Computer Vision and Image Understanding 264 (2026) 104637

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

UAVDet: A CNN–Mamba hybrid network for efficient small object detection in UAV imagery

Yiming Yang, Feng Guo *, Pei Niu

School of Qilu Transportation, Shandong University, Jinan, Shandong 250000, China

ARTICLE INFO

Communicated by Ribana Roscher

Keywords:
UAV object detection
Small objects
Complex backgrounds
Mamba structure
Real-time detection

ABSTRACT

Real-time object detection is pivotal in traffic-related Unmanned Aerial Vehicles (UAV) applications. However, UAV imagery presents significant challenges due to the predominance of small objects and complex backgrounds. Traditional backbones generally perform aggressive early-stage downsampling, causing the loss of fine-grained features. To address these issues, we propose UAVDet, a real-time detection model that combines Convolutional Neural Network (CNN) and Mamba architectures. First, we revisit the conventional backbone design by reconfiguring its depth and width, with a focus on preserving fine-grained details crucial for small object detection. Second, we propose the Cross Stage Partial Mamba (CSPMB) module, which integrates the Mamba structure into the CNN framework to enhance global feature representation and improve robustness against complex background interference. Third, we design Tiny-focused Feature Pyramid Network (TFPN) by rebalancing the feature fusion flow and replacing the large-object detection head with a tiny-object detection head, which significantly improves the perception of small objects. Comprehensive experiments on the VisDrone dataset show that our method improves AP and AP_S by 4.5% and 5.0%, respectively, while reducing parameters by 84.9% compared to the baseline. It also reaches 53 FPS on an RTX 4090, exceeding the 30 FPS real-time threshold. Additional evaluations on UAVDT and DroneVehicle further verify the method's robust generalization. These results indicate the effectiveness of the developed method in UAV image detection.

1. Introduction

In recent years, with the significant advancement of object detection algorithms and rapid development of UAV technologies, object detection has been widely applied in UAV platforms. Especially in the field of transportation (Kong et al., 2024; Zhang et al., 2024), due to high mobility and a wide perspective, the integration of object detection algorithms into UAVs can quickly detect the location and type of vehicles, even kinematics states across complex environments, such as highways, urban intersections or construction zones. These capacities enable UAVs to dynamically monitor large-scale traffic networks, efficiently avoid the congestion and prevent the road accidents, thereby improving the intelligent level of traffic management.

Another potentially valuable application is UAVs surveillance of railway safety operation, including rail surface defect detection (Wu et al., 2018), abnormal target intrusion monitoring (Yundong et al., 2020) and transmission line inspection (Liu et al., 2019). Because of its low cost and flexible, UAVs equipped with object detection algorithms are able to easily capture high-resolution images and accurately detect targets across any high-risk or inaccessible areas. However, different from images captured by fixed cameras, the UAV aerial images typically have a wide field of view, high proportion of small objects and complex background, as shown in Fig. 1. The above challenges make the current conventional detectors perform poorly in such scenarios, leading to a high rate of missed detections or false positives. Therefore, it is essential to develop new methods that go beyond the constraints of existing object detection frameworks.

Currently, most UAV-oriented object detection methods are built upon regular object detection frameworks, which can be divided into three main categories. The first category is two-stage detectors, including Fast R-CNN (Girshick, 2015) and Cascade R-CNN (Cai and Vasconcelos, 2018), which use a region proposal network (RPN) to generate region proposals, and then perform classification and bounding box regression based on these proposals. These models with complex network improve detection accuracy but result in slower inference speed. The second one is one-stage detectors, known as YOLO series (Terven et al., 2023), which directly predict bounding boxes and class probabilities from the input image. This design simplifies the network architecture and enables faster inference, though often at the cost of reduced detection accuracy. The last category is transformer-based

* Corresponding author.
E-mail address: fengg@sdu.edu.cn (F. Guo).

https://doi.org/10.1016/j.cviu.2026.104637
Received 27 July 2025; Received in revised form 10 December 2025; Accepted 5 January 2026
Available online 7 January 2026

Fig. 1. Examples of UAV imagery.

detectors, such as Vision Transformer (ViT) (Dosovitskiy et al., 2020) and Detection Transformer (DETR) (Carion et al., 2020), which excels at capturing global context and performs well at large objects. However, when they are applied to high-resolution UAV images, the computational cost increases significantly and the performance on small object detection often falls short compared to CNN-based models.

Given the challenges posed by UAV imagery, several classic works in recent years have explored specialized architectural designs to enhance small object detection. RSOD (Sun et al., 2022b) leveraged shallow feature maps and performed adaptive weighted fusion within the FPN to improve the detection accuracy of small objects. DCEF^2-YOLO (Shin et al., 2024) introduced deformable convolution and efficient feature fusion to better utilize internal feature information. CRL-YOLOv5 (Wang et al., 2024a) expanded the receptive field of the original model and incorporated an additional detection layer to enhance the model's ability to extract and use shallow features for small targets. CIE-YOLO (Ma et al., 2025) focused on context enhancement by integrating contextual cues to strengthen the representation of small object features. These approaches significantly improve small object detection performance by enhancing feature representation, context modeling, or multi-scale fusion. However, most of them rely heavily on convolutional architectures and still struggle to capture long-range dependencies effectively, which limits their robustness in dense distributions and complex background clutter commonly found in UAV data.

To address the limitations of purely CNN-based models, many researchers have integrated CNN and Transformer architectures to leverage their respective strengths and enhance detection performance. To the best of our knowledge, TPH-YOLOv5 (Zhu et al., 2021) is the first to integrate Transformers into a CNN-based UAV object detection framework, it enhanced the YOLOv5 architecture with a Transformer Prediction Head, significantly improving detection performance on drone imagery. Hendria et al. (2023) combined Swim Transformer backbone with a neck of DetectoRS to take advantage of both algorithms, achieving superior performance compared to either model alone. Liang et al. (2024) incorporated convolutions with the Swin Transformer to extract more local information, further improving the detection accuracy. Although these methods improved detection accuracy, they also introduced structural redundancy and quadratic computational cost, which hinder deployment on resource-constrained UAV platforms. This gap highlights the demand for architectures that preserve the complementary strengths of CNNs and long-range modeling, while remaining efficient enough for real-time UAV applications.

Recently, Mamba has emerged as a promising alternative sequence modeling architecture. It alleviates the locality constraints of CNNs while providing high-level representations comparable to Transformers, but with linear-time complexity and significantly reduced memory overhead (Gu and Dao, 2023; Zhu et al., 2024; Liu et al., 2024). Early explorations such as Mamba-YOLO (Wang et al., 2025) have demonstrated its potential for real-time detection tasks, achieving notable accuracy improvements with lower computational cost. Motivated by these observations, this paper proposes UAVDet, a CNN–Mamba hybrid detection framework tailored for UAV imagery. It is specifically designed to handle the high proportion and dense distribution of small objects and the interference of complex backgrounds, while maintaining real-time efficiency.

First, to fully leverage the strength of convolutional layers in local feature extraction, the original backbone network is redesigned by reconfiguring the original channel allocation strategy and adjusting the depth of feature extraction layers, which better captures and preserves fine-grained details beneficial for small object detection. Next, to incorporate global modeling while avoiding the quadratic complexity of Transformer-based approaches, the CSPMB module is introduced by combining the Mamba architecture and the CSPLayer block, which provides a global receptive field with only linear complexity while preserving local information, thereby enriching feature representation and improving detection performance under cluttered scenes. Finally, a novel neck architecture called TFPN is created by adding a tiny-object prediction head in place of the large-object one and optimizing the feature flow path, enhancing the perception and detection accuracy of the model to small objects while reducing the parameters. The main contributions of this paper are as follows:

- We propose UAVDet, a CNN–Mamba hybrid architecture for real-time UAV object detection, which effectively balances fine-grained local representation and efficient global context modeling.
- We revisit CNN backbone design to better preserve small-object features, providing a stronger foundation for downstream multi-scale feature fusion.
- We introduce the TFPN architecture in combination with the CSPMB module, which collaboratively enhance small-object perception by integrating local details with global context.

The rest of this paper is organized as follows: Section 2 provides a comprehensive review of recent advances in UAV-based object detection and the application of Mamba in this field. Section 3 describes the architecture and key components of the proposed UAVDet model. Section 4 presents the experimental setup, results, and comparative evaluations. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2. Related work

2.1. Object detection for UAV images

UAV object detection is a typical small object detection task, characterized by a high proportion of small targets and dense object distributions. Additionally, backgrounds captured by UAVs may encompass various terrains and structures, potentially confusing small targets and increasing the risk of false positives. To address the problem of

densely distributed small objects, UAV-YOLO (Liu et al., 2020) optimized the backbone by increasing convolution operations in the early layers to enrich spatial information, thus enhancing small object detection whilst maintaining performance under normal conditions. HR-FPN (Chen et al., 2023) upsampled the multi-scale features from backbone into high-resolution features to increase the chance to find out small-scale objects, and avoid the feature redundancy of vanilla FPN.

Among these studies, many approaches either rely heavily on handcrafted heuristics or multi-scale feature fusion, they often introduce additional computational complexity and lack adaptability to diverse UAV scenarios. More importantly, they largely overlook the suitability of the backbone architecture itself for small object detection. In many cases, backbone networks originally designed for generic object detection may not effectively preserve the fine-grained details crucial for small targets, particularly under complex backgrounds and dense distributions. These limitations motivate our work to rethink the backbone design specifically for UAV scenarios with a high proportion of small and densely distributed objects.

Object detection in UAV imagery is particularly challenging due to complex backgrounds influenced by factors such as variable viewpoints, inconsistent lighting, shadows, motion blur, and frequent occlusions. To address these issues, Lu et al. (2023) proposed using a cross-shaped window (CSWin) transformer as the backbone of the Mask R-CNN to capture long-distance feature dependencies and extract multi-level representations, its integration with the FPN enables more robust detection performance under various complex environments. Gao et al. (2024) combined YOLOv5 with the Transformer prediction head to capture long-range dependencies and model contextual information, which makes it particularly suitable for handling scale variations and complex spatial relationships commonly found in UAV aerial images. ESO-DETR (Liu et al., 2025) introduced a multiscale multihead self-attention (MMSA) mechanism into the AIFI module to strengthen the model's capacity for capturing detailed multiscale target features, it also mitigates background interference in complex scenes by combining multiscale features with channel attention.

In light of the limited computational resources on UAV platforms, recent research has focused on designing lightweight detection models capable of real-time inference and efficient deployment. CEASC (Du et al., 2023) employed sparse convolutions and an adaptive multi-layer masking scheme in their detection heads to deliver an optimal balance between accuracy and efficiency. SOD-YOLO (Xiao and Di, 2024) avoided the excessive use of group convolutions and element-wise operations by increasing network depth, reducing channel width, and stacking multi-layer progressive concatenation structures, which enables efficient deployment on a wide range of edge computing devices. RemDet (Li et al., 2025) revealed that using multiplication instead of feedforward networks enables efficient high-dimensional representation with lower information loss and reduced latency. Zhang et al. (2025a) proposed an efficient approach that adaptively adjusts feature scales and enhances contextual representation to improve small-object detection without adding model complexity. Although these methods advance efficient detector design for UAV platforms, they primarily focus on convolutional optimization or structural simplification. In contrast, our work introduces a CNN–Mamba hybrid architecture that simultaneously enhances global contextual modeling and fine-grained local representation, achieving a better balance between accuracy and efficiency under constrained computational budgets.

2.2. Mamba in object detection

The Mamba (Gu and Dao, 2023) architecture, derived from the State Space Model (SSM), has recently emerged as a powerful alternative to Transformer-based sequence models. By replacing token-level attention with selective and structured recurrence, Mamba compresses feature representations for faster and more memory efficient modeling.

Mamba model was originally proposed in the community of natural language processing to model long sequence to capture the long-term dependencies between words. Inspired by the high efficiency of Mamba model, Zhu et al. (2024) introduced Mamba into computer vision and proposed a Vision Mamba (Vim). The Vim model effectively captures local dependencies and maintains a global perspective by segmenting the input image into multiple local windows and applying a selective scanning mechanism within these windows, it outperforms CNNs and ViTs in image classification, object detection, and segmentation. In contrast, VMamba (Liu et al., 2024) proposed Cross-Scan Module (CSM) to scan feature maps in four different directions, which guarantees that every pixel efficiently incorporates information from all other pixels in different orientations. CSM module enables VMamba model to provide global receptive field with only linear complexity, which shows potential in several visual perception tasks, particularly in terms of improving performance as image resolution increases.

Motivated by the success of Mamba in vision applications, researchers have adopted it to tackle object detection tasks. As an example, Wang et al. (2024b) proposed a novel information fusion module for the YOLOv8 series models, which integrates Cross Stage Partial (CSP) connections with the Mamba structure, thereby substantially strengthening multi-scale feature fusion in the neck of the network, leading to improved detection accuracy for targets of varying sizes. YOLOv5-mamba (Wu et al., 2024) constructed a bidirectional dense feedback network by combining the mamba module with the C2f module, which enhances the transmission of contextual information and improves focus capabilities within the neck part of the architecture, leading to improved detection performance for UAV scenarios. These methods achieve accurate and efficient visual understanding with minimal computational cost.

Mamba improves object detection by modeling long-range dependencies and integrating global context, which strengthens feature representation and localization. Its robustness to background interference, combined with linear complexity, makes it suitable for real-time deployment on resource-constrained UAV platforms.

3. Proposed method

This paper proposes UAVDet, a real-time object detection model tailored for UAV aerial imagery, built upon the RTMDet-m framework and enhanced with the Mamba architecture. To better capture fine-grained features essential for small object detection, the backbone structure is redesigned by adjusting channel dimensions and the depth of CSPNeXt blocks, mitigating information loss from downsampling. The integration of the developed Cross Stage Partial Mamba (CSPMB) module into the neck further strengthens the model's ability to capture both global context and local details, enhancing detection robustness in complex scenes. Additionally, a proposed Tiny-focused Feature Pyramid Network (TFPN) is introduced to improve sensitivity to small objects by refining the feature fusion process and replacing large-object layers, enabling UAVDet to achieve accurate detection with reduced computational overhead.

The detailed network architecture of UAVDet is illustrated in Fig. 2. First, the original input image is resized to 640×640 before being fed into the backbone network for feature extraction. The backbone produces five feature layers, from which the last four are selected and passed into the neck network for multi-scale feature fusion. The neck outputs three enhanced feature maps with spatial resolutions of 160×160, 80×80, and 40×40, respectively. Finally, three dense prediction heads are applied to these feature maps of different scales to detect objects of varying sizes, determining both their presence and class. These heads correspond to tiny, small and medium object detection, respectively.

3.1. Backbone optimization for fine-grained feature preservation

One of the primary challenges in the field of UAV image object detection is the high proportion and dense distribution of small targets.

