov Linear System (JMLS), the stochastic equations of state and measurement equations for each model are then defined as:

$$x_{k+1} = f_j(x_k, u_k) + w_{j,k} \tag{79}$$

$$z_k = h_j(x_k, u_k) + v_{j,k} \tag{80}$$

where: $j = 1, 2 \ldots r$ is a part of the model set $M = \left\{M_j\right\}_{j=1}^{r}$, $w(j,k)$ is a Gaussian white noise with mean 0 and covariance matrix $v_{j,k}$.

The posterior probability p of the IMM is derived based on the Bayesian framework as $p(x_k, M_k|\, z_k)$, the measurements Z_k, the discrete state variables x_k, and the motion patterns M_k, which can be further decomposed using conditional probability elicitation as:

$$p(\mathbf{x}_k, M_k|\, \mathbf{z}_k) = p(\mathbf{x}_k, M_k|\, \mathbf{z}_k) p(M_k|\, \mathbf{z}_k) \tag{81}$$

$$p(\mathbf{x}_k, M_k|\, \mathbf{z}_k) = \sum_{j=1} p\left(\mathbf{x}_k, M_{j,k} \middle|\, \mathbf{z}_k\right) \underbrace{p\left(M_{j,k} \middle|\, \mathbf{z}_k\right)}_{\mu_{j,k}} \tag{82}$$

where: $\mu_{j,k}$ is the posterior probability of j motion models at moment k. It is further introduced:

$$p(\mathbf{x}_{k-1}, M_k|\, \mathbf{z}_{k-1}) = \sum_{j=1}^{r} p\left(\mathbf{x}_{k-1}, M_{j,k} \middle| \mathbf{z}_{k-1}\right) \mu_{i|j,k-1} \tag{83}$$

$$\mu_{i|j,k-1} = \frac{p_{ij}\mu_{i,k-1}}{\underbrace{\sum_{i=1}^{r} p_{ij}\mu_{i,k-1}}_{\mu_{j,k}}} \tag{84}$$

$\mu_{j,k}$ is the model matching probability of filter j at moment k. The model transfer matrix can be expressed as:

$$\begin{bmatrix} p_{11} & \cdots & p_{1r} \\ \vdots & \ddots & \vdots \\ p_{r1} & \cdots & p_{rr} \end{bmatrix} \tag{85}$$

where p_{ij} denotes the transfer matrix from model i to model j.

The IMM combines the weighted average of each j-model filter state $\hat{x}_{j,k-1}$ to determine the individual combined estimated state $\hat{x}^*_{j,k-1}$ and its corresponding variance $\hat{P}^*_{j,k-1}$:

$$x^*_{j,k-1} = \sum_{i=1}^{r} \mu_{(i|j),k-1}\hat{x}_{i,k-1}.\ i = 1,2,3,\ldots,r \tag{86}$$

$$P^*_{j,k-1} = \sum_{i=1}^{r} \mu_{(i|j),k-1}\hat{x}_{i,k-1}\left[P_{i,k-1} + \left(\hat{x}_{j,k-1} - \hat{x}^*_{j,k-1}\left(\hat{x}_{j,k-1} + \hat{x}^*_{j,k-1}\right)^T\right]\right. \tag{87}$$

Prediction of mixed initial states $\mathbf{x}^*_{j,k-1}$ and $\square P^*{}_{j,k-1}$ using Kalman filtering:

$$\mathbf{x}^-_{j,k} = F_{j,k}\mathbf{x}^*_{j,k-1} \tag{88}$$

$$\mathbf{P}^{j^-}_{k/k-1} = F_{j,k}\mathbf{P}^*_{j,k-1}F^T_{j,k} + \mathbf{Q}_{j,k} \tag{89}$$

where $Q_{j,k}$ denotes the covariance of the external interference noise and is a non-negative fixed array.

In the IMM algorithm, the model update directly affects the effectiveness of the algorithm, which uses the great likelihood function method to realize the model update: by calculating the similarity of the current motion target state to give the weight occupied by the current most suitable tracking model:

$$\mu_{j,k} = \frac{\lambda_{ij}\mu^-_{j,k}}{\sum\limits_{i=1}^{r}\lambda_{ij}\mu^-_{i,k}} \tag{90}$$

$$\lambda_{j,k} = \frac{1}{\sqrt{\left|2\pi\mathbf{S}_{j,k}\right|}}e^{0.5(\mathbf{z}_k - \hat{\mathbf{z}}^-_{j,k}1_j)^T\mathbf{S}^{-1}_{j,k}(\mathbf{z}_k - \hat{\mathbf{z}}^-_{j,k})} \tag{91}$$

where: $\mu_{j,k}$ is the probability that model i is updated, reflecting how well the current measurement fits model j; $\lambda_{j,k}$ is the Gaussian likelihood estimate of the measurement; and $R_{j,k}$ is the output of each j-filter, which is recombined into individual states $\hat{x}_k$ and $\hat{P}_k$:

$$\hat{x}_k = \sum_{i=1}^{r}\mu_{j,k}\hat{\mathbf{x}}_{j,k} \tag{92}$$

$$P_k = \sum_{i=1}^{r}\left(P_{j,k} + \left(\hat{x}_{j,k} - \hat{x}\right)\left(\hat{x}_{j,k} - \hat{x}_k\right)^T\right) \tag{93}$$

Meanwhile, in order to compensate for the effects brought by the dynamic camera, the motion parameters of the camera are estimated by image alignment technique and the prediction results of Kalman filter are corrected. First, the image keypoints of the current frame and the previous frame are extracted for feature tracking, sparse optical flow is calculated, and the affine transformation matrix is solved using the RANSAC algorithm $A_{k,k-1} \in \mathbb{R}^{2\times3}$:

$$A_{k,k-1} = \left[\begin{array}{cc} M_{2\times2} & T_{2\times1} \end{array}\right] = \left[\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{array}\right] \tag{94}$$

where M contains the scale and translation parts and T contains the translation part.

Next, the Kalman filter state is corrected by defining $\widetilde{M}_{k,k-1} \in \mathbb{R}^{8\times 8}$ and $\widetilde{T}_{k,k-1} \in \mathbb{R}^{8}$ as follows:

$$\widetilde{M}_{k,k-1} = \begin{bmatrix} M & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & M & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & M & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & M & 0 & 0 & 0 & 0 \end{bmatrix} \tag{95}$$

$$\widetilde{T}_{k,k-1} = \begin{bmatrix} a_{13} & a_{23} & 0 & 0 & \dots & 0 \end{bmatrix}^T \tag{96}$$

The corrected predicted state vector is:

$$\hat{x}'_{k|k-1} = \widetilde{M}_{k,k-1}\hat{x}_{k|k-1} + \widetilde{T}_{k,k-1} \tag{97}$$

Corrected forecast covariance matrix:

$$P'_{k|k-1} = \widetilde{M}_{k,k-1}P_{k|k-1}\widetilde{M}^T_{k,k-1} \tag{98}$$

In addition, the similarity matrix between the detection frame and the tracking target is computed by combining the IoU and Re-ID features for association matching. For the detection frame d_i, its feature vector f_i is:

$$f_i = f_{\text{ReID}}\left(\text{crop}(d_i, I_k)\right) \tag{99}$$

where I_k denotes the current frame image and $\text{crop}(d_i, I_k)$ denotes the region corresponding to the detection frame I_k cropped from the image d_i.

