International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-024-02278-1

ORIGINAL ARTICLE

Doublem-net: multi-scale spatial pyramid pooling-fast and multi-path adaptive feature pyramid network for UAV detection

Zhongxu Li[1,2] · Qihan He[2] · Hong Zhao[3] · Wenyuan Yang[1,2]

Received: 21 October 2023 / Accepted: 3 July 2024

Abstract
Unmanned aerial vehicles (UAVs) are extensively applied in military, rescue operations, and traffic detection fields, resulting from their flexibility, low cost, and autonomous flight capabilities. However, due to the drone's flight height and shooting angle, the objects in aerial images are smaller, denser, and more complex than those in general images, triggering an unsatisfactory target detection effect. In this paper, we propose a model for UAV detection called DoubleM-Net, which contains multi-scale spatial pyramid pooling-fast (MS-SPPF) and Multi-Path Adaptive Feature Pyramid Network (MPA-FPN). DoubleM-Net utilizes the MS-SPPF module to extract feature maps of multiple receptive field sizes. Then, the MPA-FPN module first fuses features from every two adjacent scales, followed by a level-by-level interactive fusion of features. First, using the backbone network as the feature extractor, multiple feature maps of different scale ranges are extracted from the input image. Second, the MS-SPPF uses different pooled kernels to repeat multiple pooled operations at various scales to achieve rich multi-perceptive field features. Finally, the MPA-FPN module first incorporates semantic information between each adjacent two-scale layer. The top-level features are then passed back to the bottom level-by-level, and the underlying features are enhanced, enabling interaction and integration of features at different scales. The experimental results show that the mAP50-95 ratio of DoubleM-Net on the VisDrone dataset is 27.5%, and that of Doublem-Net on the DroneVehicle dataset in RGB and Infrared mode is 55.0% and 60.4%, respectively. Our model demonstrates excellent performance in air-to-ground image detection tasks, with exceptional results in detecting small objects.

Keywords Object detection · Feature pyramid networks · Spatial pyramid pooling · Adaptive spatial fusion · UAV

1 Introduction

In recent years, object detection has made significant progress in computer vision. This crucial task involves identifying and localizing different objects in digital images, including people, animals, and vehicles [1, 2]. With the development and popularity of drone technologies, they have been widely applied across various domains and generated massive aerial image data. Meanwhile, deep learning-based target detection techniques have also made great strides in effectively parsing image contents. Therefore, researching target detection algorithms tailored for drone aerial images enables the integration of both technologies to play an important role in intelligent transportation [3, 4], environmental monitoring [5, 6], emergency rescue, and disaster relief [7, 8].

One-stage and two-stage detectors are two distinct research paradigms in object detection. The former directly predicts the bounding boxes and corresponding class labels

✉ Wenyuan Yang
yangwycn@gmail.com

Zhongxu Li
zhongxulee18@163.com

Qihan He
qihanhe27@163.com

Hong Zhao
hongzhaocn@163.com

[1] Fujian Key Laboratory of Granular Computing and Application, Minnan Normal University, Zhangzhou 363000, China

[2] School of Mathematics and Statistics, Minnan Normal University, Zhangzhou 363000, China

[3] School of Computer Science, Minnan Normal University, Zhangzhou 363000, China

Published online: 26 July 2024

of objects, bypassing the need for a region proposal network. In contrast, the latter relies on a regional proposal network to perform object detection. R-CNN [9], a seminal work, ushers in the era of deep learning in object detection. Due to its computationally intensive nature and time-consuming algorithms, SPPNet [10] proposes shared convolutional calculations and pyramid pooling, significantly reducing storage requirements and training time. Faster R-CNN [11] further enhances performance by leveraging a Region Proposal Network to extract and integrate proposals into the overall network. On the other hand, the Feature Pyramid Network (FPN) [12] introduces a straightforward and efficient method for creating a feature pyramid, enabling object detection across multiple scales. One-stage detectors represent a class of object detection algorithms that employ convolutional neural networks to predict objects' classes and locations directly. Among the various types of one-stage detectors, the YOLO series is a leading approach. YOLOv1-3 [13–15] stand out as groundbreaking algorithms within this series. Additionally, YOLOv4 [16] divides the network architecture into three components: the backbone, neck, and head. It leverages bag-of-freebies and bag-of-specials techniques to design a framework optimized for training on a single GPU. Other competitive one-stage object detector algorithms include YOLOv5-8 [17–20]. Recently, YOLOv9 [21] has also made its debut. Typically, one-stage algorithms excel in speed but may compromise some accuracy, whereas two-stage algorithms, while slower, can attain higher accuracy. Unmanned aerial vehicles (commonly known as drones) occupy a pivotal position in various applications. The application of object detection technology to drone-captured scenarios has garnered significant attention, primarily due to its vast array of practical uses. In recent years, object detection in drone-captured images has garnered widespread attention, with remarkable progress achieved by utilizing deep convolutional neural networks on prominent large-scale benchmark datasets.