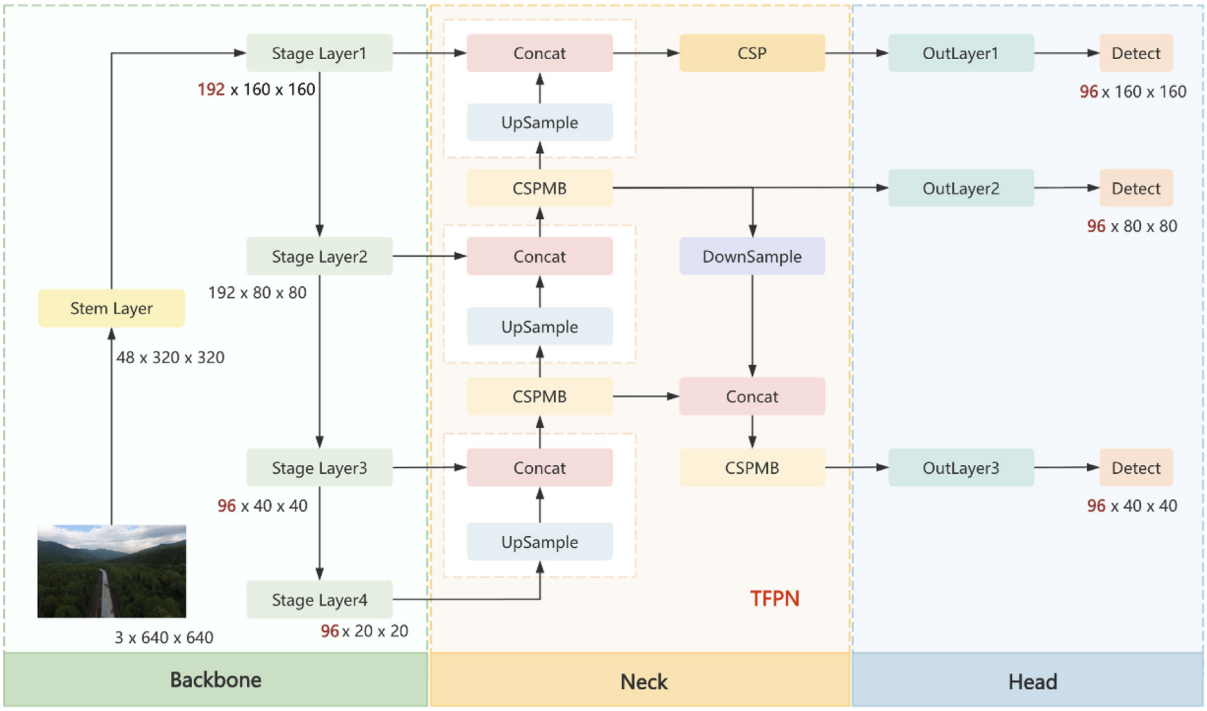


Fig. 2. Overall framework of the proposed UAVDet.

Due to the unique perspective offered by UAV aerial photography, there are often numerous small objects that only occupy a small fraction of the image pixel space but conveying crucial information. As the feature extraction network undergoes multiple downsampling, the small object feature information inevitably diminishes or even disappears. The only way to preserve information about small features is for convolutional filters in the earliest layers to encode these features and pass this information on to subsequent layers. However, in existing backbones (Zhang et al., 2025b) the number of convolutional filters in the early layers is kept to a minimum to reduce the computational burden, while progressively increasing channel width and depth in later stages for semantic abstraction.

One intuitive way to retain high-resolution features for small object detection is to intensify convolutional processing at early stages. However, this comes at the cost of substantially increased computational complexity due to the quadratic scaling of operations with feature map size. To address this issue, we redesigned the original backbone architecture with a focus on preserving fine-grained features in the early stages, while maintain a lightweight and efficient structure suitable for deployment on resource-constrained platform. In conventional backbones such as RTMDet, the number of channels increases progressively with network depth to facilitate the extraction of high-level semantic features, as illustrated in Fig. 3(a). While effective for general-purpose detection tasks, this "light-head, heavy-bottom" configuration is suboptimal for UAV imagery, where small objects demand rich texture and spatial detail early in the network. To better preserve fine-grained details beneficial for small object detection, we reverse the trend and adopt a "heavy-head, light-bottom" strategy. More convolutional channels are allocated in the early stages to better preserve high-resolution information necessary for small object representation, while fewer channels are used in deeper layers to reduce computational cost, as depicted in Fig. 3(b). This adjustment allows for better fine-grained feature capture without significantly increasing the model's complexity, making it more suitable for real-time applications on UAVs.

In addition, according to Fig. 3(a), the original backbone uses 2, 4, 4, and 2 CSP modules across stages 1 to 4, following an "ascending-then-descending" pattern designed to balance shallow detail retention and deep semantic extraction while keeping computational load manageable. However, this strategy may lead to the loss of critical low-level information, particularly in UAV scenarios where small object detection relies on detailed texture and localization cues. In our design, as shown in Fig. 3(b), we reallocate the CSP modules to emphasize early-stage processing. More CSP blocks are used in the first and second stages, while the number of blocks is progressively reduced in the third and fourth stages. This reconfiguration ensures sufficient feature extraction for small objects at early layers, preserves essential semantic abstraction in deeper stages, and maintains computational efficiency. The detailed configuration of the proposed backbone network is provided in Table 1. These modifications ensure that high-resolution information necessary for distinguishing small objects is effectively extracted and preserved, laying a solid foundation for subsequent processing stages in the network.

3.2. CSPMB module

In CNN-based object detection networks, shallow feature maps contain rich fine-grained information, which is beneficial for detecting densely distributed small objects. In contrast, deep feature maps contain abundant semantic information and are more suitable for detecting prominent large objects. FPN is a common feature fusion method that integrates fine-grained information from shallow feature maps and semantic information from deep feature maps by constructing bottom-up and lateral connections, generating feature maps containing multiscale feature information. This approach allows objects to be represented with both detailed boundaries and rich semantic context, which helps distinguish visually similar but semantically different regions and enhances boundary awareness. As a result, it alleviates the impact of background interference.

Considering the challenges posed by cluttered backgrounds in UAV image object detection tasks, FPN is widely adopted to help the detector better isolate objects from noisy surroundings, especially for small or partially occluded targets. However, due to the limitations of fixed receptive fields, these methods (Zhang et al., 2025c) struggle to capture rich global feature information and fail to fully exploit the

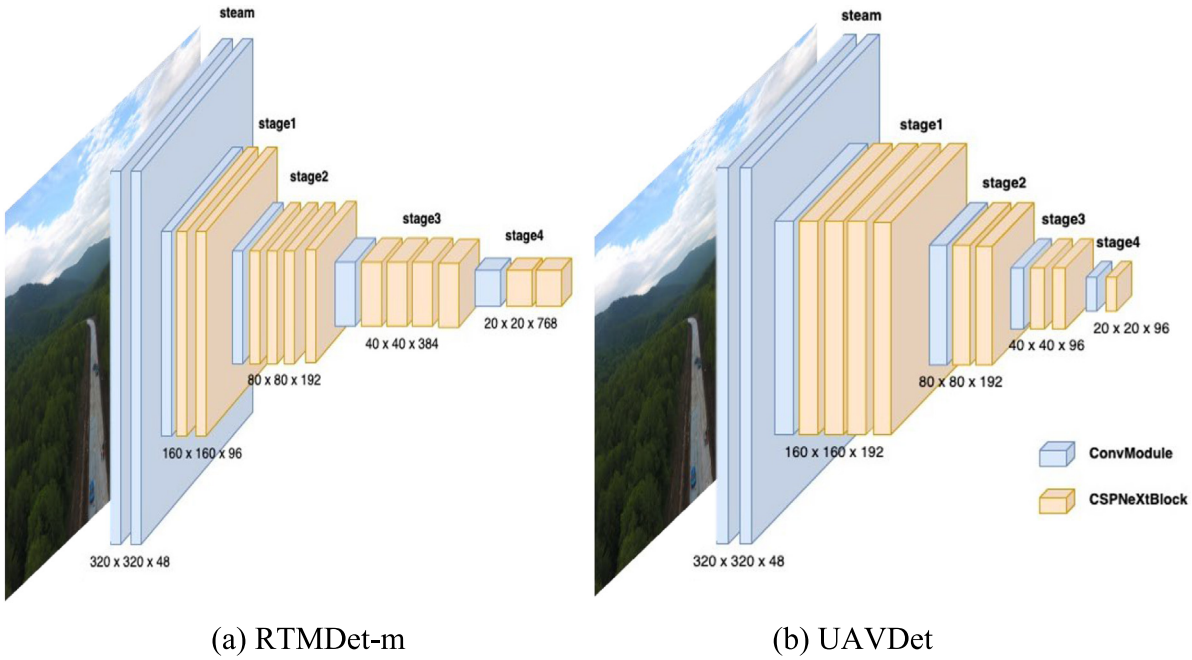


Fig. 3. Illustration of different backbone architecture.

Table 1
The detailed configuration of the proposed backbone network.

Models	Stage 1	Stage 2	Stage 3	Stage 4	Params	FLOPs
RTMDet-m	48	96	192	384	24.71M	39.27G
	96	192	384	768		
	2	4	4	2		
Ours	48	192	192	96	23.96M	42.68G
	192	192	96	96		
	4	2	2	1		

interrelationships between different feature maps. This issue becomes particularly pronounced in UAV images with expansive backgrounds, where insufficient global context modeling makes it difficult to suppress background noise, often resulting in frequent false positives and missed detections.

Although studies (Hendria et al., 2023; Liang et al., 2024; Zhu et al., 2021) have introduced Transformer architectures to capture global contextual information and enhance model robustness in complex environments, the quadratic computational complexity of Transformers inevitably leads to significant computational overhead. This makes them difficult to deploy on resource-constrained UAV platforms, limiting their practicality in real-world applications. To mitigate these limitations, we explore a more efficient global modeling mechanism based on Mamba and design a novel CSPMB block.

The key idea behind CSPMB is to combine the partial gradient flow of CSP with the linear-state sequence modeling of Mamba, allowing the network to preserve lightweight local feature extraction while efficiently capturing long-range dependencies. Specifically, the CSP structure splits feature maps into two parallel paths, one retaining original spatial details for stable gradient propagation, and the other employing the Mamba module for global context modeling. The outputs are then fused to achieve complementary representation learning. This design not only enables effective interaction between local and global semantics but also maintains low computational overhead, achieving a favorable trade-off between accuracy, efficiency, and deployability under complex UAV scenes.

The structure of CSPMB is shown in Fig. 4, it consists of two main blocks in parallel, a VSSBlock and a CSPNeXtBlock. The feature extraction process of CSPMB is described as follows. First, the input feature map with a size of $C_{in} \times H \times W$ is divided into two branches, in each branch, a 1×1 convolution operation is performed to change the channel dimension, resulting in a feature map size of $C_{out}/2 \times H \times W$. Then, one branch passes through the VSSBlock to capture global information, maintaining the same dimensions and forming a residual connection with the original input. The other branch passes through a CSPNeXtBlock to extract local features, also preserving the original dimensions and forming a residual connection. Finally, the outputs of the two branches are concatenated and passed through another 1×1 convolution for feature fusion, producing an output feature map of size $C_{out} \times H \times W$.

As demonstrated on the left side of Fig. 4, LS Block, SS2D and RG Block are the three main steps of the VSSBlock algorithm. Given the input feature $Z^{l-3} \in R^{C \times H \times W}$, it first goes through a 1×1 convolution layer to refine local channel-wise information, while maintain the spatial resolution. Then, the resulting feature Z^{l-2} is passed through the LS Block, which enhances local spatial perception. The LS Block consists of a depthwise separable convolution followed by batch normalization and two standard convolutions with an activation function, forming a residual structure. This stage outputs an intermediate feature X^l, which retains fine-grained spatial detail. This process can be represented in Eqs. (1) and (2).

$$Z^{l-2} = Conv(Z^{l-3}) \tag{1}$$

$$X^l = LSBlock(Z^{l-2}) \tag{2}$$

Next, X^l is normalized via a LayerNorm operation and then is fed into the SS2D module, a state space modeling block designed to

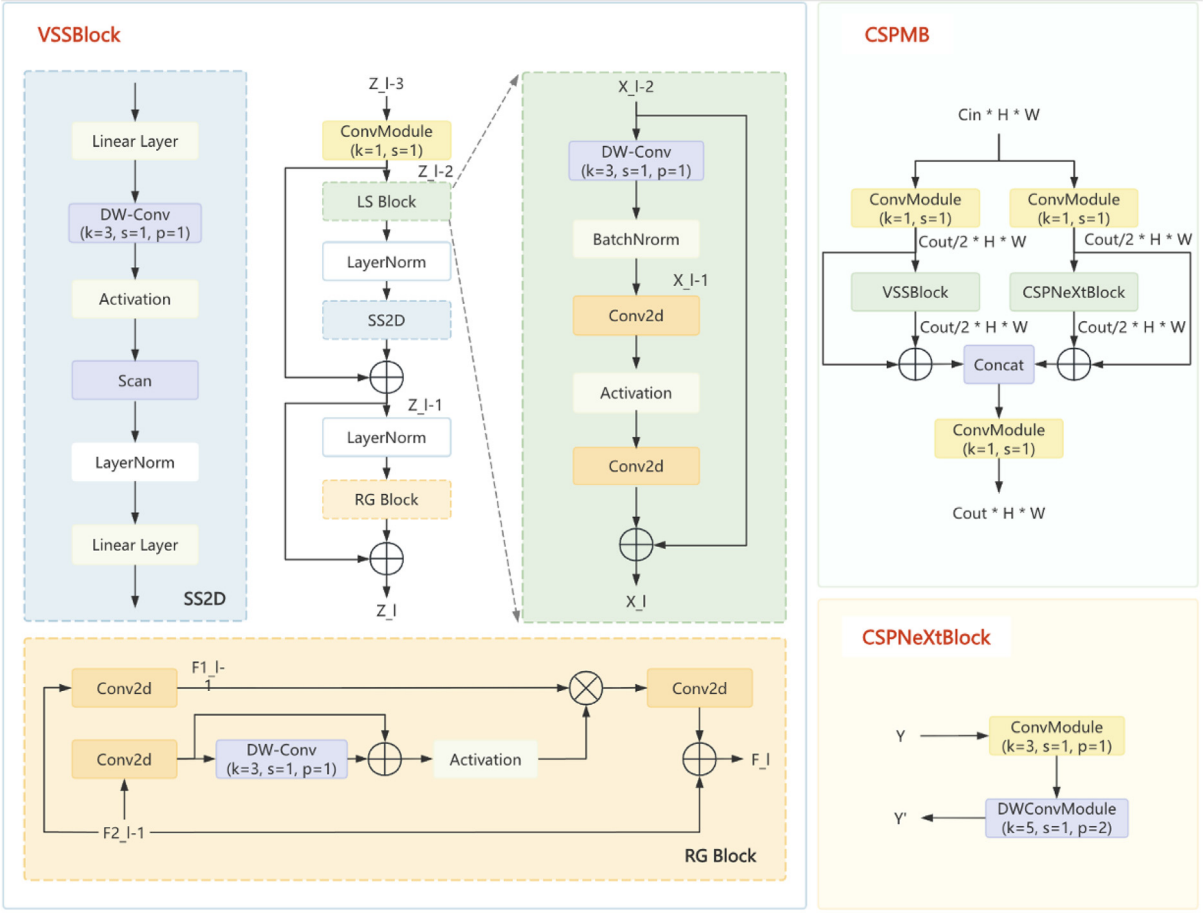


Fig. 4. The structure of CSPMB.

capture long-range dependencies. The feature is first normalized and linearly transformed, then passed through a 2D state-space propagation mechanism consisting of depthwise convolution, activation, scan operation, and a final linear projection. The scan expansion operation partitions the input image into a series of directional subimages, each representing a specific scanning perspective. Observed from the diagonal viewpoint, the expansion operation proceeds along four symmetric directions: top-down, bottom-up, left–right, and right-left, respectively.