For each tracking target t_i and detection frame d_j, calculate the intersection area $\text{Intersection}(t_i, d_j)$ and concatenation area $\text{Union}(t_i, d_j)$ of their bounding boxes with IoU values:

$$\text{IoU}(t_i, d_j) = \frac{\text{Intersection}(t_i, d_j)}{\text{Union}(t_i, d_j)} \tag{100}$$

IoU distance is defined as:

$$C_{\text{Iov},\hat{i},j} = 1 - \text{IoU}(t_i, d_j) \tag{101}$$

For each tracking target t_i's appearance feature e_i and detection frame d_j's feature vector f_j, calculate their cosine similarity:

$$\text{CosSim}(e_i, f_j) = \frac{e_i \cdot f_j}{\|e_i\|\|f_j\|} \tag{102}$$

The cosine distance is defined as:

$$C_{\cos,i,j} = 1 - \text{CosSim}(e_i, f_j) \tag{103}$$

Next, the cosine distance is thresholded to filter out candidate matches with low similarity:

$$\hat{d}_{\cos,i,j} = \begin{cases} 0.5 \cdot C_{\cos,i,j}, & \text{if } C_{\cos,i,j} < \theta_{\text{emb}} \;\&\; C_{\text{IoU},i,j} < \theta_{\text{IoU}} \\ 1, & \text{otherwise} \end{cases} \tag{104}$$

where θ_{emb} and θ_{IOU} are the thresholds for ReID and IoU, respectively.

Then, the minimum of the IoU distance and the processed cosine distance is taken as the final similarity metric:

$$C_{i,j} = \min\left\{C_{\text{Iov},i,j}, \hat{d}_{\cos,i,j}\right\} \tag{105}$$

Finally, the Hungarian algorithm is used to match the detection frame and tracking target based on the similarity matrix to minimize the total cost.

The method uses a flexible multi-model fusion mechanism to replace a single constant velocity motion model, which in turn improves the adaptability of the algorithm to complex motion patterns and realizes real-time tracking of highly maneuverable non-cooperative UAVs in complex backgrounds.

4. Experiment

4.1. Datasets

In this study, we conduct experiments using the publicly available Anti-UAV dataset that contains both static images and video sequences (20 FPS), which is rigorously partitioned into training and validation sets at a 7:3 ratio. Through frame extraction from videos and geometric transformations including random cropping, rotation, and color adjustment, we expand the dataset to 25,000 images. To enhance training stability, we implement phased data augmentation: Mosaic augmentation is activated for the initial 290 epochs and deactivated during the final 10 epochs. The enhanced dataset covers diverse environments (urban, rural, mountainous) with synchronized infrared-visible image pairs, where infrared captures thermal signatures and visible retains texture details. This multimodal configuration enables complementary fusion detection that significantly improves UAV recognition accuracy. The combined strategy of balanced data partitioning, scene diversity, and phased augmentation ensures robust generalization across real-world scenarios, as demonstrated in Figure 14.

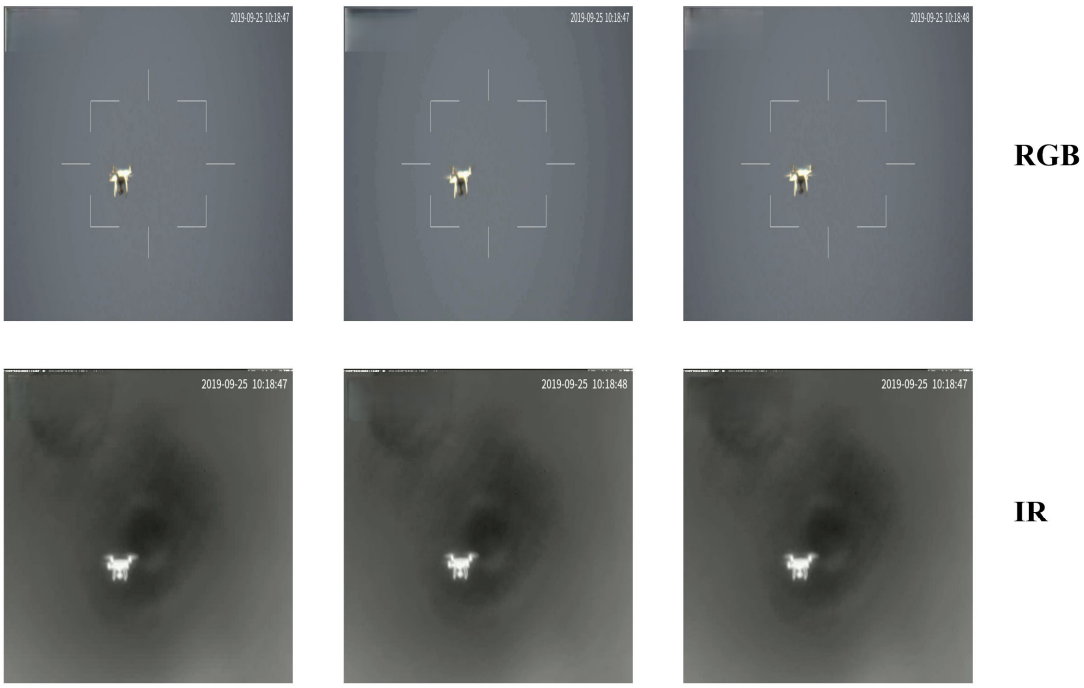


Figure 14. Datasets Samples.

4.2. Experiment Environment

The experiments in this paper were conducted using the Ubuntu 20.04 operating system with NVIDIA GeForce RTX 4090 GPUs. The programming language for the experi-

ments was Python 3.8, and the deep learning frameworks were PyTorch 2.0 and CUDA 11.8. The models were subjected to 300 rounds of iterations during the training process to ensure sufficient convergence. In order to ensure the fairness of the comparison of the experimental results, all the comparison algorithms and ablation experiments of the experiments in this paper were conducted under exactly the same experimental conditions and training parameters to ensure that the differences between the different models and methods only originate from the algorithms themselves, so as to provide a reliable basis for evaluating the performance of the models.

4.3. Evaluate Metrics

4.3.1. Metrics of Object Detection

In order to intuitively and comprehensively evaluate the performance of the improved network, this paper adopts the computational FLOPs, precision rate P, recall rate R, average detection precision mAP, and the number of detected frames per second FPS as the evaluation indexes. Among them, the smaller value of FLOPs indicates the smaller complexity of the algorithm; the higher value of P represents that the algorithm's detection results are more reliable with fewer false detections; the higher value of R represents that the algorithm is able to detect all targets as much as possible with fewer missed detections; the higher value of mAP indicates that the algorithm's detection precision is higher; and the higher value of FPS indicates that the algorithm's detection speed is faster. The formulas for the computation of P, R and mAP are as follows:

$$P = \frac{TP}{TP + FP} \tag{106}$$

$$R = \frac{TP}{TP + FN} \tag{107}$$

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{108}$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{109}$$

4.3.2. Metrics of MOT

In order to intuitively and accurately quantify the tracking accuracy and efficiency, and to provide a key basis for optimizing the model, taking into account a variety of factors, such as background error, error matching and omission error, this paper adopts six kinds of metrics to evaluate the tracking effect of the model, namely, MOTA, MOTP, FN, FP, IDS, and IDF1, of which MOTA, MOTP, and IDF1 are computed as shown below:

$$MOTA = 1 - \frac{FP + FN + IDS}{GT} \tag{110}$$

$$MOTP = \frac{\sum_{i=1}^{N} d_i}{N} \tag{111}$$

$$IDF1 = \frac{2 \times IDTP}{IDTP + IDFP + IDFN} \tag{112}$$

where FP is the number of false detections, i.e., the number of times the model incorrectly identifies a non-target object as a target object; FN is the number of missed detections, i.e., the number of times the model fails to detect a target object that is actually present; IDS is the number of identity exchanges, i.e., the number of times the model incorrectly changes

the target identity in a trajectory; GT is the total number of occurrences of all target objects. is the distance between the ith correctly matched target position and the true position; N is the total number of correct matches; IDTP is the number of correctly matched identities, IDFP is the number of incorrectly matched non-target identities, and IDFN is the number of unmatched target identities.