However, air-to-ground images differ significantly from natural images, posing numerous challenges for object detection in aerial images. The flight altitude of the drone is tens to hundreds of meters high, resulting in a large field of view, small target size, varied viewpoints, and dynamic environments. As shown in Fig. 1, the target scale distribution, target center point distribution, and some picture

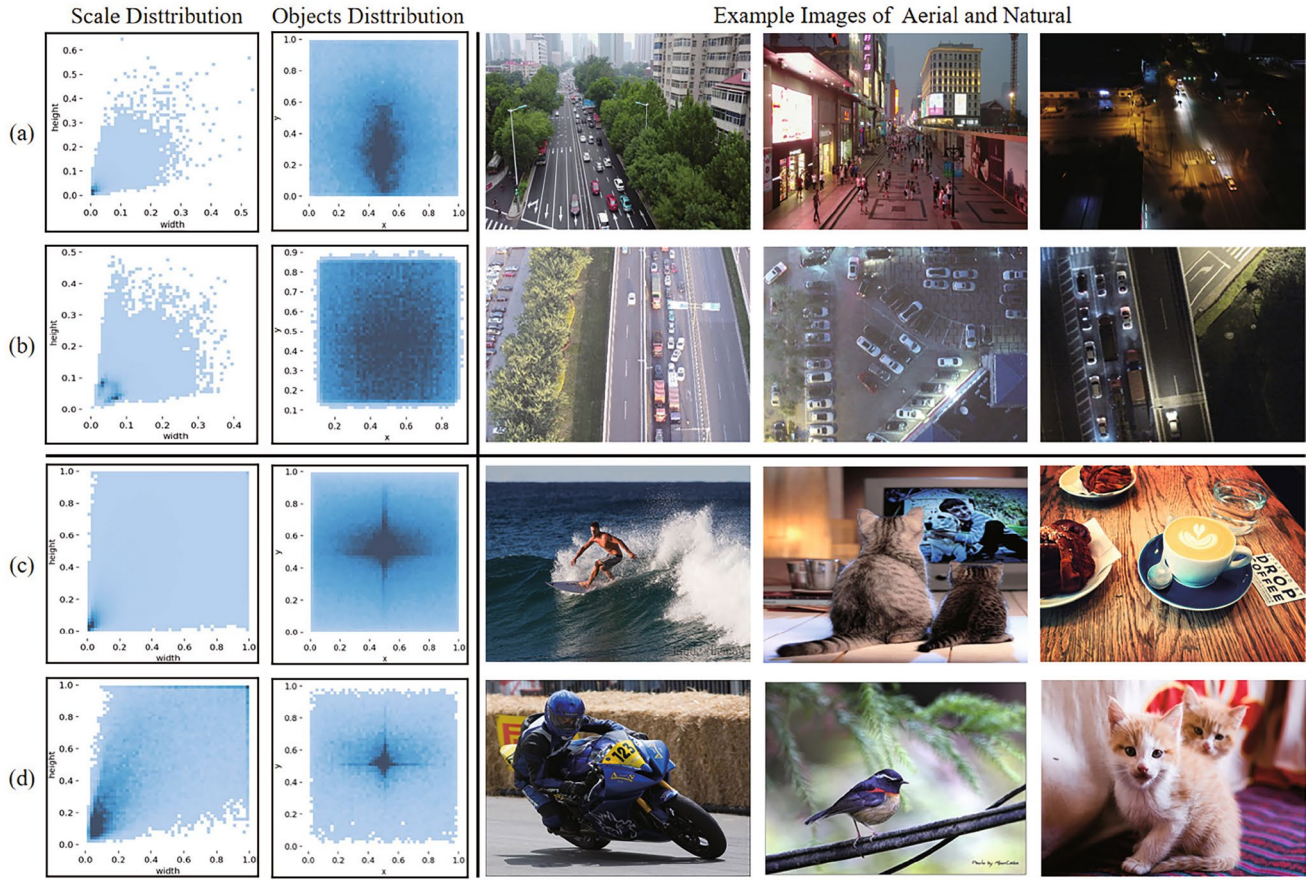

Fig. 1 **a–d** are the object scale distribution, target center point distribution, and some image samples of the VisDrone, DroneVehicle, COCO, and VOC datasets, respectively

samples of the VisDrone, DroneVehicle, COCO, and VOC datasets are exhibited. Most objects in the aerial images are smaller than 32×32 pixels, with targets throughout the image. This brings unique difficulties when designing deep learning-based target detection algorithms. It is imperative to thoroughly study the characteristics of UAV images, and devise detection frameworks that can handle small targets and varied viewpoints, for example, by utilizing multi-scale feature extraction, fusing high- and low-level semantic information, etc., to improve target detection performance on UAV images further.

In this study, we propose an object detection model for UAV images named DoubleM-Net, which is specially designed with a multi-scale spatial pyramid pooling-fast (MS-SPPF) module and multi-path adaptive feature pyramid network (MPA-FPN) module to effectively handle challenges like large scale variations and complex scenes in UAV images. Specifically, the backbone network first extracts multi-scale feature maps from the raw images. Then, the MS-SPPF module repeatedly conducts pooled operations at varied scales to form feature maps with rich multi-scale receptive fields, which can capture information on different-sized objects and enhance the model's robustness to scale changes. Moreover, the MPA-FPN module first fuses semantic information between adjacent scale layers. Then, it enhances lower-level features by propagating top-level semantic features down in a multi-path manner to realize interaction and integration of multi-scale features. This retains fine-grained low-level features while sufficiently incorporating high-level semantics. By jointly utilizing MS-SPPF and MPA-FPN modules, DoubleM-Net can fully exploit multi-scale feature information to detect small objects in UAV images effectively and improve adaptability to complex scenes, achieving superior detection performance.