This direction decomposition not only ensures comprehensive spatial coverage of the input image, but also provides a rich multi-dimensional context for subsequent feature extraction, enabling the model to capture diverse patterns more effectively. The corresponding scan merge operation takes the directionally scanned sequences as inputs and integrates them into a final 2D feature map. The SS2D module facilitates the transformation of local directional features into globally aggregated representations, thereby enhancing the model's ability to reason over spatial dependencies. The above process can be described as follows.

$$Z^{l-1} = Z^{l-2} + SS2D\left(LN\left(X^{l}\right)\right) \tag{3}$$

Finally, the output of SS2D is passed through a LayerNorm layer and input into the RG Block for residual-guided fusion. The RG Block takes two parallel 2D convolution layer to produce two intermediate features, denoted as F_1^{l-1} and F_2^{l-1}. Then, F_2^{l-1} is passed through a depthwise convolution with a kernel size of 3, stride 1, and padding 1. The resulting feature is element-wise added to the original input F_2^{l-1}, forming a residual connection. Next, the sum is passed through a nonlinear activation function to generate an attention map, which is then element-wise multiple with F_1^{l-1}, allowing the network to recalibrate the important of different spatial locations in F_1^{l-1}. The refined result is further processed by another 2D convolution layer, and finally added to the initial input via a residual connection to produce the output feature map F^l. Eq. (4) represents the specific process.

$$Z^{l} = Z^{l-1} + RGBlock\left(LN\left(Z^{l-1}\right)\right) \tag{4}$$

The CSPNeXtBlock is the original module used in the baseline, and its structure is shown in the lower right part of Fig. 4. The processing flow is as follows: the input feature map of size $C_{in} \times H \times W$ first passes through a standard convolution module with a 3×3 kernel, stride 1, and padding 1. This operation captures local spatial patterns while preserving the spatial resolution of the feature map. The output is then processed by a depthwise separable convolution module with a larger 5×5 kernel, stride 1, and padding 2. This helps expand the receptive field and enhances the ability to model context with minimal increase in computational cost. The final output retains the same dimensions as the input $C_{in} \times H \times W$, enriched with both local detail and moderately extended context information. This process can be described as Eq. (5).

$$Y' = DWConv(Conv(Y)) \tag{5}$$

Overall, the CSPMB module integrates Mamba's global modeling capability with linear complexity. As a result, it effectively captures long-range dependencies while preserving local spatial detail. This dual-perspective feature modeling allows the network to better distinguish foreground targets from background noise, particularly in UAV scenarios with cluttered scenes, occlusions, or scale variations.

3.3. TFPN

The original neck structure adopts the PAFPN (Liu et al., 2018) architecture, as illustrated in the upper part of Fig. 5, which introduces a

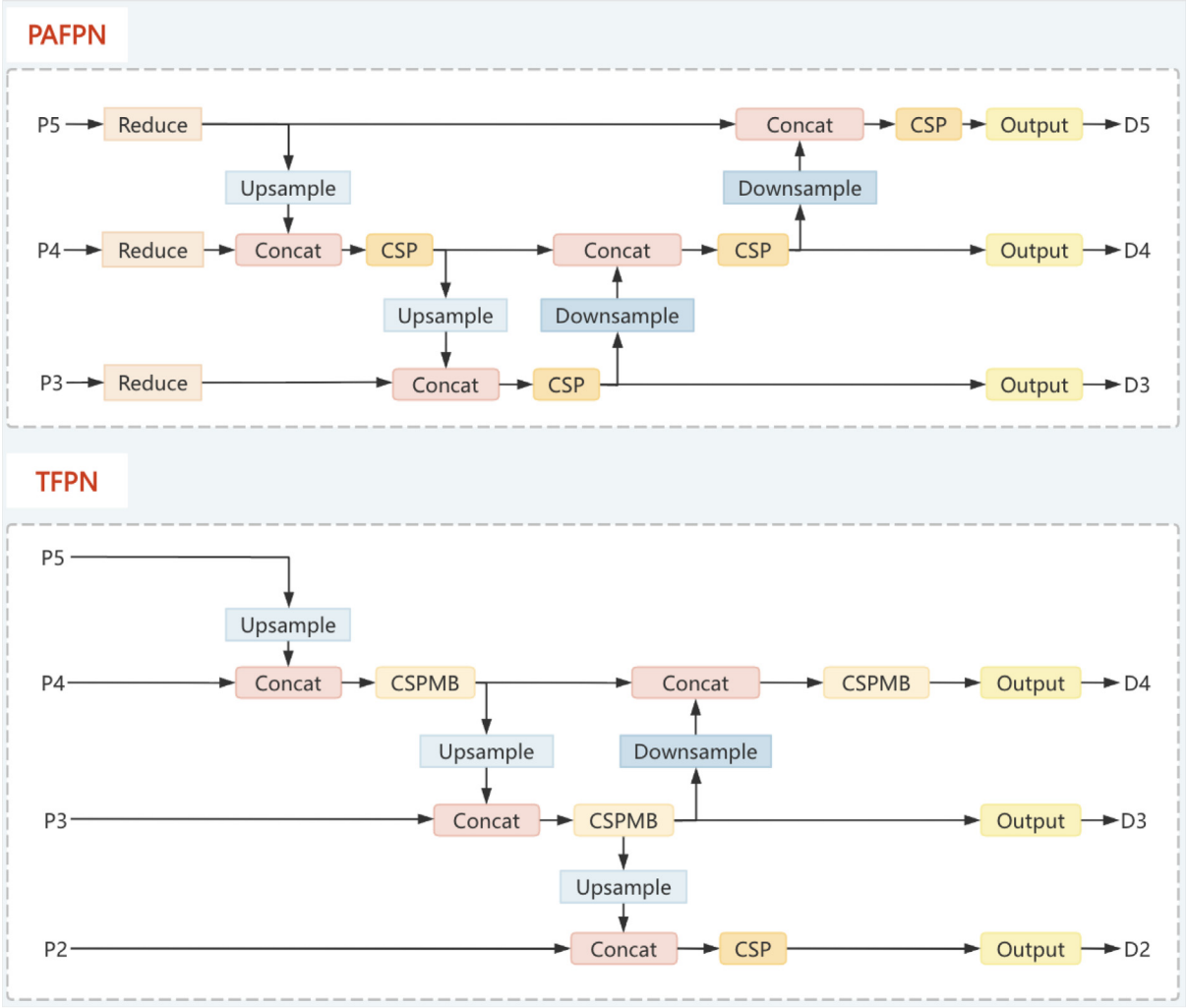


Fig. 5. Comparison between PAFPN structure and TFPN structure.

bottom-up path to enhance multi-scale feature fusion and improves the localization accuracy of objects. Although PAFPN does provide some benefits for small object detection by propagating low-level features to enhance entire feature hierarchy, its improvements are limited in UAV scenarios where small objects are densely packed and often blurred or occluded. On the one hand, its multi-path aggregation mechanism may result in excessive background information being propagated, leading to false positives, particularly in cluttered scenes with rich background textures. On the other hand, its dual-path information flow inevitably incurs a large amount of redundant computation, leading to increased model complexity and reduced inference efficiency.

In the MS COCO dataset (Lin et al., 2014), objects are categorized based on size: small objects are defined as those with dimensions smaller than 32 × 32 pixels, medium objects range from 32 × 32 to 96 × 96 pixels, and large objects exceed 96 × 96 pixels. Although the original RTMDet model includes three detection heads, with one operating on a high-resolution 80 × 80 feature map for small object detection, this is still insufficient for UAV-based images, where targets are often extremely small and densely distributed. Many small objects may still be missed even at this resolution. We argue that the existing feature fusion design (Zhang et al., 2025d) lacks sufficient emphasis on fine-grained spatial details critical to small object representation. As a result, when feature maps are downsampled, the already limited information of small objects is further degraded, making them difficult to extract and leading to suboptimal detection performance.

To address these issues, we propose an improved feature fusion architecture named TFPN, as shown in the lower part of Fig. 5, which aims to enhance the model's sensitivity to small objects while improving information flow efficiency and reducing redundancy. TFPN retains the ability to model multi-scale features but introduces higher-resolution shallow features and more targeted fusion pathways, enabling more effective preservation and enhancement of fine-grained details. Furthermore, CSPMB modules are incorporated in the deep feature fusion stages to capture global relationships across different feature scales, effectively mitigating the interference caused by the propagation of complex background information. This design significantly boosts detection performance for small objects, particularly in complex UAV-based scenarios.

Firstly, we introduce an additional detection head, D2, dedicated to detecting smaller objects by leveraging a higher-resolution (160 × 160) feature map, as depicted in Fig. 6. Specifically, we upsample the P3 feature map (80 × 80) to 160 × 160, and fuse it with the backbone's P2 feature map of the same resolution. The fused feature is then refined using a CSP block to enhance representation capacity, and subsequently fed into the detection head for classification and localization, alongside other multi-scale prediction branches. The spatial resolution of feature maps in the TPFN can be formulated as:

$$P3'_{160\times160} = Upsample\left(P3_{80\times80}\right) \tag{6}$$

$$D2_{160\times160} = CSP(Concat(P3'_{160\times160}, P2_{160\times160})) \tag{7}$$

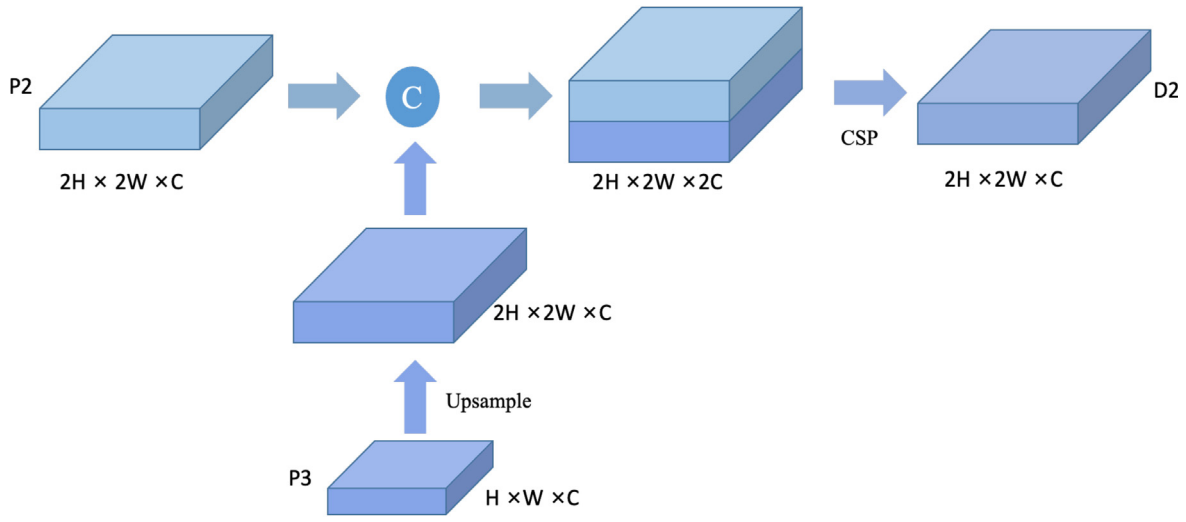


Fig. 6. The structure of D2.

It is worth noting that after introducing the D2 detection head, we do not follow the conventional approach (Sun et al., 2022b; Tang et al., 2024; Zhu et al., 2021; Zhang et al., 2025e). Instead, we treat it as a terminal output and exclude it from further top-down propagation, as illustrated in the TFPN structure in Fig. 5. Subsequent downsampling and feature fusion operations begin from the P3 layer. We suggest that small object detection is best accomplished at shallow layers where fine spatial details are preserved. Further propagation may introduce noise to other scales and unnecessarily increase computational cost, potentially degrading both accuracy and efficiency.

Furthermore, our analysis of the VisDrone2019 (Du et al., 2019) dataset indicates that extremely small objects account for approximately 60% of all annotated targets, whereas large objects represent only about 5%, highlighting a strong imbalance in object scale distribution. This dataset characteristic motivates our design to prioritize small and tiny object detection, as large object instances are relatively scarce in this UAV scenario. To improve efficiency, as illustrated in Fig. 5, we remove the large-object detection head D5. In other words, the P4 feature map is no longer downsampled and fused with P5 in the bottom-up pathway, thereby reducing the model's parameters and minimizing unnecessary computations.

Secondly, to better preserve informative features for small object detection, we remove the channel reduction layers previously applied before multi-scale fusion in the top-down pathway. We consider that reducing channels at this stage may lead to information loss, which is especially detrimental to small object detection. Given that small objects inherently contain limited discriminative features, preserving as much detail as possible is essential for accurate localization and recognition.

Last, in our TFPN design, we replace the standard CSP modules in the P4 and P3 layers with the proposed CSPMB modules to enhance the network's capacity for cross-scale semantic modeling. The motivation behind this substitution lies in the characteristics of these layers: P4 and P3 are middle-level features that contain both spatial detail and semantic abstraction, making them ideal for integrating global contextual information to suppress background noise and reinforce object representations, especially for ambiguous or occluded small objects.

However, we deliberately avoid introducing CSPMB into the P2 layer. This is because P2 is a very shallow feature map, primarily responsible for preserving low-level spatial details critical to detecting extremely small targets. Applying CSPMB, which involves more complex global modeling and larger receptive fields, could blur fine-grained local features and potentially introduce unnecessary semantic redundancy at this level. Moreover, adding such heavy modules to early layers would significantly increase computational cost without proportional performance gain, especially when P2 is already optimized for resolution-sensitive tasks. Therefore, the use of CSPMB is selectively applied to layers where both semantic enhancement and context reasoning are most beneficial, while preserving the lightweight and detail-preserving nature of shallow layers like P2.

In summary, compared to conventional neck designs (Liu et al., 2018; Tan et al., 2020; Zhang et al., 2025f), TFPN enables more efficient small object feature extraction and promotes a semantically aligned yet lightweight fusion strategy.