4.4. Results Analysis of Object Detection

4.4.1. Overall Comparative Analysis of Models

Figure 15c shows the total loss graph, it can be found that the improved YOLO v11 model and the traditional YOLO v11 model show significant differences in the training process. First of all, in the initial stage of training, the starting loss value of the improved YOLO v11 model (about 3.8) is significantly lower than that of the traditional model (about 12.2), which indicates that the initial parameter settings of the improved model are more reasonable, and it is able to enter into the effective learning state more quickly. Further observation of the training process reveals that both models show a decreasing trend in the loss value, but the improved model converges to a loss value of about 2.5 in about 100 epochs, while the traditional model requires about 100 epochs to reach a relatively stable state, and its loss value is still maintained at about 4.0. This characteristic indicates that the improved model has a more efficient parameter optimization ability, and can adapt to the complex motion characteristics of high mobility UAVs and adjust the network parameters faster, which significantly reduces the cost of training time. Analyzing the fluctuation trend of the loss curve, it can also be found that the fluctuation of the loss curve of the improved YOLO v11 model is relatively small, especially in the middle stage of training (20–100 epochs), where the curve is smoother and more stable. In contrast, the traditional model still exhibits more pronounced fluctuations at the same stage. This feature indicates that the improved model has better training stability, reduces the risk of overfitting, and improves the model's generalization ability to highly maneuverable UAVs under different attitudes, speeds, and lighting conditions. Near the completion of training (200–300 epochs), the loss values of the two models stabilize, but the final loss value of the improved YOLO v11 model (~2.3) is significantly lower than that of the conventional model (~3.2), with a loss reduction of about 28%. This means that the improved model has better characterization and prediction ability for high maneuvering UAV targets, and can more accurately identify and localize UAV targets in high-speed motion, effectively reducing the cases of missed detection and misdetection.

Figure 15a shows the results of the comparison of the mAP@50 metrics between the improved YOLO v11 model and the traditional YOLO v11 model in the fast identification and detection task of high maneuvering UAVs. It can be analyzed that the improved model (blue curve) shows significantly superior learning ability from the early stage of training, and rapidly improves to a mAP@50 value close to 0.99 within the first 20 epochs, while the traditional model (red curve) only reaches about 0.6–0.7 during the same period of time, which shows large fluctuations. This suggests that the improved model is able to learn the key feature representations of high maneuverability UAVs more quickly and stably in the initial phase, shortening the training time required for the model to reach a usable state. As the training process deepens (20–100 epochs), the mAP@50 curve of the improved YOLO v11 model rapidly stabilizes and remains at a high level around 0.99, showing high detection accuracy and stability, compared to the traditional model which still shows obvious fluctuations and a slow upward trend in this phase, proving the improved measures in the model training process continued effectiveness, enabling the improved model to more accurately detect UAV targets in high-speed motion. In the middle and late stages of training (100–300 epochs), the mAP@50 curves of both models tend to

stabilize, improved model mAP@50 up to 99%, while the upper limit of the performance of the traditional model is only about 0.85, with a performance enhancement ratio of about 16.5%, which indicates that the improved model has made a substantial breakthrough in the balance between detection rate and accuracy rate, effectively This indicates that the improved model has made a substantial breakthrough in the balance between detection rate and accuracy rate, effectively strengthening the model's ability to recognize high-mobility UAVs. It is worth noting that the mAP@50 curve of the improved model has almost no fluctuation, while the traditional model has fluctuation of different degrees throughout the training process. This stability difference further proves that the improved model has stronger feature extraction capability and higher training stability, and is able to maintain consistent detection performance under different attitudes, velocities, and background conditions, which is particularly important for real-time tracking and recognition of highly maneuverable UAVs.

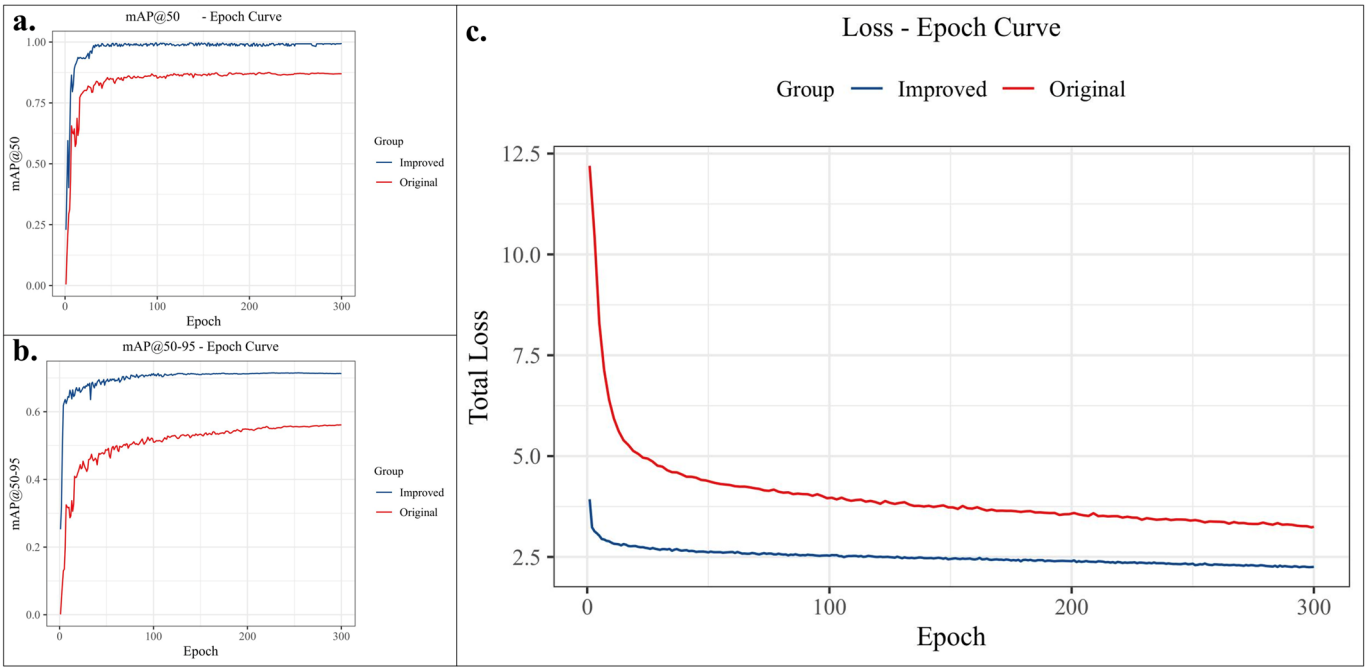


Figure 15. Overall Comparison of Loss, mAP@50, and mAP@50-95 Epoch Curves.

Figure 15b shows the results of the comparison between the improved YOLO v11 model and the traditional YOLO v11 model mAP@50-95 metrics. It can be found through the analysis that from the early stage of training, the improved model demonstrates a faster learning rate, which rapidly improves to a mAP@50-95 value of about 0.62 within the first 25 epochs, whereas the traditional model only reaches about 0.33, indicating that it is able to learn the precise positional feature representations of the highly maneuverable UAVs more quickly in the initial stage. As the training process progresses (25–100 Epochs), the upward trend of the two curves gradually slows down, but the improved YOLO v11 model consistently maintains relatively high mAP@50-95 values, indicating that the improvements continue to play a role in the training process of the model, which enables the model to more accurately detect the precise position and silhouette of the high-maneuvering UAVs. In the late stage of training (100–300 Epochs), the mAP@50-95 curve of the improved model tends to stabilize, finally, mAP@50-95 stop in 0.72, while the upper limit of the performance of the traditional model is only about 0.56, with a performance enhancement ratio of about 28.6%, which strengthens the model's ability to accurately identify and localize the high-mobility

UAVs, proving that in the feature extraction ability, bounding box regression accuracy and classification accuracy of the model optimization effectiveness.