The main contributions of this paper are as follows:

- We construct a novel plug-and-play feature extraction module MS-SPPF. The module incorporates the ideas of SPP and SPPF by using pooling kernels of different sizes (k = 5, 9, 13) for multiple pooling operations. This design enables MS-SPPF to capture spatial features at different scales simultaneously and enhance the richness of the features through multiple pooling operations. MS-SPPF further compensates for the shortcomings of traditional methods in multi-scale feature extraction and improves the accuracy of target detection.
- In order to overcome the limitations of traditional feature pyramid networks in solving the scale change problem, we propose an original feature pyramid structure called MPA-FPN. By designing the feature fusion method, MPA-FPN effectively reduces the information contradiction between non-neighboring features and enhances the interaction between low-level and high-level semantic information. MPA-FPN not only improves the detection effect of the model on small targets but also provides new ideas and methods to cope with the scale change problem in target detection.
- Based on MS-SPPF and MPA-FPN, we further construct a model called DoubleM-Net. It is validated on two challenging datasets, VisDrone and DroneVehicle, and its performance is comprehensively evaluated. The experimental results show that DoubleM-Net achieves a mAP50-95 of 27.5% on the VisDrone dataset and 55.0% and 60.4% on the DroneVehicle dataset in RGB and Infrared modes, respectively.

The structure of this paper can be outlined as follows. In Sect. 2, the related literature is discussed. Section 3 presents the design of the DoubleM-Net model, providing a comprehensive characterization. Section 4 elaborates on the implementation of the proposed method, including the setup and results. Finally, Sect. 5 concludes the paper, highlighting the findings, and proposes potential directions for future research.

2 Related work

This section provides an overview of notable techniques and methods in object detection, which serve as the foundation for developing our proposed DoubleM-Net model. Specifically, we discuss the YOLO series model, spatial pyramid pooling, feature pyramid-related technologies, and object detectors designed for aerial images.