4. Experiments and analysis

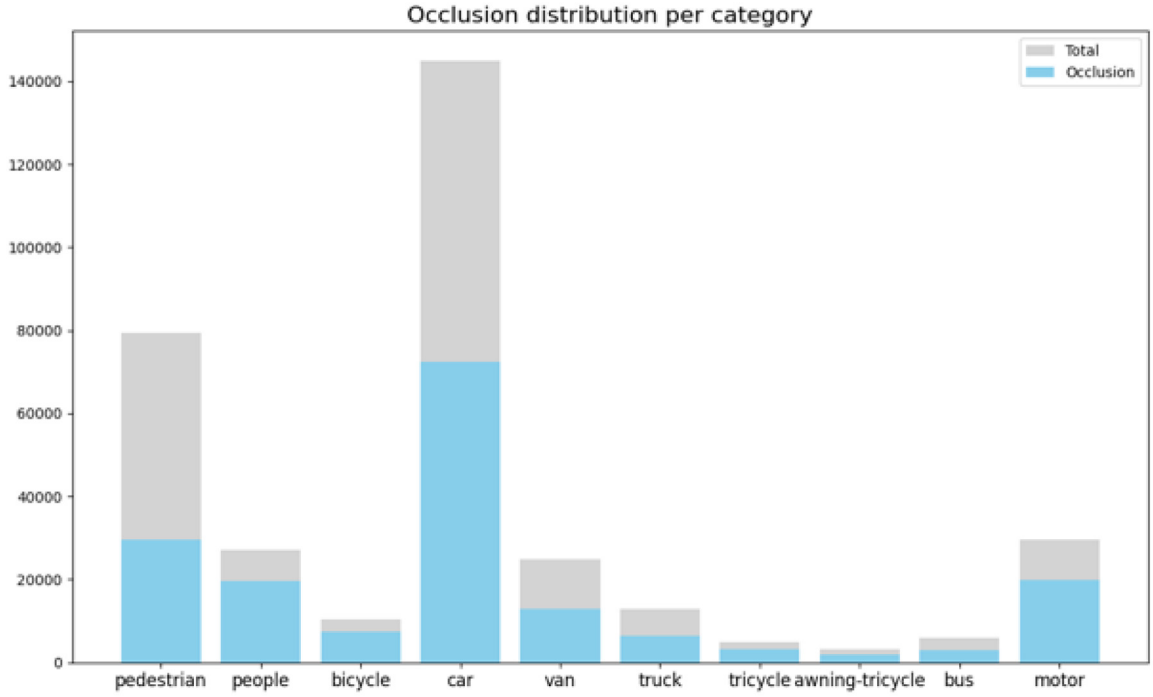
4.1. Datasets

To evaluate our method, we conduct extensive UAV object detection experiments on three datasets: VisDrone (Du et al., 2019), UAVDT (Du et al., 2018) and DroneVehicle (Sun et al., 2022a). The VisDrone2019-DET dataset is a widely used benchmark for object detection in UAV aerial imagery. It contains 8629 static images captured by UAVs across various regions, scenes, and altitudes, covering a wide range of spatial resolutions. The dataset poses significant challenges due to diverse object scales, a high proportion of very small objects, and complex backgrounds that often lead to interference and reduced detection accuracy. Fig. 7 displays that a significant portion of the objects in the dataset are occluded, and small objects dominate the overall target distribution, which is highly consistent with the research background and the core problem addressed in this paper.

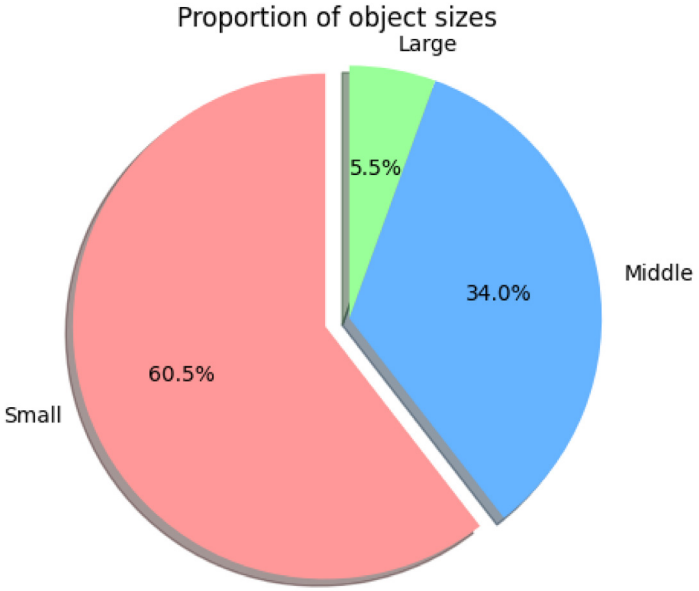
The VisDrone dataset is divided into training, validation, and testing subsets, comprising 6471 images for training, 548 for validation, and 1610 for testing. Each image is annotated with bounding boxes corresponding to ten predefined object categories: pedestrian, person, car, van, bus, truck, motorbike, bicycle, awning-tricycle, and tricycle. In our experiments, we use the training set for model training and the testing set for evaluation.

UAVDT is a large-scale and challenging dataset designed for object detection from a UAV perspective. It was collected by drones in diverse real-world scenarios, including urban roads, highways, and rural areas, covering targets observed under different altitudes, viewing angles, and lighting conditions. The dataset focuses on three object categories: car, truck, and bus, and contains 23,258 training images and 15,069 validation images.

DroneVehicle is a wide-coverage UAV-based RGB–Infrared vehicle detection dataset that contains 28,439 paired RGB–Infrared images, covering various scenes such as urban roads, residential areas, parking



(a) Occlusion distribution of targets



(b) Distribution of object sizes

Fig. 7. Quantitative analysis of labels in the training dataset.

lots, and other environments from daytime to nighttime. Each image is annotated with oriented bounding boxes for five categories: car, truck, bus, van, and freight car. In our experiments, only the RGB images are used, including 17,990 images for training, 1469 images for validation, and 8980 images for testing.

4.2. Evaluation metrics

We adopt the standard COCO evaluation metrics to assess and compare the performance of different methods, primarily focusing on Average Precision (AP). AP measures the mean precision across all categories and is widely used as a comprehensive indicator of detection accuracy, with higher values indicating better performance. It is computed as the average of AP scores over 10 Intersection over Union (IoU) thresholds ranging from 0.5 to 0.95 in increments of 0.05. The calculation method of AP is as shown in Eq. (8), (9) and (10).

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$AP = \frac{\sum_{k-1}^{K} \int_{0}^{1} P(R) dR}{K} \tag{10}$$

where k denotes the category index, K is the total number of categories, P and R represent precision and recall, respectively. P and R need to be

Table 2
Training parameters.

Hyperparameter	Value
Image size	640 × 640
Epoch	200
Batch size	16
Optimizer	AdamW
Initial learning rate	1e−3
Minimum learning rate	2e−4
Weight decay	0.05
Learning rate scheduler	Cosine Annealing

calculated using True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

We also report AP_{50} and AP_{75}, which represent AP scores at fixed IoU thresholds of 0.5 and 0.75, respectively, for all categories. In addition, to evaluate the scale-awareness of the models, we include AP scores for objects of different sizes: AP_S for small objects, AP_M for medium objects, and AP_L for large objects. These metrics are particularly important in UAV imagery, where small object detection is a primary challenge. All images are resized to 640 × 640 during training and testing to ensure consistency with COCO definitions and comparability across methods. However, we note that UAV datasets contain a large proportion of extremely small targets, which are not fully captured by the standard COCO small object definition. Therefore, in our ablation studies, we introduce an additional metric, AP_T (Average Precision for Tiny objects, $< 16 \times 16$), to more precisely evaluate the model's ability to detect these micro-scale targets.

To further assess the model's efficiency, we also report computational complexity indicators, including the number of parameters (Params) and Floating Point Operations (FLOPs). FLOPs reflect the total number of floating-point operations performed during a single inference pass, measured in GFLOPs, and are calculated based on an input resolution of 640 × 640. A lower FLOPs value generally indicates higher inference efficiency, making the model more suitable for resource-constrained applications. These metrics help evaluate the trade-off between model accuracy and computational cost. When it comes to some real-time application scenarios, Frames Processed per Second (FPS) is used to measure the inference speed of model, indicating how many frames the model can process per second. The higher the FPS, the faster the model's inference speed.

4.3. Implementation details

To ensure the reproducibility of experimental results, all experiments were conducted on the same high-performance computing environment. The system ran on Ubuntu 18.04 with an NVIDIA RTX 4090 GPU (24 GB memory). Our method was implemented using PyTorch framework and the MMDetection (Chen et al., 2019) toolbox, with CUDA 11.8 and Python 3.8.

During training, images were resized to 640 × 640, and a batch size of 16 was used. The optimizer was AdamW with an initial learning rate of 1e-3 and a weight decay of 0.05. The learning rate schedule consisted of a linear warm-up phase for the first 1000 iterations, followed by Cosine Annealing during the final 100 epochs, gradually reducing the learning rate to a minimum of 2e-4. All other hyperparameters followed the default settings of the RTMDet (Lyu et al., 2022) architecture. The training parameters are summarized in Table 2.

The model was trained for 200 epochs on the VisDrone2019-DET dataset, using a phased data augmentation strategy. In the last 20 epochs, aggressive augmentations such as Mosaic and MixUp were disabled, retaining only basic techniques including Resize, RandomCrop, RandomFlip, HSV adjustment, and Padding to stabilize convergence. For inference, the input resolution remained consistent at 640 × 640, the confidence threshold was set to 0.3, and Non-Maximum Suppression (NMS) was applied with an IoU threshold of 0.5.

4.4. Comparison with SOTA

As illustrated in Table 3, our proposed method achieves an excellent trade-off between detection accuracy and computational efficiency on the VisDrone2019-DET dataset. Specifically, it outperforms all ten state-of-the-art (SOTA) models in overall detection accuracy, achieving the highest AP score of 26.8. In terms of small object detection, our model also attains the best AP_S score of 15.3, demonstrating its strong capability in handling densely distributed and small-scale targets that are common in UAV imagery. Compared to the baseline model RTMDet-m, our method improves AP by 4.5%. While there is a slight increase in computational cost, the number of parameters is reduced by approximately 85%, significantly improving model efficiency. Notably, methods such as ViTDet (Li et al., 2022) and RT-DETR (Zhao et al., 2024) offer competitive AP scores, but incur substantially higher computational complexity, often exceeding 100 GFLOPs, which limits their applicability in resource-constrained UAV platforms. In contrast, our method achieves similar or even better accuracy with only a fraction of the computational load.

Furthermore, when compared to other UAV-specific detectors like REMDet-m (Li et al., 2025) and TPH-YOLOv5m (Zhu et al., 2021), our approach demonstrates superior performance in detecting small objects, as evidenced by its higher AP_S, reinforcing its suitability for UAV scenarios where small targets predominate. Architecturally, compared to YOLO models based on either CNN or Mamba alone, our hybrid model yields notable improvements in detection accuracy (AP and AP_S), along with a reduction in parameters and FLOPs. These results validate the effectiveness of our design in combining the advantages of both structures for superior performance. Although the proposed model lags behind the YOLO series in terms of real-time speed, it achieves a comparable frame rate to the baseline (53 FPS vs. 59 FPS), indicating its potential for real-time deployment in UAV-based object detection applications.

Overall, these results validate the effectiveness of our proposed architecture in delivering high-accuracy detection under tight resource constraints, making it a promising solution for practical UAV aerial visual perception systems.

To further evaluate the generalization capability of our method, we conducted cross-dataset experiments by training the model on the UAVDT and DroneVehicle datasets, respectively. As shown in Table 4, UAVDet achieves 20.9% AP on UAVDT and 45.5% AP on DroneVehicle, consistently outperforming all compared methods. Moreover, UAVDet yields at least a 1.0% AP_S improvement over the baseline across both datasets, demonstrating enhanced sensitivity to small objects. In addition, although the D5 head was removed to improve efficiency, the model's ability to detect large scale targets remains stable. Specifically, UAVDet attains 39.8% AP_L on UAVDT and 39.5% AP_L on DroneVehicle, showing only a negligible drop ($\leq$1.5%) compared with the best performing large object detectors. These results demonstrate that the proposed design modules substantially enhance small object sensitivity without compromising semantic representation or generalization performance. Overall, UAVDet maintains robust cross-domain generalization and effectively adapts to diverse UAV imagery.

4.5. Ablation studies

To evaluate the contribution of each proposed component, we conduct a comprehensive ablation study on the VisDrone2019-DET dataset, as shown in Table 5. Starting from the baseline model, we progressively incorporate the optimized backbone, CSPMB module, and TFPN, analyzing their individual and combined impact on detection performance and model efficiency. Each component leads to varying degrees of improvement in detection accuracy. Unless otherwise stated, all ablation variants of TFPN and CSPMB are evaluated independently on the baseline model without mutual inclusion, to ensure a fair assessment of their respective contributions.

Table 3
Evaluation results on the VisDrone2019 test set.

Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params(M)	GFLOPs	FPS
ViTDet (Li et al., 2022)	24.1	37.6	26.8	5.5	33.9	53.8	110	409	28
DINO (Zhang et al., 2022)	19.6	37.3	18.4	11.9	28.0	39.9	47.56	274	34
RT-DETR (Zhao et al., 2024)	24.5	42.3	24.8	13.5	34.7	51.3	42	136	68
YOLOv8m	19.3	33.9	19.3	9.4	29.7	39.9	25.9	79.1	106
YOLO11m	20.6	35.7	21.1	10.7	31.6	42.9	20.1	68.2	133
Mamba YOLO-M (Wang et al., 2025)	18.3	32.2	18.5	9.0	28.1	37.9	21.8	49.6	141
CEASC (Du et al., 2023)	17.8	32.3	17.9	7.2	28.5	36.9	43.28	105.9	51
TPH-YOLOv5m (Zhu et al., 2021)	18.1	31.9	18.5	8.6	25.2	36.1	27.4	68.2	127
REMDet-m (Li et al., 2025)	23.3	39.2	24.1	11.6	34.7	48.7	23.3	34.4	188
RTMDet-m (Lyu et al., 2022)	22.3	37.5	23.2	10.3	33.6	49.5	24.71	39.27	59
UAVDet(ours)	**26.8**	**44.8**	**27.8**	**15.3**	**38.3**	**54.9**	3.74	46.63	53

Table 4
Evaluation results on the UAVDT val and DroneVehicle test sets.

Methods	UAVDT						DroneVehicle					
	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RT-DETR (Zhao et al., 2024)	18.2	29.8	20.0	13.1	29.2	27.4	44.4	64.3	51.8	17.8	45.1	37.9
YOLO11m	17.5	29.6	19.0	12.3	28.2	29.1	43.7	65.3	51.1	17.1	43.7	37.5
Mamba YOLO-M (Wang et al., 2025)	16.9	28.6	18.3	12.4	26.3	28.1	43.9	65.9	51.4	16.5	44.5	37.5
CEASC (Du et al., 2023)	17.1	30.9	17.8	–	–	–	37.7	62.7	40.5	17.2	38.3	29.2
TPH-YOLOv5m (Zhu et al., 2021)	15.4	30.4	13.4	9.9	25.6	31.4	44.6	66.1	52.9	17.0	44.8	40.9
REMDet-m (Li et al., 2025)	20.0	32.6	21.7	14.0	30.5	34.2	44.2	64.7	51.9	17.1	43.9	**41.0**
RTMDet-m (Lyu et al., 2022)	20.4	33.7	22.4	13.5	**32.4**	36.8	44.7	66.1	52.1	17.2	44.6	40.0
UAVDet(ours)	**20.9**	**35.0**	**22.7**	**14.5**	32.3	**39.8**	**45.5**	**67.3**	**53.4**	**18.8**	**45.7**	39.5

Table 5
Ablation study of the proposed modules.