4.4.2. Ablation Study

To further verify the validity and rigor of the improved method in this paper, a series of ablation experiments were designed in this study. Each group of experiments was performed under the same parameter settings to ensure comparable results. The specific experimental settings are as follows:

Baseline model: YOLOv11n is used as the baseline model to record its original performance on the dataset.

(1) Group 2 experiment: The C3k2-DATB module is introduced into YOLOv11n to replace the original ordinary convolution.

(2) Group 3 experiment: add BRSA attention mechanism module

(3) Group 4 experiment: replace the traditional CONTACT connection with the Semantic and Detail Injection module.

(4) Group 5 experiment: replace the original detection head with PCHead.

(5) Group 6 experiment: replace the original CIoU loss function with Wasserstein Distance LOSS loss function.

Through these ablation experiments, this paper systematically analyzes the impact of each improvement on the model performance, thus verifying the effectiveness of this paper's method in improving the detection accuracy, optimizing the computational efficiency and enhancing the model robustness. The experimental results are shown in Table 2.

Table 2. Ablation Study (* indicates that this module is being added into model).

	C3k2-DATB	BRSA	SDI	PCHead	WD-Loss	Params/M	GFLOPS	mAP@50	mAP@50_95
1						2.91	7.7	81.9	56.1
2	*					2.92	7.9	82.5	57.9
3	*	*				2.13	6.2	87.6	65.4
4	*	*	*			2.31	6.5	91.4	69.3
5	*	*	*	*		2.72	7.3	95.7	70.8
6	*	*	*	*	*	2.54	7.8	99.3	71.3

Our systematic ablation study reveals progressive performance improvements through sequential integration of five components into the YOLOv11 baseline (81.9% mAP@50, 56.1% mAP@50-95). The C3k2-DATB module enhances feature extraction via dynamic attention, yielding +0.6% mAP@50 and +1.8% mAP@50-95 with negligible complexity increase (Δ + 0.01M parameters). Subsequent BRSA implementation establishes explicit feature dependencies, achieving significant gains (+5.1% mAP@50, +7.5% mAP@50-95) while reducing parameters by 27.1% (2.13M) and computations by 21.5% (6.2 GFLOPS). The SDI module further elevates performance to 91.4% mAP@50 (+3.8%) through optimized feature fusion, followed by PCHead's windmill-shaped convolution expanding receptive fields to reach 95.7% mAP@50 (+4.3%). Final WD-Loss integration optimizes regression accuracy, culminating in 99.3% mAP@50 (+3.6%) and 71.3% mAP@50-95 (+0.5%) with practical efficiency (2.54M parameters, 7.8 GFLOPS). Collectively, these innovations deliver 17.4% mAP@50 and 15.2% mAP@50-95 improvements over baseline while maintaining deployability, demonstrating robust adaptation to high-maneuverability UAV detection scenarios.

4.4.3. Multi Model Comparison

To further validate the performance of the improved YOLOv11 model in the rapid identification and detection of highly maneuverable UAVs, representative models such

as RT-DETR and other YOLO series models were selected for comparative experiments. ** The experimental results are shown in Table 3.

Table 3. Multi Model Comparison.

Model	Precision/%	Recall/%	Params/M	GFLOPS	mAP@50	mAP@50_95
RT-DETR	65.3	83.4	427.6	130.5	71.2	43.1
YOLOv5	67.2	81.5	2.5	7.2	73.5	45.6
YOLOv6	66.5	81.9	4.2	11.9	77.1	44.3
YOLOv8	70.1	80.3	3.1	8.2	81.5	53.7
YOLOv9	70.8	79.4	2	7.8	76.7	51.2
YOLOv10	73.7	82.6	2.71	8.4	80.2	49.8
YOLOv11	74.1	77.2	2.91	7.7	81.9	56.1
Improved-YOLOv11	98.9	98.5	2.54	7.8	99.3	71.3

Comparison and analysis of the Improved-YOLOv11 model with other classic object detection models in the rapid detection task of highly maneuverable UAVs show that the Improved-YOLOv11 model has significant advantages in key indicators such as accuracy, speed, and model complexity. ** Specifically, the detection accuracy of the Improved-YOLOv11 network reaches 98.9%, an increase of 24.8 percentage points compared to the original YOLOv11 version, while the Precision of other comparative models is generally below 75%, proving that the positioning and classification capabilities of the Improved-YOLOv11 model have been significantly enhanced and the identification of UAV targets in high-speed motion is more accurate. In terms of the Recall indicator, the Improved-YOLOv11 model reaches 98.5%, an increase of 21.3 percentage points compared to the original YOLOv11 version (77.2%), and also surpasses the traditional RT-DETR model, which is known for its high recall rate (83.4%), achieving a dual breakthrough in high precision and high recall rate and the best overall detection performance. This is mainly due to the in-depth optimization of the feature extraction mechanism, multi-scale feature fusion, and loss function.

A deeper analysis of the mAP evaluation indicators shows that the Improved-YOLOv11 model has an mAP@50 of 99.3%, an increase of 17.4 percentage points compared to the original YOLOv11, and 17.8 percentage points higher than the best-performing YOLOv8 model (81.5%) among other models, reflecting a high level of detection accuracy. ** Under the stricter mAP@50-95 evaluation criteria, the Improved-YOLOv11 model reaches 71.3%, an increase of 15.2 percentage points compared to the original YOLOv11 (56.1%), and leads the second-placed YOLOv8 model (53.7%) by 17.6 percentage points. The above results show the excellent performance of the improved model in precise target localization, which can accurately capture the position and outline of UAVs in high-speed motion, providing a reliable basis for subsequent situation analysis and behavior prediction.

From the perspective of model parameters and computational efficiency, the Improved-YOLOv11 model has only 2.54M parameters, a reduction of 12.7% compared to the original YOLOv11's 2.91M. ** Among all the models compared horizontally, it is only surpassed by YOLOv9 (2.0M) and YOLOv5 (2.5M), and is significantly lower than models such as RT-DETR (427.6M) and YOLOv6 (4.2M). In terms of computational complexity, the GFLOPS of the improved model is 7.8, basically the same as the original YOLOv11 (7.7), and much lower than models such as RT-DETR (130.5) and YOLOv6 (11.9). This indicates that the Improved-YOLOv11 model has effectively controlled computational costs while significantly improving detection performance, achieving the ideal balance of "high performance-low complexity". This advantage is mainly due to the design and optimization of key components such as the C3k2-DATB module, BRSA attention mechanism, and

SDI semantic injection module, which enhance feature representation capabilities while maximizing the compression of redundant calculations.

In summary, the series of in-depth optimizations carried out in this study on the core components of the YOLOv11 model, such as feature extraction, multi-scale feature fusion, loss function, and detection head, have effectively improved the model's performance in the rapid detection task of highly maneuverable UAVs. ** It has comprehensively surpassed the compared classic object detection models in key indicators such as accuracy, speed, and model complexity.

4.4.4. Multi-Model Scenario Application Comparison

To intuitively demonstrate the superiority of the Improved-YOLOv11 model in the rapid detection of highly maneuverable UAVs, this study selected five sets of visible light and infrared images under different brightness and contrast conditions during the day and night. ** Four models, namely YOLOv6, YOLOv8, the traditional YOLOv11, and the improved Improved-YOLOv11, were used to conduct a horizontal comparative experiment in UAV detection scenarios, and the results are shown in Figure 16. The detection results are presented in the form of bounding boxes, which include UAV labels and corresponding confidence scores.