2.1 YOLO-series model

The YOLO object detection framework achieves an excellent balance between speed and accuracy, making it stand out among various object detection algorithms for efficiently and accurately detecting objects in images. YOLOv1-v3 [13–15] establish the foundational YOLOs, introducing a single-stage detection architecture with backbone-neck-head components. It enabled multi-scale object detection through branches, becoming a prominent single-stage object detection model. YOLOv4 [16] utilizes CSPDarknet to improve computational efficiency. YOLOv5 [17] is the first PyTorch implementation, developing a new CSP-based backbone and decoupled classification and regression detection heads. Based on YOLOv5, Xu et al. [22] propose Lite-YOLOv5, an on-board SAR ship detection model that is both lightweight and high-performance. It introduces a lightweight cross stage partial (L-CSP) module combined with network pruning techniques to reduce the computational complexity. In addition, Lite-YOLOv5 has been successfully ported to the

NVIDIA Jetson TX2 embedded platform, providing robust support for on-board evaluation. YOLOv6 v3.0 [18] simplifies the Spatial Pyramid Pooling-Fast (SPPF) module in YOLOv5 to SimSPPF, improving accuracy with negligible change in speed. YOLOv7 [19] proposes the E-ELAN module to accelerate convergence. Alibaba's DAMO-YOLO [23] employs automatic neural architecture search to obtain an efficient backbone and designed a new Efficient RepGFPN neck structure to fuse multi-scale features through CSP-Satge. DAMO-YOLO also utilize AlignOTA for dynamic label assignment and knowledge distillation for further speed improvements through model compression. YOLOv8 [20] incorporates the C2f component to enhance feature expression and applied a decoupled anchor-free head design for multi-task recognition. Subsequently, the Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Network (GELAN) architectures in YOLOv9 [21] introduces a new paradigm for network design and optimization. The PGI allows the model to adaptively regulate the gradient flow during training, potentially mitigating problems such as gradient vanishing or gradient explosion. The GELAN architecture, on the other hand, uses gradient-based path planning to efficiently aggregate features across multiple scales and resolutions, improving the model's ability to detect targets of different sizes. Through continuous architectural evolution, YOLO series models have consistently optimized model efficiency and effectiveness, advancing single-stage real-time recognition technologies.

2.2 Spatial pyramid pooling

The Spatial Pyramid Pool is a network structure that handles objects at different scales and is designed to capture features at different scales. He et al. [24] introduce SPP into a deep convolutional neural network to solve the feature extraction problem and classify images with different input sizes. The core idea of SPP is to realize scale invariance by mapping features of different scales of the input image onto a fixed-size feature vector through the pyramid pooling layer. Motivated by SPP, the semantic segmentation model DeepLabv2 [25] proposes Atrous Spatial Pyramid Pooling. The module uses multiple parallel atrous convolution layers with different sampling rates. The convolution kernel of different receptive fields is constructed through different atrous rates to obtain multi-scale object information. From simulating the receptive field of human vision to enhancing the feature extraction capability of the network, Liu et al. [26] introduce Inception [27] into the proposed Receptive Field Block module. The main idea is to add an atrous convolution based on Inception, thus effectively increasing the receptive field. YOLOv5 [17] introduces a novel Spatial Pyramid Pooling-Fast method built upon SPP to improve speed. It will apply the maximum pooling of different scales on a feature layer of the network, and the pooling kernel is 5×5, 9×9, and 13×13, respectively. Finally, 1×1 convolution is applied to channel integration of the feature maps of different scales, and they are fused into one feature graph. Pooling at different scales can capture a more extensive range of content and enhance the multi-scale processing capability of the model. Based on the SPPF, YOLOv6 v3.0 [18] proposes that the Simplified SPPF. Although it only has one activation function from SPPF, it is much faster than SPPF. YOLOv7 [19] introduces the idea of a Cross Stage Partial (CSP) Network based on SPP and proposes the SPPCSPC module. CSP divides the features into two parts: routinely processed, and the SPP structure processes the other. Finally, the two parts are combined. Although the calculation amount and the number of parameters have been improved, they also gain accuracy. Overall, the development of SPP enables target detection networks to be more flexible and efficient in handling targets at different scales, laying the foundation for subsequent improvement and development.