Optimized Backbone	CSPMB	TFPN	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params (M)	GFLOPs
Baseline			22.3	37.5	23.2	10.3	33.6	49.5	24.71	39.27
√			25.6	43.5	26.7	14.3	36.9	49.2	23.96	42.68
	√		22.9	38.3	24.1	10.8	34.3	50.9	29.23	43.80
		√	24.7	42.0	25.5	13.5	35.3	51.4	16.54	39.03
√		√	26.1	44.0	27.3	15.2	37.5	49.8	3.40	45.18
	√	√	25.2	42.6	26.1	14.0	36.4	49.5	18.26	42.63
√	√	√	**26.8**	**44.8**	**27.8**	**15.3**	**38.3**	**54.9**	3.74	46.63

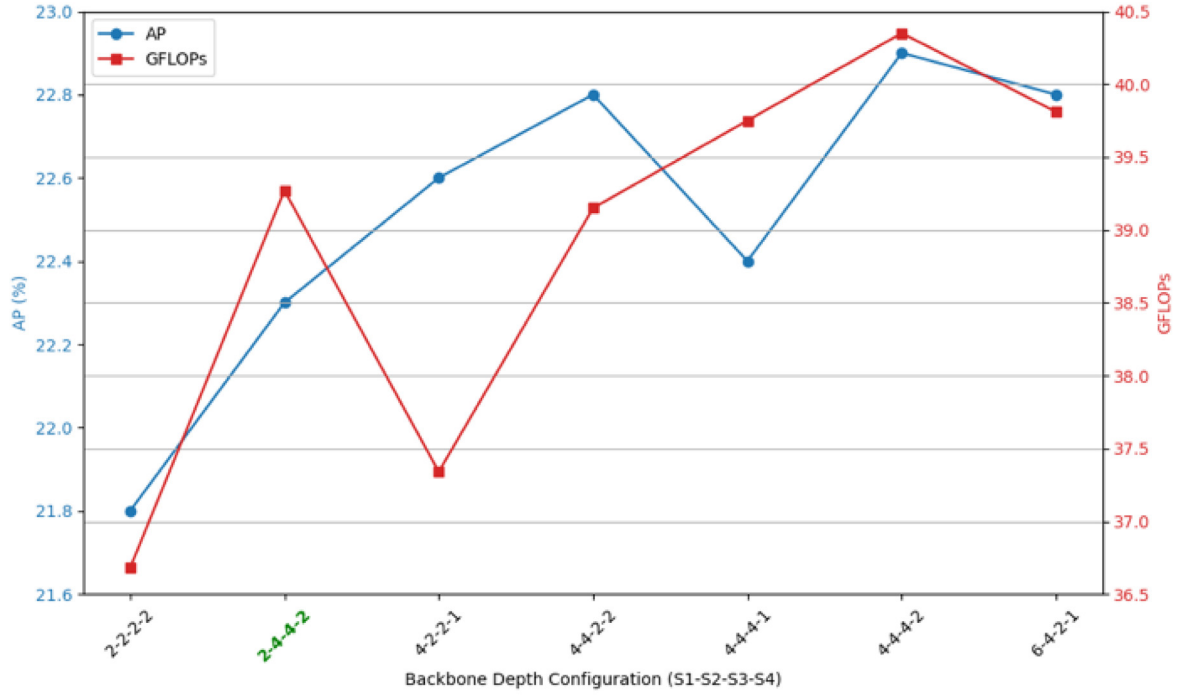
Among them, introducing optimized backbone increases AP by 3.3%, and $\mathbf{AP}_S$ by 4.0%. The inclusion of the CSPMB module alone yields a modest improvement in AP by 0.6%, while TPFN boost AP and AP_S by 2.4% and 3.2%, respectively. Notably, building on the TFPN component, incorporating either the optimized backbone or the CSPMB module further enhances the model's detection performance. When all three components are integrated, the model achieves the best performance, with an AP improvement of 4.5% and an $\mathbf{AP}_S$ improvement of 5.0% over the baseline. While the full configuration introduces an additional 7.36 GFLOPs, it reduces 20.97M parameters, representing an 84.9% decrease. These results validate that each module contributes both independently and synergistically, and that the complete design offers an optimal balance between accuracy and efficiency for UAV-based object detection.

We explore various configurations for the number of repetitions of the CSPNeXtBlock in the backbone. As shown in Fig. 8(a), different backbone depths affect both detection performance and computer complexity. Increasing the depth in the early stages (stage 1 and 2) leads to improved detection accuracy, even if the depth in the later stages is reduced. For example, the AP of [4-2-2-1] surpasses that of the baseline [2-4-4-2]. In contrast, increasing the depth in the later stages yields only marginal performance gains while significantly increasing computational cost. This is evidence when comparing [4-4-4-2] and [4-4-2-2]: the AP improvement is minimal, while the GFLOPs rise sharply. These results suggest that enhancing low-level feature extraction is particularly beneficial for small object detection, whereas adding depths introduces computational overhead without commensurate accuracy gains.

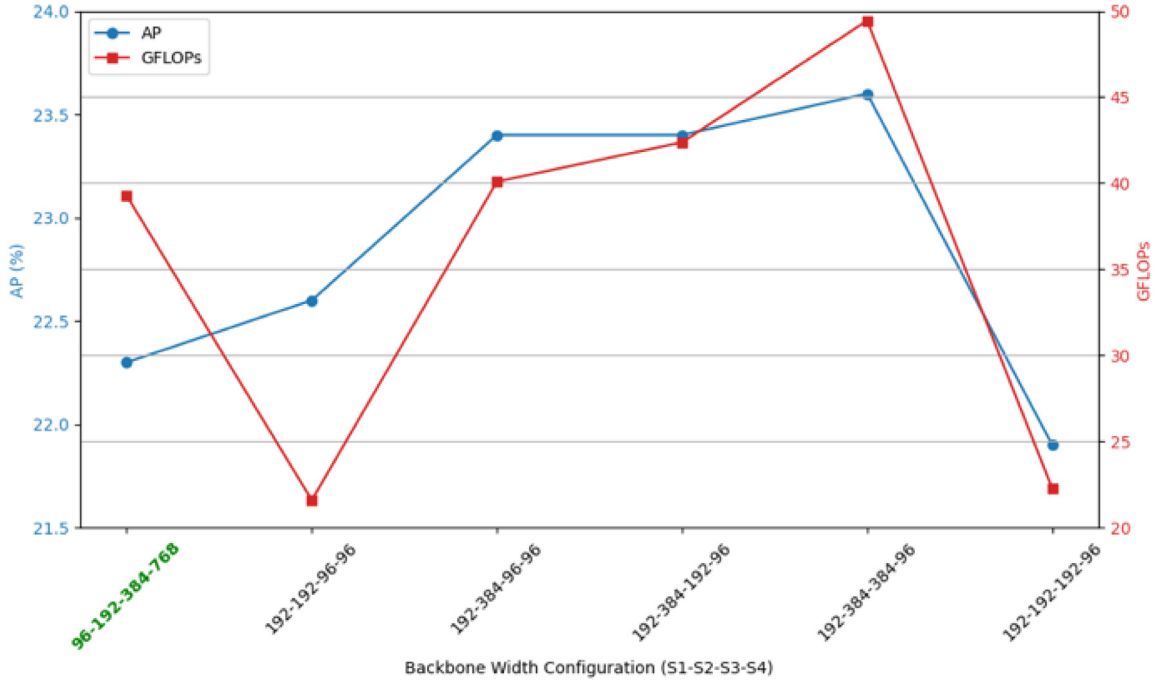
Furthermore, we investigate the impact of varying the output channel dimension at each stage, as illustrated in Fig. 8(b). Increasing channel width in the early stages (stage 1 and 2) also contributes to improved detection performance, even when the later stages have reduced width. For instance, the configuration of [192-192-96-96] achieves higher AP than [96-192-384-768]. However, increasing early-stage channel width significantly raises computer cost. The configuration of [192-384-384-96] doubles the GFLOPs compared to [192-192-96-96], making it less suitable for deployment on resource-constrained platforms such as UAVs.

Based on the analysis of both experiments, we conclude that enhancing shallow feature extraction in the early stages significantly benefits small object detection. However, this improvement in accuracy comes at the cost of increased computational complexity. Therefore, in order to balance these trade-offs, UAVDet adopts a backbone configuration with block repetitions [4-2-2-1] and channel dimensions [192-192-96-96]. This setup achieves an optimal balance between accuracy and computational efficiency, making it well-suited for UAV-based object detection tasks that require both high performance and lightweight design.

As shown in Table 6, we analyze the impact of inserting the CSPMB module at different stages within the neck network. The baseline model, without any CSPMB, achieves an AP of 22.3%. When CSPMB is applied individually to L3, L4, or L5 layers, only marginal improvements are observed, with L3 yielding the best result at 22.4% AP. In contrast, the most substantial performance gains occur when CSPMB is inserted into multiple layers simultaneously. Specifically, applying CSPMB to both L3 and L4 increases the AP to 22.8% and AP_S to 10.8%. Further adding L5 boosts the AP to 22.9% with a same AP_S, representing gains of 0.6% and 0.5% over the baseline, respectively. These results indicate that multi-layer integration of CSPMB more



(a) Backbone depth configuration



(b) Backbone width configuration

Fig. 8. Ablation study on the configuration of the four stages in the backbone.

effectively promotes information flow and enhances cross-scale features fusion by combining local detail and global context.

However, this improvement comes at the cost of a consistent rise in computational complexity. Especially, when CSPMB is inserted into all stages, the computational cost reaches a peak of 43.80 GFLOPs. Compared with the L3–L4 setting, this configuration increases parameters by 2.83M and computation by 1.11 GFLOPs, yet only improves AP by 0.1%, which is relatively marginal and comes at a disproportionately high cost. This observation suggests that while full-stage integration yields the best detection results, it may not be optimal for deployment in resource-constrained environments such as UAVs. Therefore, in the final UAVDet architecture, we adopt the more cost-effective L3–L4

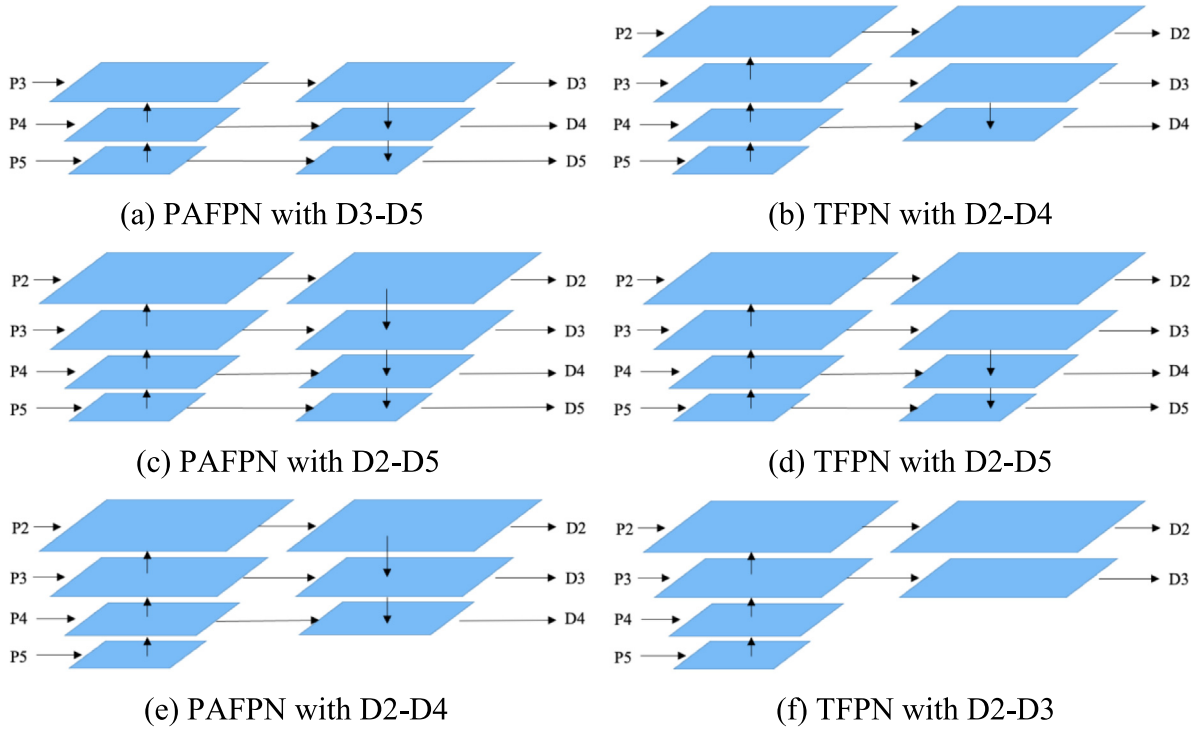


Fig. 9. Comparison between PAFPN and TFPN with different detection heads.

Table 6
Ablation study of the CSPMB placement in the neck.

Layers	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params(M)	GFLOPs
–	22.3	37.5	23.2	10.3	33.6	49.5	24.71	39.27
L3	22.4	37.9	23.5	10.5	33.8	50.5	24.92	40.36
L4	22.3	37.4	23.3	10.4	33.3	50.4	26.33	41.42
L5	22.2	37.4	23.1	10.4	33.4	50.1	27.51	40.21
L3–L4	22.8	38.1	23.9	**10.8**	**34.3**	50.7	26.58	42.69
L4–L5	22.1	37.3	23.2	10.5	33.1	48.0	29.17	42.53
L3–L5	**22.9**	**38.3**	**24.1**	**10.8**	**34.3**	**50.9**	29.41	43.80

insertion strategy, which achieves a better trade-off between accuracy and efficiency, while also maximizing performance in small object detection.

In addition, we conducted an ablation study to evaluate the impact of introducing global modeling on detection efficiency, as summarized in Table 7. When only the CSP module is applied, the model improves AP by 3.8% over the baseline and achieves a 17 FPS increase with only an 8 MB increase in memory consumption, demonstrating the high efficiency of our framework. Adding the Mamba module further raises AP by 0.7% while requiring only 9 MB more memory, although FPS decreases by 23. In contrast, integrating the Transformer causes a drastic drop in both speed and memory efficiency, the CSP + Transformer variant introduces additional quadratic-attention overhead, resulting in a 38% increase in FLOPs and an 8× rise in GPU memory consumption compared to the baseline, which significantly undermines real-time performance.

These findings confirm that the Mamba module offers a more hardware-friendly and scalable alternative for global context modeling in UAV detection frameworks. Its linear-time state-space formulation enables efficient long-range dependency learning without the heavy computational and memory burdens of attention mechanisms. In contrast, Transformer-based modules often require careful architectural redesign to align with CNN feature extraction, making them less compatible and harder to deploy efficiently.

This experiment therefore validates our design choice and highlights that linear sequence modeling is more suitable than quadratic attention for edge-oriented UAV deployment. Furthermore, in real-time-critical scenarios, the Mamba module can be omitted to form an even lighter version of UAVDet, achieving flexible trade-offs between accuracy and efficiency.

A structural comparison of TFPN and PAFPN with varying detection head configurations is illustrated in Fig. 9, and the corresponding performance metrics are summarized in Table 8. The baseline model (a) uses PAFPN with detection heads D3, D4 and D5, achieving an AP of 22.3% and 39.27 GFLOPs, but exhibiting relatively low detection performance on tiny and small objects, reflected in an AP_T and AP_S of only 4.2% and 14.5%, respectively. By adding the high resolution D2 detection head, as in (c), the model shows a notable improvement in tiny and small object detection, increasing AP_T by 2.7% and AP_S by 4.1%, with only a modest rise of 4.58 GFLOPs in computation. Interestingly, removing the D5 head while retaining D2–D4 (e) maintains comparable overall performance (25.0% AP, 6.9% AP_T, 18.4% AP_S), while sightly reduces computation by 2.59 GFLOPs and significantly reducing parameters by 6.14M. Compared with the baseline, this configuration achieves higher accuracy with fewer parameters, despite a marginal 1.99 GFLOPs increase in computation. These results indicate that deeper low-resolution heads may contribute less to tiny object detection but consume disproportionate computational resources.

Similarly, the proposed model using TFPN with detection heads form D2 to D5 (d) achieves 24.6% AP, 6.4% AP_T, and 18.7% AP_S. When the D5 head is remove, the model (b) still attains comparative results (24.7% AP), notably, AP_T increases by 0.7% while AP_S slightly decreases by 0.4%, suggesting that eliminating the deepest head benefits extremely small-object detection while marginally affecting small objects. This configuration also reduces parameters and computational cost by 6.15M and 2.6GFLOPs, respectively, and increasing FPS by 3, further confirming that D5 adds computational overhead without proportional accuracy gains. However, removing an additional D4 head (f) leads to a clear drop in overall AP to 23.8%, AP_T to 6.7% and AP_S to 17.9%, indicating that excessive removal of mid-level heads compromises multi-scale representation. Nevertheless, even this simplified version maintains competitive performance compared with the baseline, showing the robustness of the TFPN design.