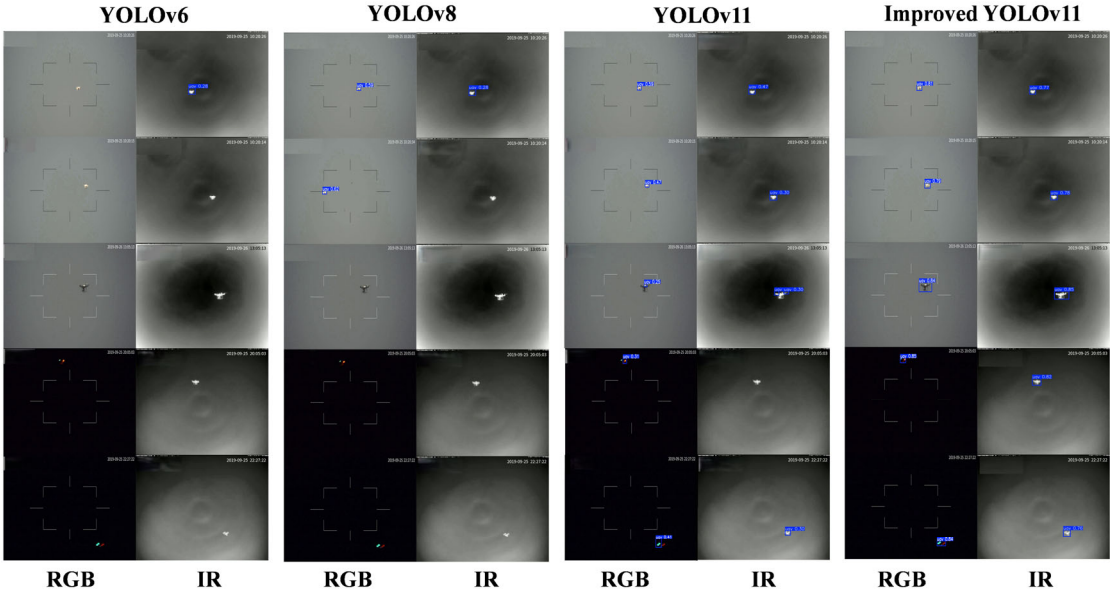


Figure 16. Scenario comparison. (The blue mark indicates the drone recognition detection frame and confidence level).

Analysis reveals that while all four models achieved basic target recognition in the task of identifying highly maneuverable UAVs, there were significant performance differences. When comparing detection results under different lighting and sensor conditions, the improved YOLOv11 model demonstrated a clear advantage.

Under daytime visible light conditions (rows 1–3, columns 1, 3, 5, and 7), although the YOLOv6 and YOLOv8 models could detect UAV targets, their confidence scores were generally low (all below 0.65), and there were severe cases of missed detections. The original YOLOv11 had slightly higher detection confidence for highly maneuverable UAVs, but it suffered from imprecise bounding box localization under low-contrast conditions. In contrast, the improved YOLOv11 model not only successfully detected UAV targets in all scenarios but also achieved confidence scores of 0.79 or higher. Moreover, its bounding boxes had the highest degree of fit with the actual target contours, demonstrating superior localization accuracy and recognition reliability. In the infrared image recognition scenarios

(rows 1–3, columns 2, 4, 6, and 8), the performance differences among the models were particularly pronounced. YOLOv6 and YOLOv8 had severe missed detection issues in low-contrast infrared images. Although the original YOLOv11 could detect the main targets, its detection boxes were oversized, misaligned, and had low confidence scores (below 0.5). The improved YOLOv11 model performed excellently in all infrared scenarios, not only precisely locating the targets (with confidence scores above 0.75) but also adapting to different thermal imaging features, showing strong cross-modal recognition capabilities.

Under low-light night-time conditions (rows 4–5), the performance differences were even more significant. YOLOv6 and YOLOv8 failed to detect targets in most night-time scenarios. The original YOLOv11 could barely recognize the targets under night-time visible light conditions, but its confidence scores dropped significantly (to around 0.4), and the detection boxes were severely biased. The improved YOLOv11 model maintained high detection performance in all night-time scenarios, with stable confidence scores around 0.85. It was the only model that could maintain stable detection performance under extremely low-light conditions (row 5). In infrared night vision conditions (rows 4–5, columns 2, 4, 6, and 8), the advantages of the improved YOLOv11 model were even more evident. In the extreme low signal-to-noise ratio scenario in row 5, YOLOv6, YOLOv8, and the original YOLOv11 all had missed detections, while the improved YOLOv11 not only successfully detected the targets (with a confidence score of 0.82) but also accurately delineated the target contours. This is of great significance for the monitoring of covert UAVs at night.

In summary, the improved YOLOv11 model, through the introduction of a series of optimizations such as the C3k2-DATB module, BRSA attention mechanism, and SDI semantic injection module, has effectively enhanced the model's target detection capabilities under various lighting conditions and across sensor modalities. ** Compared to models such as YOLOv6, YOLOv8, and the original YOLOv11, the improved YOLOv11 has demonstrated significant advantages in detection confidence, bounding box precision, and adaptability to complex environments. Its robustness in low-light and low-contrast scenarios provides key technical support for an all-weather UAV monitoring system.

4.5. Analysis of the Results of the Target Tracking Experiment

Figure 17 demonstrates the detection and tracking results of the improved BoT-SORT real-time object tracking algorithm for highly maneuverable drones under infrared and visible light conditions, with quantitative evaluation indicators provided in Table 4. Through comprehensive analysis of visual results and performance metrics, the tracking performance of the improved algorithm under different sensor modalities can be objectively evaluated.

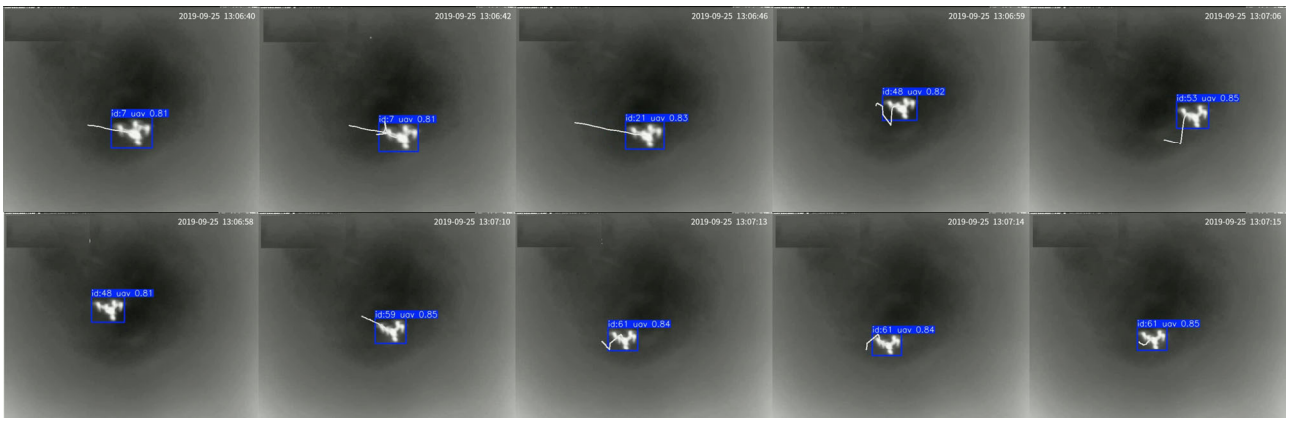


Figure 17. MOT Application. (The blue mark indicates the drone recognition detection frame and confidence level).

Table 4. Performance Analysis.