2.3 Feature pyramid network

Feature pyramid is a crucial component utilized in various fields such as object detection, semantic segmentation, behavior recognition, etc. It plays a significant role in enhancing the performance of models. Prior to the introduction of Feature Pyramid Network [12], SSD [28] directly employs feature maps from different stages to detect objects of varying scales. FPN is designed as a top-down unidirectional fusion mechanism incorporating features extracted from the model's backbone. In this process, due to the limitation of one-way feature fusion. PANet [29] adds a bottom-up path based on FPN to enable deep features to obtain detailed information in shallow features. BiFPN [30] is an advanced iteration of FPN. It further enhances its performance by eliminating nodes with a single input edge and introducing additional edges from the original input at the same level. Zhang et al. [31] propose a novel quad feature pyramid network (Quad-FPN) for SAR ship detection. Made up of four unique FPNs, They are Deformable Convolutional FPN, Content-Aware Feature Reassembly FPN, Path Aggregation Space Attention FPN, and Balance Scale Global Attention FPN. Generalized-FPN (GFPN) [32] introduces an innovative cross-scale connection method known as "queen-fusion", which effectively incorporates hierarchical features from preceding and current layers. Additionally, the $\log_2 n$ skip-layer connections are integrated to facilitate enhanced information transmission and enable the scaling of deeper networks with greater effectiveness. Xu et al. [33] propose a new group-wise feature enhancement-and-fusion network (GWFEFNet) with dual-polarization feature enrichment. It contains four key modules: dual-polarization feature enrichment, group-wise feature enhancement, group-wise

feature fusion, and hybrid pooling channel attention. This leads to better dual-polarized SAR ship detection. DOMO-YOLO [23] builds upon the foundation of GFPN and introduces an enhanced variant known as Efficient-RepGFPN, which empowers real-time object detection. Recently, the Asymptotic Feature Pyramid Network (AFPN) [34] has broken through the pattern of conventional FPN to avoid significant semantic gaps between non-adjacent levels. AFPN initiates fusion in the first phase for backbone bottom-up features by combining two shallow features at different scales. As we enter the later stage, the deep features are gradually integrated into the fusion process, and finally the complete fusion of the top features of the backbone is achieved.

2.4 UAV aerial images object detection methods

In recent years, the field of small object detection has garnered significant research interest, and numerous scholars have made notable advancements in this domain. Deep learning-based UAV object detection techniques are evaluated by Saqib et al. [35], who use the migration learning method to train a pre-trained model for the network with sparse training samples. Chen et al. [36] have incorporated adaptive resampling techniques and regression modules into their RRNet model. These integrations offer superior data augmentation and precise bounding boxes, effectively tackling the intricacies of detecting diminutive objects within dense environments. Khan et al. [37] propose a framework for satellite images with complex backgrounds, arbitrary viewpoints, and significant variations in object size. The framework comprises two phases: the first generates multi-scale object proposals, and the second categorizes each proposal into different classes. Furthermore, in GDF-Net [38], dilated convolutions are employed to refine density features, thereby broadening the network's receptive field. This refinement bolsters the model's efficacy and resilience. Tian et al. [39] introduce a double neural network verification approach, which secondarily identifies overlooked target regions, ensuring exceptional detection quality for small targets. DMF [40] model is a detection method based on difference depth, which solves the problem of low accuracy of long-distance small-object traffic detection by clustering the difference maps with different depths and mapping the different difference regions to two-dimensional candidate regions. Li et al. [41] propose a novel multi-scale detection network to reduce the redundant information transfer between scales. The network divides objects according to their distance from the viewpoint. A multi-branch architecture is constructed to provide specialized detection for each scale of objects separately. Ma et al. [42] propose an UAV tracking control algorithm based on incremental reinforcement learning. The algorithm achieves proper exploration and efficient learning in new environments by transforming into a Markov decision process and applying policy mitigation and importance weighting methods. Zhang et al. [43] develop an adaptive and dense pyramid network to address multi-scale challenges in UAV aviation images. The network integrates a pyramid density module and a target detection module to align density information and instance recognition features. This alignment improves network performance and detection accuracy. In PETNet [44], a novel Prior Enhanced Transformer (PET) module and One-to-Many Feature Fusion (OMFF) mechanism are introduced. The PET module is designed to capture enhanced global information, while the OMFF mechanism fuses multiple features. These advancements contribute to improved detection performance.