Table 7
Ablation study of the CSP with global modeling.

Methods		AP	AP_{50}	AP_{75}	AP_T	AP_S	AP_M	AP_L	Params (M)	GFLOPs	FPS	Memory (MB)
Baseline		22.3	37.5	23.2	4.2	14.5	33.6	49.5	24.71	39.27	59	167
Ours	CSP-only	26.1	44.0	27.3	7.2	20.5	37.5	49.8	3.40	45.18	76	179
	+ Trans	25.6	42.9	26.8	7.0	19.6	37.0	48.2	3.54	54.20	46	1337
	+ Mamba	26.8	44.8	27.8	7.3	20.7	38.3	54.9	3.74	46.63	53	188

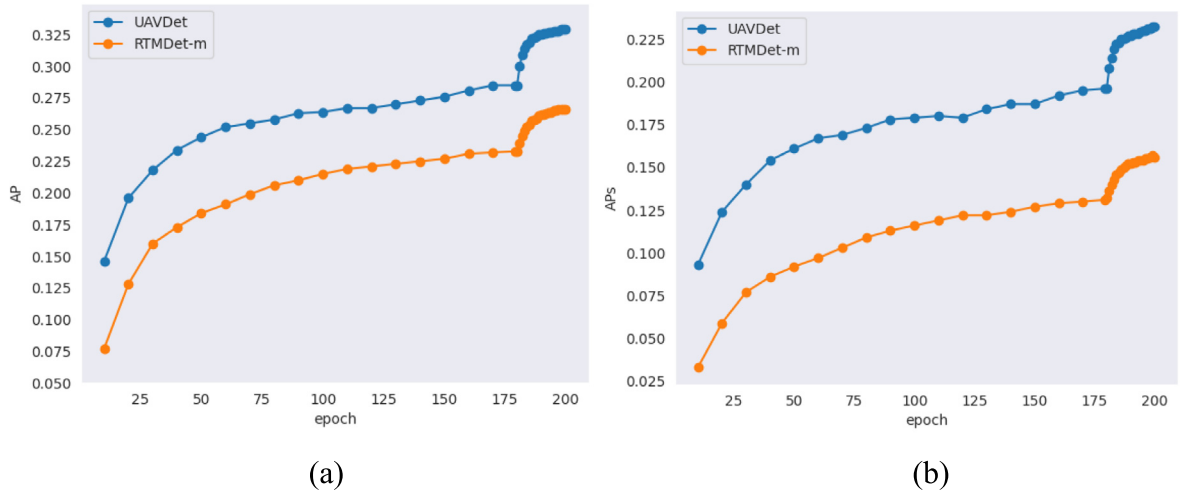


Fig. 10. Comparison of validation performance curves on VisDrone2019 dataset.

Table 8
Ablation study of TFPN and PAFPN with different detection heads.

Methods	AP	AP_{50}	AP_{75}	AP_T	AP_S	AP_M	AP_L	Params(M)	GFLOPs	FPS
(a)	22.3	37.5	23.2	4.2	14.5	33.6	49.5	24.71	39.27	59
(b)	24.7	42.0	25.5	**7.1**	18.3	35.3	**51.4**	16.54	39.03	54
(c)	**25.2**	**42.3**	**26.3**	6.9	18.6	**36.2**	49.1	23.03	43.85	50
(d)	24.6	41.7	25.4	6.4	**18.7**	35.2	49.9	22.69	41.63	51
(e)	25.0	**42.3**	25.8	6.9	18.4	36.1	49.2	16.89	41.26	56
(f)	23.8	40.7	24.3	6.7	17.9	34.3	47.2	14.83	35.76	63

Overall, these results confirm that introducing the high-resolution D2 head plays a critical role in enhancing detection of extremely small objects, which are abundant in UAV imagery. Preserving high-resolution, low-level features substantially improves recall for dense tiny instances that often vanish after multiple down-sampling stages.

When comparing TFPN with D2–D4 (b) to PAFPN with D2–D4 (e), the former achieves slightly lower overall performance but maintains nearly identical AP_T and AP_S. This outcome stems from the removal of the direct connection from P2 to P3, which weakens the propagation of fine-grained details into deeper layers and slightly affects medium-object performance. However, this modification helps suppress background noise in large object detection and further improve inference efficiency. In addition, the TFPN variant shows reducing model complexity and higher FPS compared its PAFPN counterpart, demonstrating the efficiency of our design. Based on these findings above, our final UAVDet model adopts TFPN with D2–D4 detection head configuration to achieve a better trade-off between performance and efficiency.

4.6. Visualization

Fig. 10 illustrates the validation performance curves of UAVDet and RTMDet-m throughout the training process on the VisDrone2019 dataset. The left subfigure shows the evolution of overall AP, while the right subfigure focuses on AP for small objects. As training progresses, UAVDet consistently outperforms RTMDet-m in both AP and AP_S across all epochs. Particularly in the small object detection task, UAVDet maintains a significant performance gap, demonstrating its enhanced sensitivity to small-scale targets. Notably, UAVDet exhibits a faster convergence speed in the early training stages and achieves a higher final performance margin, indicating its superior feature extraction and optimization capability. These results validate the effectiveness of the proposed architectural improvements.

To intuitively validate the rationality of our network design, we visualize the heatmaps of feature maps at each backbone stage (S1–S4), from shallow to deep, as shown in Fig. 11. Fig. 11(a) displays a Transformer-based detector, due to the global attention mechanism, it exhibits widespread activation across the entire image in the early stages, which potentially introducing noise that propagates through the network. In contract, the CNN-based model in Fig. 11(b) shows localized activation early on, and gradually expanding to broader contextual regions in deeper layers. Mamba-YOLO strikes a balance between these two approaches by integrating both local and global attention mechanisms, enabling early-stage detail focus and late-stage semantic awareness, as demonstrated in Fig. 11(c).

Fig. 11(d) illustrates a detection model specifically designed for small objects in UAV imagery, but its S1 stage activates nearly the entire scene, causing irrelevant regions to persist into S4. Similarly, the baseline exhibits stronger low-level responses, but much of this activation is scattered across irrelevant background areas. In the deeper stages (e.g., S4), the activation tends to focus only on close-up objects or high-contrast non-target regions, indicating poor semantic discrimination. By comparison, UAVDet presents more concentrated and semantically meaningful activation throughout all stages. It highlights true

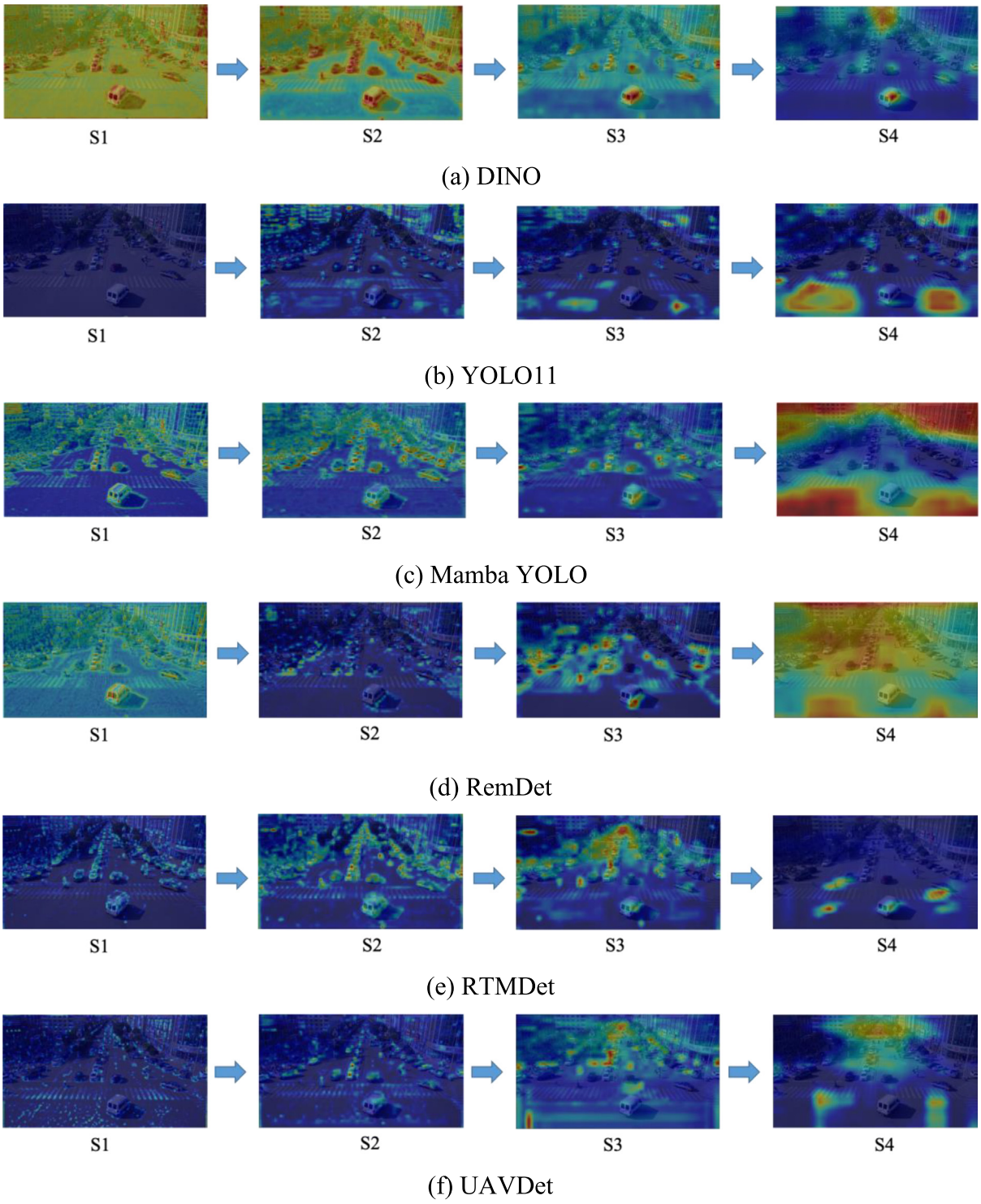


Fig. 11. Comparison of feature representation at different backbone stages.

target regions early while suppressing background noise, and maintains attention to small or distant targets in deeper layers. These results demonstrate that UAVDet more effectively preserves spatial detail and enhances focus on small, dense targets, validating the effectiveness of its backbone design in UAV detection scenarios.

While quantitative metrics such as AP offer an overall assessment of detection performance, they often overlook specific classification errors across object categories. To gain deeper insights into the category-wise prediction behavior and the impact of inter-class confusion and background interference, we present the normalized confusion matrices of six representative detectors on the test set in Fig. 12. It can be observed that UAVDet significantly improves classification accuracy across all categories, especially for small and confusing object classes such as pedestrian, people, bicycle, and motor.

In most confusion matrices, a large number of foreground objects are misclassified as background, for instance, over 60% of pedestrians and people are misclassified. Particularly, in Mamba-YOLO (b), up to 75% pedestrians and 82% of people are incorrectly classified. In contrast, UAVDet (f) significantly reduces such confusion to 53%, correctly identifying 42% of pedestrians and 27% of people, while

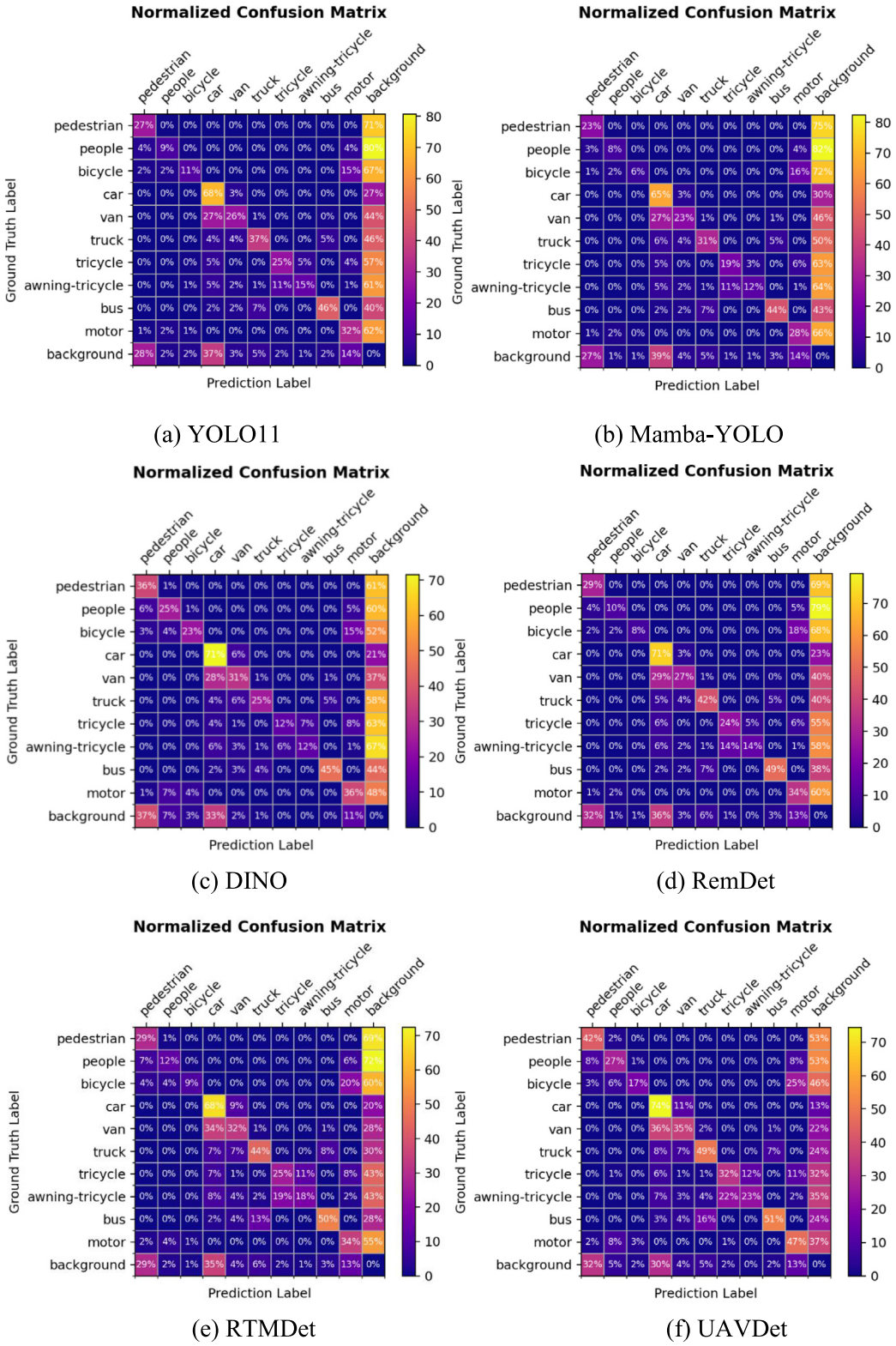


Fig. 12. Comparison of confusion matrices.