Video	MOTA/%	MOTP/%	FN	FP	IDS	IDF1/%
MOT01(IR)	93.2	87.5	5	0	15	93.5
MOT01(RGB)	89.4	82.7	7	0	17	91.4
Overall	91.3	85.1	12	0	5.0	93.0

From the visual results, the improved BoT-SORT algorithm exhibits stable target tracking capability in infrared image sequences (the second row of Figure 16), maintaining consistent detection of drone targets across consecutive frames. The bounding boxes closely fit the target contours, ID labels remain consistent, and confidence scores stabilize between 0.81 and 0.85. It sustains tracking during rapid target maneuvers (Frames 4 and 5), indicating strong motion prediction ability. In visible light sequences (the second row of Figure 16), despite relatively low image contrast, the algorithm accurately captures and tracks targets with consistent ID labels and confidence scores ranging from 0.83 to 0.84, demonstrating adaptability to different imaging conditions.

Quantitative evaluation results in Table 3 show that the improved BoT-SORT algorithm achieves a multi-object tracking accuracy (MOTA) of 93.2% and a multi-object tracking precision (MOTP) of 87.5% in infrared video tracking (MOT01(IR)), significantly outperforming the visible light video (MOT01(RGB)) with MOTA of 89.4% and MOTP of 82.7%. This indicates better tracking accuracy and positioning precision under infrared conditions, likely due to the more distinct thermal contrast between targets and backgrounds in infrared images, reducing complex background interference.

In terms of false detection analysis, the algorithm avoids false positive detections (FP values were both 0) in both modalities, demonstrating that the improved matching strategy effectively suppresses false alarms. False negatives (FN) occurred in 5 frames for infrared video and 7 frames for visible light video, resulting in a low overall false negative rate (12 total FN). Notably, the number of ID switches (IDS) was similar between infrared (15 times) and visible light (17 times) videos; considering the highly maneuverable nature of tracked targets, this level of ID switching is within an acceptable range.

Regarding comprehensive evaluation indicators, the ID F1-score (IDF1) reaches 93.5% under infrared conditions, slightly higher than 91.4% under visible light conditions, indicating that infrared modalities are more conducive to consistent target identity recognition. Overall, the improved BoT-SORT algorithm performs excellently under both sensor modalities, with an overall MOTA of 91.3%, MOTP of 85.1%, and IDF1 of 93.0%, verifying its robustness and effectiveness in multi-modal conditions.

5. Discussion

This study addresses the technical challenges of rapid identification, detection, and tracking of highly maneuverable drones by proposing a detection algorithm based on the improved YOLOv11 network, which enables precise recognition and positioning of high-speed moving drones under multi-illumination conditions and cross-sensor modalities. Meanwhile, an improved BoT-SORT real-time object tracking algorithm is designed to validate the feasibility of real-time tracking of highly maneuverable UAVs and effectively resolve the technical bottlenecks of traditional detection methods, such as insufficient recognition accuracy in complex backgrounds and low-light environments, providing a specific technical pathway for all-weather high-precision drone monitoring.

Although the proposed scheme demonstrates excellent performance in the identification and detection of highly maneuverable drones, there remain areas for improvement in engineering practice. First, while the existing dataset includes multiple illumination

conditions and sensor modalities, the sample size under extreme meteorological environments (such as low visibility, rain, snow, etc.) is insufficient, which moderately limits the model's generalization ability in harsh conditions. Second, the current analysis of real-time detection performance primarily focuses on the model itself, with no in-depth exploration of end-edge-cloud collaborative detection mechanisms in distributed deployment scenarios. Third, constrained by the computational capabilities of existing hardware, although the improved model has significantly reduced parameter count and computational complexity, there is still room for optimization in deploying it on low-power edge devices. Finally, small and remote airports typically utilize stitched surveillance footage with limited field of view, necessitating rapid tracking when highly maneuverable UAVs exit the monitored image boundaries to confirm their departure from airport protection zones. The improved target tracking algorithm presented in this paper serves merely as a complementary component to the UAV identification and detection work, highlighting the practical significance of our research, rather than being subjected to an in-depth comparative study.

Subsequent research will focus on the following breakthrough directions:

1. Construct a more comprehensive dataset of highly maneuverable drones across multi-environment and multi-scene scenarios, and introduce multi-modal cross-domain adaptive strategies to enhance the model's adaptability and generalization ability in extreme environments (such as low visibility, rain, snow, etc.), thereby improving the robustness of the detection system in uncontrolled conditions;
2. Leveraging the emerging advantages of large language models (LLM) in intelligent parsing, explore methods for fusing LLMs with target detection models to enable intelligent analysis and prediction of the behavioral intentions of highly maneuverable drones, thus promoting the detection system's capability upgrade from the perception layer to the cognitive decision-making layer;
3. Conduct in-depth comparative evaluations of detection efficiency between edge and cloud deployments, optimize model quantization and pruning strategies to enhance edge computing capabilities, while designing distributed detection network architectures to achieve collaborative monitoring across multiple nodes, thereby expanding surveillance coverage and improving the overall monitoring effectiveness of the system;
4. Further investigate the effectiveness of target tracking detection algorithms and incorporate temporal information into the detection process. Integrate temporal dynamic information into the detection process, combined with trajectory prediction techniques, to achieve accurate prediction of the motion trajecto-ries and future position estimation of highly maneuverable drones, and construct an intelligent early-warning system that integrates detection, tracking, and prediction. This will provide a more sufficient response time window for the rapid identification and dispelling of drones in airport airspace.

6. Conclusions

To address the technical challenges of high maneuverable drone identification, such as difficult recognition, low detection accuracy, and strict real-time requirements, this study proposes a rapid identification and detection algorithm for highly maneuverable drones based on an improved YOLOv11 network. A dual-path fusion image input architecture for RGB visible light and IR infrared is designed to strengthen information fusion and complementarity under multi-modal and multi-scale conditions, thereby enhancing the model's feature extraction capability for high-speed moving targets in complex backgrounds. The DATB dynamic attention module is embedded into the C3k2 structure to achieve dynamic reconstruction of feature space and channels, effectively capturing the key features of highly

maneuverable drones. A bi-level attention routing mechanism based on proxy queries (BRSA) is designed to significantly enhance the model's perception of global contextual information while reducing model parameters and computational complexity. The semantic and detail injection (SDI) module replaces traditional feature connection methods to optimize the feature fusion process across different scales, effectively solving the problems of information conflicts and feature dilution caused by simple feature concatenation. A detection head based on windmill-shaped convolution (PCHead) is designed to expand the receptive field while maintaining computational efficiency, and the Wasserstein Distance loss function is introduced to significantly improve the regression accuracy of bounding boxes and model convergence speed by accurately measuring the distribution difference between predicted and ground-truth boxes, thus achieving all-weather and cross-modal high-precision drone identification.

To verify the system's practicality and account for UAV motion characteristics, we propose an improved BoT-SORT algorithm based on interactive multiple model filtering, enabling rapid identification and tracking of non-cooperative UAVs. This algorithm integrates multiple motion models to predict target states, achieves adaptive tracking of complex drone motion patterns through probabilistic interaction and weight fusion among models, and compensates for predicted states using sparse optical flow to estimate camera motion parameters, solving the problem of tracking drift in dynamic camera environments.

The Mosaic data augmentation technique is used to perform random cropping, stitching, and color adjustment on images from the Anti-UAV dataset. Ablation experiments, horizontal model comparison experiments, and scenario application contrast tests are designed to verify the effectiveness of the improved model. Comprehensive comparison results show that the improved YOLOv11 model increases the mAP@50 to 99.3%, representing improvements of 17.4 and 17.8 percentage points over the standard YOLOv11 and the best-performing YOLOv8, respectively. Under the stricter mAP@50-95 evaluation criterion, it reaches 71.3%, leading the second-place model by 17.6 percentage points. Meanwhile, the improved model maintains lightweight characteristics (only 2.54 million parameters) and high computational efficiency (7.8 GFLOPS), significantly reducing computational costs compared to heavyweight models such as RT-DETR.