lowering the background misclassification rate. Additionally, for classes with high intra-class similarity, UAVDet shows stronger discrimination ability, resulting in fewer incorrect predictions. For example, UAVDet correctly identifies 49% of trucks, a 5% improvement over RTMDet (e), and improves classification of tricycles and awning-tricycles by 11% and 8%, respectively. In contrast, the poorest performing model, DINO (c), achieves only 25% accuracy for trucks and merely 12% for both tricycles and awning-tricycles, highlighting its limitations in distinguishing visually similar categories. Overall, the confusion matrix analysis demonstrates that UAVDet is more robust against inter-class

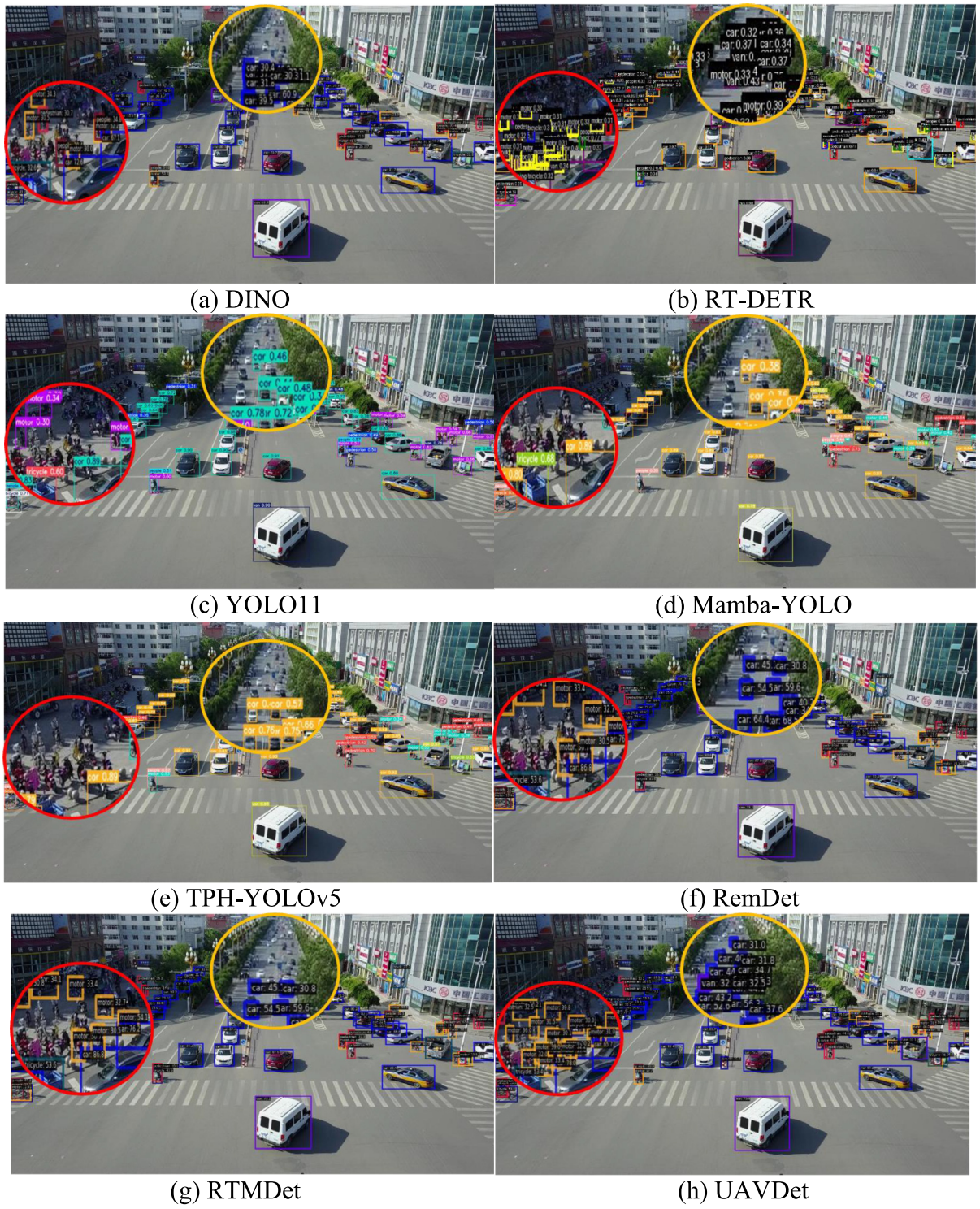


Fig. 13. Compare the detection results in daytime conditions.

confusion and background interference, particularly in complex aerial scenes with densely packed small objects. This further validates its superior detection capability.

To better showcase the superiority of our model's detection performance, we conduct a visual comparison with several state-of-the-art detectors on the VisDrone dataset under different challenging scenarios, as illustrated in Figs. 13 to 15. In Fig. 13, two regions are highlighted with red and yellow circles to emphasize crowded motor areas and distant small objects, respectively. In the red-circled region, most detectors struggle with overlapping instances. Methods like YOLO11, TPH-YOLOv5, and RTMDet suffer from missed or inaccurate detections, largely due to insufficient feature resolution or poor localization. In contrast, UAVDet generates more precise bounding boxes and achieve higher recall, successfully detecting individual motors with minimal misidentification or omission.

The yellow-circled area contains small and distant objects that are inherently difficult to detect. Transformer-based methods like RT-DETR and DINO leverage global context but often produce false positives

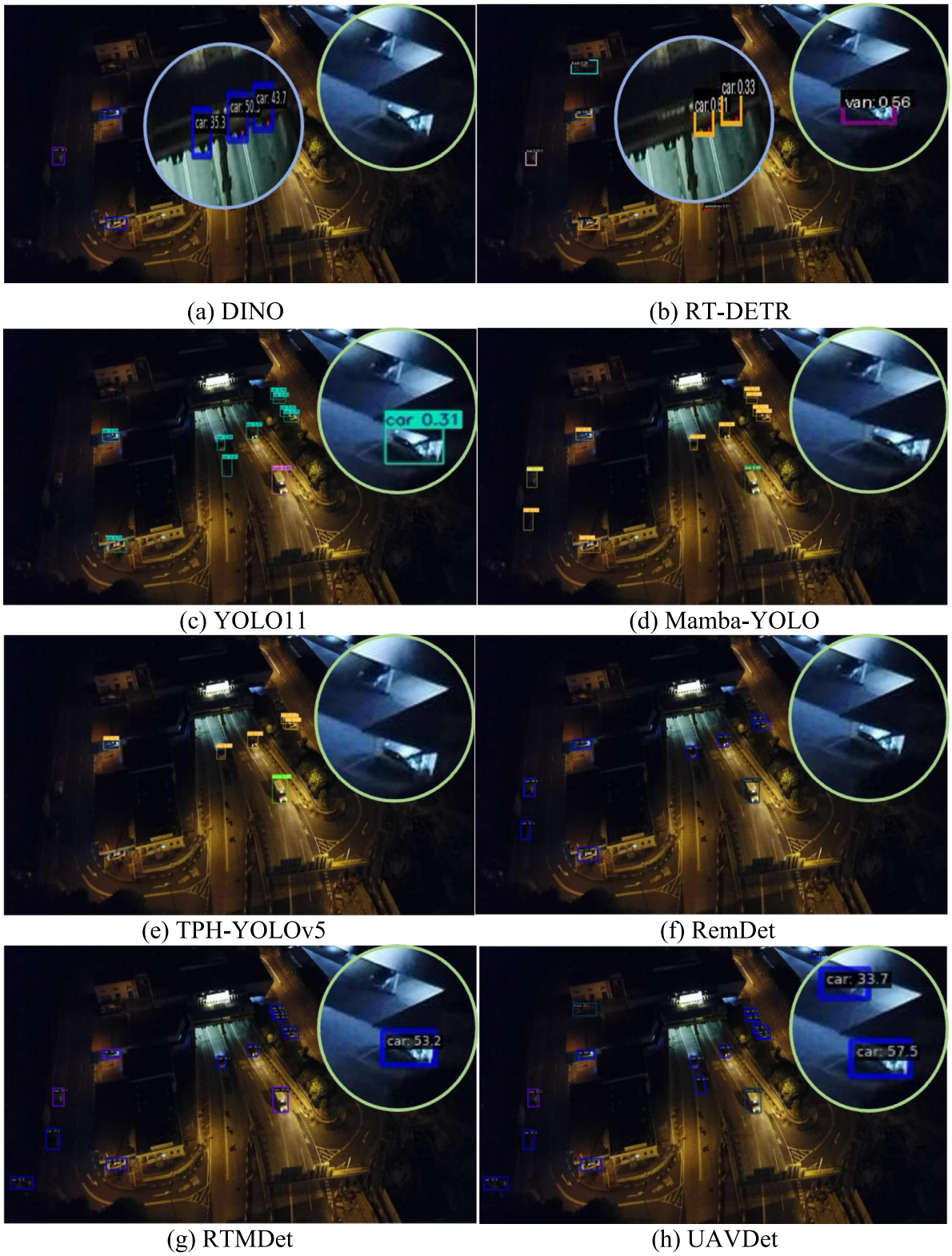


Fig. 14. Compare the detection results in nighttime conditions.

or uncertain classifications. Conversely, UAVDet achieves better localization and confidence for these challenging targets, showing strong robustness to small-scale and low-resolution instances.

Fig. 14 presents the detection performances under nighttime conditions, where green-circled areas include occluded targets and blue-circled areas denote background regions that are prone to false detection due to limited illumination. Most detectors exhibit significant performance degradation in these conditions. In contrast, models like DINO and RT-DETR tend to misclassify background clutter as foreground objects due to insufficient local feature modeling. UAVDet effectively mitigates this issue, showcasing superior robustness and discrimination under low-light scenarios.

Fig. 15 compares detection results under cluttered urban environments. The green-circled regions contain a high density of objects and

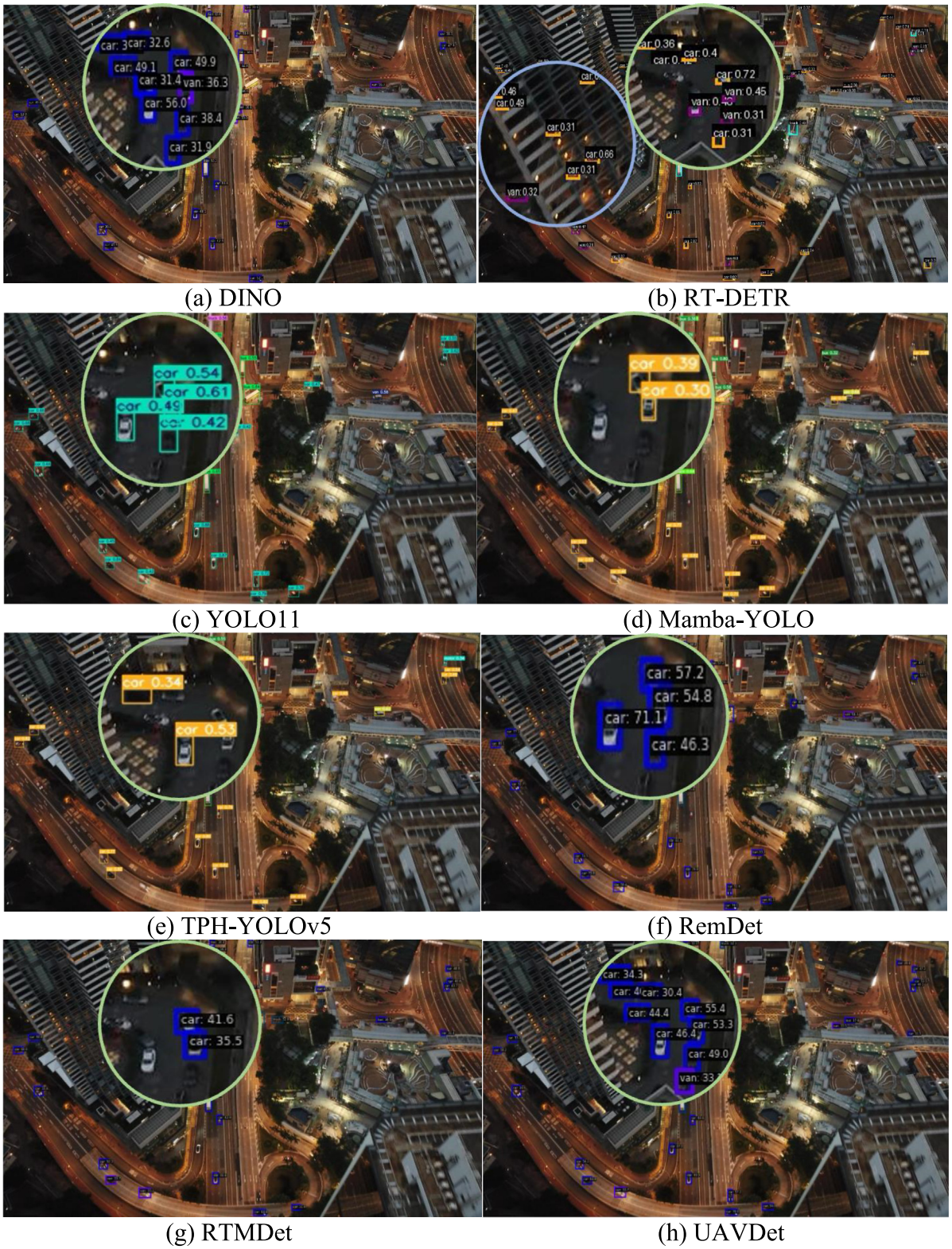


Fig. 15. Compare the detection results in cluttered environment.

complex background textures, while the blue-circled area highlights regions with strong background interference. Transformer-based detectors, such as RT-DETR, suffer from severe background confusion and frequent false positives. Meanwhile, models like Mamba-YOLO and RTMDet struggle to separate closely spaced vehicles, resulting in low-confidence or partial detections. By contrast, UAVDet achieves more complete and precise detection, benefiting from enhanced context-awareness and refined multi-scale fusion.

To further evaluate the generalization capability of our model, we performed additional visual analyses on the UAVDT and DroneVehicle datasets. Fig. 16 compares several representative methods on UAVDT under cross-region scenes involving both urban roads and coastal areas.

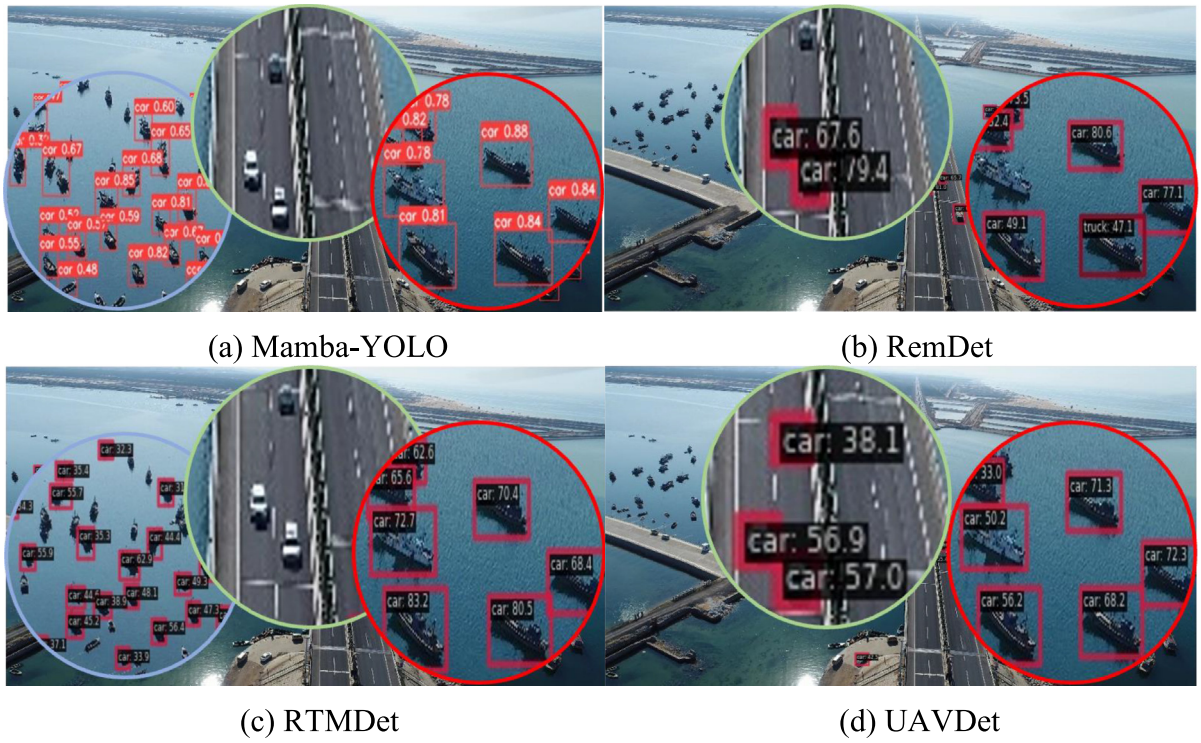


Fig. 16. Compare the detection results in UAVDT.

In the highway region (green circles), UAVDet accurately detects small and distant vehicles, whereas other detectors miss these targets to varying extents. In the offshore region (blue circles), both UAVDet and RemDet effectively suppress false positives, showing stronger robustness to background interference and better discrimination for small objects than Mamba-YOLO and RTMDet. However, in the nearshore region (red circles), all models occasionally misclassify ships as vehicles, revealing a shared limitation among current detectors rather than an issue specific to UAVDet.

Fig. 17 illustrates detection results on DroneVehicle under low-illumination nighttime scenes. As shown, RemDet completely fails to detect any targets, and RTMDet produces three false positives in dark background regions. In contrast, UAVDet accurately detects all valid objects with only one false positive, achieving the closest alignment with the ground-truth annotations. This result confirms that UAVDet effectively maintains feature consistency and semantic discrimination even when texture information is severely degraded by darkness.

In summary, UAVDet consistently outperforms existing methods across various challenging scenarios, involving small, dense, occluded, and low-visibility targets, offering both high precision and reliable coverage. Although UAVDet still encounters difficulties in extremely dark or visually ambiguous regions, it nevertheless demonstrates strong robustness and generalization capability across diverse environments.

5. Conclusion

This work presents UAVDet, a CNN–Mamba hybrid object detector tailored for UAV imagery, where detecting small, densely distributed objects under complex backgrounds remains highly challenging. UAVDet combines the fine-grained local representation of CNNs with the efficient global modeling capability of Mamba, while incorporating the CSPMB and TFPN modules to enhance global–local feature interaction and tiny object sensitivity. The proposed architecture achieves a strong balance between detection accuracy, computational efficiency, and real-time performance, demonstrating robust cross-domain generalization and stable scale-aware detection across diverse UAV environments.

Although the Mamba module achieves linear computational complexity, its sequential state-update mechanism limits the full utilization of GPU parallelism compared with attention-based architectures. In addition, the current implementation has not yet been thoroughly validated on embedded or multimodal UAV platforms, where strict memory and energy constraints may further amplify latency differences. Future work will focus on enhancing the parallelism and scalability of Mamba-based architectures, applying model compression and quantization for efficient edge deployment, and extending UAVDet to multimodal and low-light UAV perception tasks to further broaden its applicability.

CRediT authorship contribution statement

Yiming Yang: Writing – original draft, Visualization, Methodology. **Feng Guo:** Writing – original draft, Visualization, Supervision. **Pei Niu:** Writing – original draft, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by the Natural Science Foundation of China (Grant Number: 52308457), China Postdoctoral Science Foundation (Grant Number: 2024M761811) and Natural Science Foundation of Shandong Province (Grant Number: ZR2023QE220). All the supports are highly appreciated.

Data availability

Data will be made available on request.

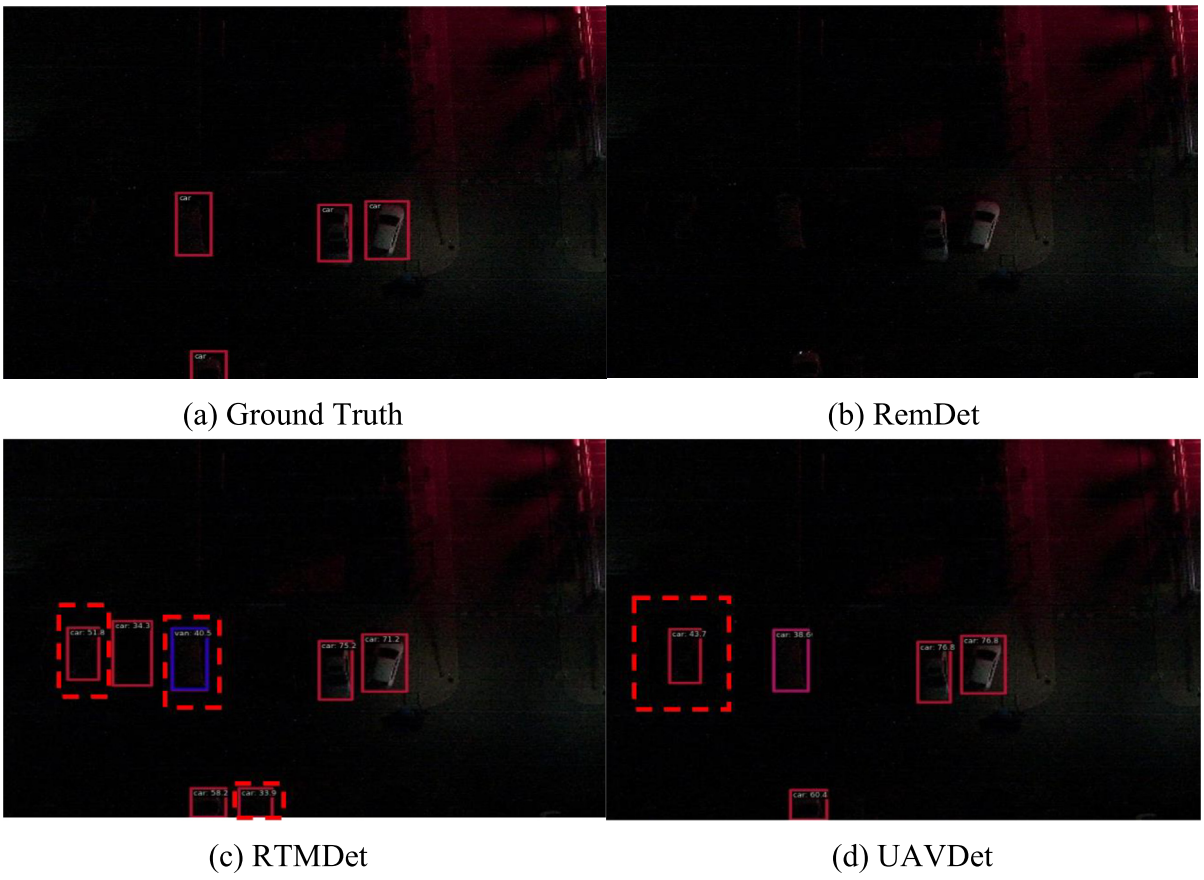


Fig. 17. Compare the detection results in DroneVehicle.

References

Cai, Z., Vasconcelos, N., 2018. Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6154–6162.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229.

Chen, Z., Ji, H., Zhang, Y., Zhu, Z., Li, Y., 2023. High-resolution feature pyramid network for small object detection on drone view. IEEE Trans. Circuits Syst. Video Technol. 34 (1), 475–489.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., et al., 2019. MMDetection: Open mmlab detection toolbox and benchmark. p. 5, arXiv 2019. arXiv preprint arXiv:1906.07155.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint, arXiv:2010.11929.

Du, B., Huang, Y., Chen, J., Huang, D., 2023. Adaptive sparse convolutional networks with global context enhancement for faster object detection on drone images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13435–13444.

Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., et al., 2018. The unmanned aerial vehicle benchmark: Object detection and tracking. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 370–386.

Du, D., Zhu, P., Wen, L., Bian, X., Lin, H., Hu, Q., et al., 2019. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0.

Gao, Y., Ding, R., Zhou, F., Wu, Q., 2024. UAV object detection based on joint YOLO and transformer. In: 2024 International Conference on Ubiquitous Communication. Ucom, pp. 202–206.

Girshick, R., 2015. Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448.

Gu, A., Dao, T., 2023. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752.

Hendria, W.F., Phan, Q.T., Adzaka, F., Jeong, C., 2023. Combining transformer and CNN for object detection in UAV imagery. ICT Express 9 (2), 258–263.

Kong, X., Ni, C., Duan, G., Shen, G., Yang, Y., Das, S.K., 2024. Energy consumption optimization of UAV-assisted traffic monitoring scheme with tiny reinforcement learning. IEEE Internet Things J. 11 (12), 21135–21145.

Li, Y., Mao, H., Girshick, R., He, K., 2022. Exploring plain vision transformer backbones for object detection. In: European Conference on Computer Vision. pp. 280–296.

Li, C., Zhao, R., Wang, Z., Xu, H., Zhu, X., 2025. Remdet: Rethinking efficient model design for UAV object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 4643–4651.

Liang, W., Tan, J., He, H., Xu, H., Li, J., 2024. Detection of small objects from UAV imagery via an improved swin transformer. In: IGARSS 2024-2024 IEEE International Geoscience and Remote Sensing Symposium. pp. 9134–9138.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al., 2014. Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755.

Liu, Y., He, M., Hui, B., 2025. ESO-DETR: An improved real-time detection transformer model for enhanced small object detection in UAV imagery. Drones 9 (2), 143.

Liu, Z., Lyu, Y., Wang, L., Han, Z., 2019. Detection approach based on an improved faster RCNN for brace sleeve screws in high-speed railways. IEEE Trans. Instrum. Meas. 69 (7), 4395–4403.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8759–8768.

Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., et al., 2024. Vmamba: Visual state space model. In: Advances in Neural Information Processing Systems, vol. 37, 103031-103063.

Liu, M., Wang, X., Zhou, A., Fu, X., Ma, Y., Piao, C., 2020. UAV-yolo: Small object detection on unmanned aerial vehicle perspective. Sensors 20 (8), 2238.

Lu, W., Lan, C., Niu, C., Liu, W., Lyu, L., Shi, Q., Wang, S., 2023. A CNN-transformer hybrid model based on cswin transformer for UAV image object detection. IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens. 16, 1211–1231.

Lyu, C., Zhang, W., Huang, H., Zhou, Y., Wang, Y., Liu, Y., et al., 2022. Rtmdet: An empirical study of designing real-time object detectors. arXiv preprint arXiv:2212.07784.

Ma, Z., Zhou, L., Wu, D., Zhang, X., 2025. A small object detection method with context information for high altitude images. Pattern Recognit. Lett. 188, 22–28.

Shin, Y., Shin, H., Ok, J., Back, M., Youn, J., Kim, S., 2024. DCEF2-YOLO: Aerial detection YOLO with deformable convolution–efficient feature fusion for small target detection. Remote. Sens. 16 (6), 1071.

Sun, Y., Cao, B., Zhu, P., Hu, Q., 2022a. Drone-based RGB-infrared cross-modality vehicle detection via uncertainty-aware learning. IEEE Trans. Circuits Syst. Video Technol. 32 (10), 6700–6713.

Sun, W., Dai, L., Zhang, X., Chang, P., He, X., 2022b. RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. Appl. Intell. 52 (8), 8448–8463.

Tan, M., Pang, R., Le, Q.V., 2020. Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10781–10790.

Tang, S., Zhang, S., Fang, Y., 2024. HIC-YOLOv5: Improved YOLOv5 for small object detection. In: 2024 IEEE International Conference on Robotics and Automation. ICRA, pp. 6614–6619.

Terven, J., Córdova-Esparza, D.-M., Romero-González, J.-A., 2023. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Mach. Learn. Knowl. Extr. 5 (4), 1680–1716.

Wang, Z., Li, C., Xu, H., Zhu, X., Li, H., 2025. Mamba yolo: A simple baseline for object detection with state space model. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 8205–8213.

Wang, Z., Men, S., Bai, Y., Yuan, Y., Wang, J., Wang, K., Zhang, L., 2024a. Improved small object detection algorithm CRL-YOLOv5. Sensors 24 (19), 6437.

Wang, J., Zhao, W., Liu, C., Yang, H., Xu, W., 2024b. Real-time object detection based on mamba and YOLOv8. In: 2024 4th International Conference on Industrial Automation, Robotics and Control Engineering. IARCE, pp. 255–260.

Wu, S., Lu, X., Guo, C., 2024. YOLOv5_mamba: unmanned aerial vehicle object detection based on bidirectional dense feedback network and adaptive gate feature fusion. Sci. Rep. 14 (1), 22396.

Wu, Y., Qin, Y., Wang, Z., Jia, L., 2018. A UAV-based visual inspection method for rail surface defects. Appl. Sci. 8 (7), 1028.

Xiao, Y., Di, N., 2024. SOD-YOLO: A lightweight small object detection framework. Sci. Rep. 14 (1), 25624.

Yundong, L., Han, D., Hongguang, L., Zhifeng, X., 2020. Multi-block SSD based on small object detection for UAV railway scene surveillance. Chin. J. Aeronaut. 33 (6), 1747–1755.

Zhang, H., Deng, L., Lin, S., Zhang, H., Dong, J., Wan, D., et al., 2024. LES-YOLO: efficient object detection algorithm used on UAV for traffic monitoring. Meas. Sci. Technol. 36 (1), 016008.

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., et al., 2022. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605.

Zhang, Y., Liu, T., Zhen, J., Kang, Y., Cheng, Y., 2025a. Adaptive downsampling and scale enhanced detection head for tiny object detection in remote sensing image. IEEE Geosci. Remote. Sens. Lett. 22, 1–5.

Zhang, Y., Wang, S., Zhang, Y., Yu, P., 2025b. Asymmetric light-aware progressive decoding network for RGB-thermal salient object detection. J. Electron. Imaging 34 (1), 013005-013005.

Zhang, Y., Xiao, Y., Zhang, Y., Zhang, T., 2025c. Video saliency prediction via single feature enhancement and temporal recurrence. Eng. Appl. Artif. Intell. 160, 111840.

Zhang, Y., Yang, Y., Kang, W., Zhen, J., 2025d. Cross-erasure enhanced network for occluded person re-identification. Pattern Recognit. Lett.

Zhang, Y., Yu, P., Xiao, Y., Wang, S., 2025e. Pyramid-structured multi-scale transformer for efficient semi-supervised video object segmentation with adaptive fusion. Pattern Recognit. Lett.

Zhang, Y., Zhang, T., Wang, S., Yu, P., 2025f. An efficient perceptual video compression scheme based on deep learning-assisted video saliency and just noticeable distortion. Eng. Appl. Artif. Intell. 141, 109806.

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., et al., 2024. Detrs beat yolos on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16965–16974.

Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X., 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. arXiv preprint arXiv:2401.09417.

Zhu, X., Lyu, S., Wang, X., Zhao, Q., 2021. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2778–2788.