In multi-scenario application comparisons, the improved YOLOv11 model performs excellently, especially under low-light night vision conditions, where the detection confidence remains stable at approximately 0.85, and the bounding box positioning accuracy is significantly better than that of other comparative models. Tracking performance evaluation indicates that the improved BoT-SORT algorithm demonstrates strong cross-modal tracking capabilities in infrared and visible light drone video detection, achieving a comprehensive MOTA of 91.3%, MOTP of 85.1%, and IDF1 of 93.0%, providing reliable technical support for drone monitoring in small and medium-sized remote airports.

Future research will further expand the scale of extreme environment datasets (such as low visibility, rain, snow, etc.), explore the fusion mechanism between large language models and detection models, and incorporate temporal information into the analysis framework to achieve intelligent parsing and prediction of the behavioral intentions of highly maneuverable drones. Additionally, the distributed detection network architecture will be optimized to expand monitoring coverage, providing a more comprehensive drone monitoring system for small and medium-sized remote airports and ensuring the safety of airport airspace.

Author Contributions: Conceptualization, Y.Z. and W.P.; Data curation, T.L.; Funding acquisition, W.P.; Investigation, T.L. and S.Z.; Methodology, Y.Z., W.P. and T.L.; Project administration, W.P.; Resources, T.L.; Software, Y.Z., T.L. and S.Z.; Supervision, Y.Z. and S.Z.; Validation, Y.Z., T.L. and

S.Z.; Writing—original draft, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (U2333209); This research was funded by the Civil Aviation Flight Technology and Flight Safety Engineering Technology Research Institute of Sichuan Province (GY2024-29D); This research was funded by the Civil Aircraft Fire Science and Safety Engineering Key Laboratory of Sichuan Province (MZ2024JB01).

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to restriction.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. CBS News. Gatwick Airport Drones: 2 Arrested for Suspected "Criminal Use of Drones". 22 December 2018. Available online: https://www.cbsnews.com/news/gatwick-airport-drones-2-arrested-for-suspected-criminal-use-of-drones-today-2018-12-22/ (accessed on 22 December 2018).
2. CCTV. Tianjin Airport: Flight Operations Affected due to Public Safety Reasons Caused by Drones. *CCTV News. 12 September 2024.* Available online: https://news.cctv.com/2024/09/12/ARTIuf566famown5rIcmvsaP240912.shtml (accessed on 12 September 2024).
3. Federal Aviation Administration (FAA). UAS Sightings Report. 2022. Available online: https://www.faa.gov/uas/resources/public_records/uas_sightings_report (accessed on 1 April 2025).
4. Chiper, F.; Marţian, A.; Vlădeanu, C.; Marghescu, I.; Craciunescu, R.; Fratu, O. Drone Detection and Defense Systems: Survey and a Software-Defined Radio-Based Solution. *Sensors* **2022**, *22*, 1453. [CrossRef] [PubMed]
5. Al-Sa'D, M.; Al-Ali, A.; Mohamed, A.; Khattab, T.; Erbad, A. RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database. *Future Gener. Comput. Syst.* **2019**, *100*, 86–97. [CrossRef]
6. Singha, S.; Aydin, B. Automated Drone Detection Using YOLOv4. *Drones* **2021**, *5*, 95. [CrossRef]
7. Peng, G.; Yang, Z.; Wang, S.; Zhou, Y. AMFLW-YOLO: A Lightweight Network for Remote Sensing Image Detection Based on Attention Mechanism and Multiscale Feature Fusion. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4600916. [CrossRef]
8. Ye, Y.; Ren, X.; Zhu, B.; Tang, T.; Tan, X.; Gui, Y.; Yao, Q. An Adaptive Attention Fusion Mechanism Convolutional Network for Object Detection in Remote Sensing Images. *Remote Sens.* **2022**, *14*, 516. [CrossRef]
9. Chen, X.; Li, Y.; Nakatoh, Y. Pyramid attention object detection network with multi-scale feature fusion. *Comput. Electr. Eng.* **2020**, *104*, 108436. [CrossRef]
10. Grác, Š.; Beňo, P.; Duchoň, F.; Dekan, M.; Tölgyessy, M. Automated detection of multi-rotor UAVs using a machine-learning approach. *Appl. Syst. Innov.* **2020**, *3*, 29. [CrossRef]
11. Hoffmann, F.; Ritchie, M.; Fioranelli, F.; Charlish, A.; Griffiths, H. Micro-Doppler based detection and tracking of UAVs with multistatic radar. In Proceedings of the 2016 IEEE Radar Conference (RadarConf), Philadelphia, PA, USA, 2–6 May 2016; pp. 1–6. [CrossRef]
12. Hengy, S.; Laurenzis, M.; Schertzer, S.; Hommes, A.; Kloeppel, F.; Shoykhetbrod, A.; Geibig, T.; Johannes, W.; Rassy, O.; Christnacher, F. Multimodal UAV detection: Study of various intrusion scenarios. In Proceedings of the Electro-Optical Remote Sensing XI, Warsaw, Poland, 11–14 September 2017; Volume 10434. [CrossRef]
13. Qin, F.; Bu, X.; Zeng, Z.; Dang, X.; Liang, X. Small Target Detection for FOD Millimeter-Wave Radar Based on Compressed Imaging. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4020705. [CrossRef]
14. Nguyen, P.; Ravindranathan, M.; Nguyen, A.; Han, R.; Vu, T. Investigating Cost-effective RF-based Detection of Drones. In Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use, Singapore, 26 June 2016. [CrossRef]
15. Medaiyese, O.; Syed, A.; Lauf, A. Machine Learning Framework for RF-Based Drone Detection and Identification System. In Proceedings of the 2021 2nd International Conference On Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), Tangerang, Indonesia, 12–13 October 2021; pp. 58–64. [CrossRef]
16. Sciancalepore, S.; Ibrahim, O.; Oligeri, G.; Pietro, R. Detecting Drones Status via Encrypted Traffic Analysis. In Proceedings of the ACM Workshop on Wireless Security and Machine Learning, Miami, FL, USA, 15–17 May 2019. [CrossRef]
17. Svanström, F.; Alonso-Fernandez, F.; Englund, C. Drone Detection and Tracking in Real-Time by Fusion of Different Sensing Modalities. *arXiv* **2022**, arXiv:2207.01927. [CrossRef]
18. Hammer, M.; Borgmann, B.; Hebel, M.; Arens, M. A multi-sensorial approach for the protection of operational vehicles by detection and classification of small flying objects. In Proceedings of the Electro-Optical Remote Sensing XIV, Online, 21–25 September 2020; Volume 11538, pp. 1153807–1153812. [CrossRef]

19. Wang, D.; Yang, Z.; Chen, F. Dual-modality Object Detection Approach Utilizing Enhanced and Fused Features. In Proceedings of the 2024 IEEE 6th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Hangzhou, China, 23–25 October 2024; pp. 709–715. [CrossRef]
20. Hiba, A.; Gáti, A.; Manecy, A. Optical navigation sensor for runway relative positioning of aircraft during final approach. *Sensors* **2021**, *21*, 2203. [CrossRef]
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *60*, 84–90. [CrossRef]
22. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
23. Zhao, J.; Zhang, J.; Li, D.; Wang, D. Vision-Based Anti-UAV Detection and Tracking. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 25323–25334. [CrossRef]
24. Munir, A.; Siddiqui, A.J. Vision-based UAV Detection in Complex Backgrounds and Rainy Conditions. In Proceedings of the 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 15–16 March 2024; pp. 1097–1102. [CrossRef]
25. Shovon MH, I.; Gopalan, R.; Campbell, B. A comparative analysis of deep learning algorithms for optical drone detection. In Proceedings of the Fifteenth International Conference on Machine Vision (ICMV 2022), Rome, Italy, 18–20 November 2022; Volume 12701, pp. 1270104–1270108. [CrossRef]
26. Xun DT, W.; Lim, Y.L.; Srigrarom, S. Drone detection using YOLOv3 with transfer learning on NVIDIA Jetson TX2. In Proceedings of the 2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 20–22 January 2021; pp. 1–6. [CrossRef]
27. Isaac-Medina BK, S.; Poyser, M.; Organisciak, D.; Willcocks, C.G.; Breckon, T.; Shum, H.P.H. Unmanned Aerial Vehicle Visual Detection and Tracking using Deep Neural Networks: A Performance Benchmark. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 1223–1232. [CrossRef]
28. Hakani, R.; Rawat, A. Edge Computing-Driven Real-Time Drone Detection Using YOLOv9 and NVIDIA Jetson Nano. *Drones* **2024**, *8*, 680. [CrossRef]
29. Liu, J.; Zhang, F.; Zhao, H.; Lu, Q.D.; Feng, B.; Feng, L. Recognition and Detection of UAV Based on Transfer Learning. In Proceedings of the 2023 8th International Conference on Multimedia and Image Processing, Tianjin, China, 21–23 April 2023. [CrossRef]
30. Khan, M.U.; Dil, M.; Misbah, M.; Orakazi, F.A.; Alam, M.Z.; Kaleem, Z. TransLearn-YOLOx: Improved-YOLO with Transfer Learning for Fast and Accurate Multiclass UAV Detection. In Proceedings of the 2023 International Conference on Communication, Computing and Digital Systems (C-CODE), Islamabad, Pakistan, 17–18 May 2023; pp. 1–7. [CrossRef]
31. Liu, H.; Fan, K.; Ouyang, Q.; Li, N. Real-Time Small Drones Detection Based on Pruned YOLOv4. *Sensors* **2021**, *21*, 3374. [CrossRef]
32. Zamri, F.; Gunawan, T.; Yusoff, S.; Alzahrani, A.; Bramantoro, A.; Kartiwi, M. Enhanced Small Drone Detection Using Optimized YOLOv8 With Attention Mechanisms. *IEEE Access* **2024**, *12*, 90629–90643. [CrossRef]
33. Gökçe, F.; Üçoluk, G.; Şahin, E.; Kalkan, S. Vision-based detection and distance estimation of micro unmanned aerial vehicles. *Sensors* **2015**, *15*, 23805–23846. [CrossRef]
34. Wang, C.; Meng, L.; Gao, Q.; Wang, J.; Wang, T.; Liu, X.; Du, F.; Wang, L.; Wang, E. A Lightweight Uav Swarm Detection Method Integrated Attention Mechanism. *Drones* **2023**, *7*, 13. [CrossRef]
35. Tian, X.; Jia, Y.; Luo, X.; Yin, J. Small Target Recognition and Tracking Based on UAV Platform. *Sensors* **2022**, *22*, 6579. [CrossRef]
36. Wu, H.; Nie, J.; He, Z.; Zhu, Z.; Gao, M. One-Shot Multiple Object Tracking in UAV Videos Using Task-Specific Fine-Grained Features. *Remote Sens.* **2022**, *14*, 3853. [CrossRef]
37. Keawboontan, T.; Thammawichai, M. Toward Real-Time UAV Multi-Target Tracking Using Joint Detection and Tracking. *IEEE Access* **2023**, *11*, 65238–65254. [CrossRef]
38. Ma, J.; Liu, D.; Qin, S.; Jia, G.; Zhang, J.; Xu, Z. An Asymmetric Feature Enhancement Network for Multiple Object Tracking of Unmanned Aerial Vehicle. *Remote Sens.* **2023**, *16*, 70. [CrossRef]
39. Wang, P.; Wang, Y.; Li, D. DroneMOT: Drone-based Multi-Object Tracking Considering Detection Difficulties and Simultaneous Moving of Drones and Objects. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; pp. 7397–7404. [CrossRef]
40. Ma, J.; Tang, C.; Wu, F.; Zhao, C.; Zhang, J.; Xu, Z. STCMOT: Spatio-Temporal Cohesion Learning for UAV-Based Multiple Object Tracking. In Proceedings of the 2024 IEEE International Conference on Multimedia and Expo (ICME), Niagara Falls, ON, Canada, 15–19 July 2024; pp. 1–6. [CrossRef]

41. Wen, J.; Wang, D.; Fang, J.; Li, Y.; Xu, Z. Multi-Object Tracking for Unmanned Aerial Vehicles Based on Multi-Frame Feature Fusion. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 4180–4184. [CrossRef]
42. Zhang, B.; Su, W.; Lu, G.; Zang, D.; Li, X. Research on Drone Multi-Target Tracking Algorithm Based on Pseudo Depth. In Proceedings of the 2024 3rd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC), Mianyang, China, 5–7 July 2024; pp. 485–489. [CrossRef]
43. Yao, M.; Peng, J.; He, Q.; Peng, B.; Chen, H.; Chi, M.; Liu, C.; Benediktsson, J. MM-Tracker: Motion Mamba for UAV-platform Multiple Object Tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025; pp. 9409–9417. [CrossRef]
44. Yavariabdi, A.; Kusetogullari, H.; Çelik, T.; Cicek, H. FastUAV-NET: A Multi-UAV Detection Algorithm for Embedded Platforms. *Electronics* **2021**, *10*, 724. [CrossRef]
45. Ntousis, O.; Makris, E.; Tsanakas, P.; Pavlatos, C. A Dual-Stage Processing Architecture for Unmanned Aerial Vehicle Object Detection and Tracking Using Lightweight Onboard and Ground Server Computations. *Technologies* **2025**, *13*, 35. [CrossRef]
46. Liu, S.; Li, X.; Lu, H.; He, Y. Multi-Object Tracking Meets Moving UAV. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 8866–8875. [CrossRef]
47. Song, I.; Lee, J. SFTrack: A Robust Scale and Motion Adaptive Algorithm for Tracking Small and Fast Moving Objects. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; pp. 10870–10877. [CrossRef]
48. Ji, D.; Gao, S.; Zhu, L.; Zhao, Y.; Xu, P.; Lu, H.; Zhao, F. View-Centric Multi-Object Tracking with Homographic Matching in Moving UAV. *arXiv* **2024**, arXiv:2403.10830. [CrossRef]
49. Wu, X.; Li, W.; Hong, D.; Tao, R.; Du, Q. Deep Learning for Unmanned Aerial Vehicle-Based Object Detection and Tracking: A survey. *IEEE Geosci. Remote Sens. Mag.* **2021**, *10*, 91–124. [CrossRef]
50. Jegham, N.; Koh, C.Y.; Abdelatti, M.; Hendawi, A. Evaluating the Evolution of Yolo (You Only Look Once) Models: A Comprehensive Benchmark Study of Yolo11 and Its Predecessors. *arXiv* **2024**, arXiv:2411.00201. *preprint*. [CrossRef]
51. Li, J.; Zhang, Z.; Zuo, W. Rethinking Transformer-Based Blind-Spot Network for Self-Supervised Image Denoising. *arXiv* **2024**. [CrossRef]
52. Nguyen, L.; Zhang, C.; Shi, Y.; Hirakawa, T.; Yamashita, T.; Matsui, T.; Fujiyoshi, H. DeBiFormer: Vision Transformer with Deformable Agent Bi-level Routing Attention. *arXiv* **2024**. [CrossRef]
53. Liu, H.; Peng, B.; Ding, P.; Wang, D. Rethinking Target Label Conditioning in Adversarial Attacks: A 2D Tensor-Guided Generative Approach. *arXiv* **2025**. [CrossRef]
54. Yang, J.; Liu, S.; Wu, J.; Su, X.; Hai, N.; Huang, X. Pinwheel-shaped Convolution and Scale-based Dynamic Loss for Infrared Small Target Detection. *arXiv* **2024**. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.