3 DoubleM-Net

In this section, we thoroughly explain the DoubleM-Net model, including the process and setup information, as well as the relevant algorithms and expressions. We describe the MS-SPPF, ASFF, MPA-FPN, and the loss function, etc. The complete network architecture of the proposed DoubleM-Net can be seen in Fig. 2.

3.1 Extracting multi-resolution features

The aerial images captured by drones are first fed into the backbone network for feature extraction. In the first convolution layer of the backbone network, a 3×3 kernel is used with smaller receptive fields to extract low-level features from the images, such as edges, textures, and other detailed features. As the number of network layers increases, the convolution layers in the backbone network will gradually downsample the images, using different-sized receptive fields to perform convolution operations to extract features of different scales. They cover more expansive areas and can extract higher-level features, forming more abstract feature representations such as object parts, shapes, etc. These features contain more global information and provide rich and varied features for subsequent feature fusion. In our proposed model, the backbone networks of DoubleM-Net-p6 extract multi-scale features from images, which is denoted as $\{C1, C2, C3, C4\}$ as shown in Fig. 2.

3.2 Multi-scale spatial pyramid pooling-fast

Figure 3 shows the structures of SPP and SPPF in (a) and (b), respectively, while (c) shows our proposed MS-SPPF. In SPP, several pooling kernels of different sizes $k = [5, 9, 13]$ capture spatial features at various scales. In contrast, SPPF, an evolved version of SPP, uses a single pooling kernel $k = 5$ and simulates the effect of multi-scale

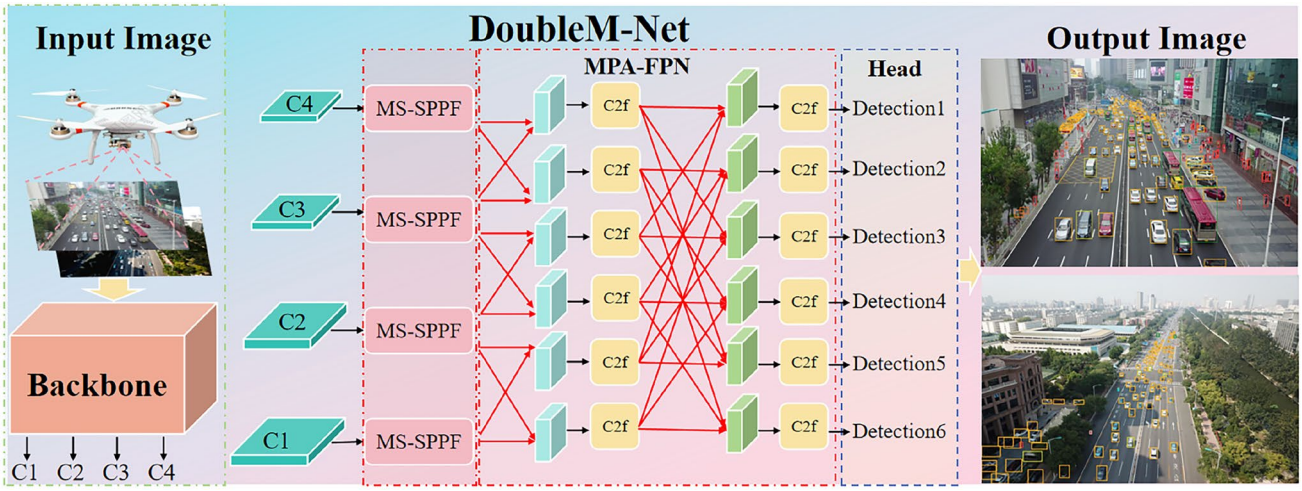

Fig. 2 The architecture of our proposed DoubleM-Net. It includes the backbone network extracting multi-scale feature graphs $\{C1, C2, C3, C4\}$ for air-to-ground images, MS-SPPF enhances the adaptability of the model to scale change, and MPA-FPN better allows high-low-level semantic information to interact and six detection heads

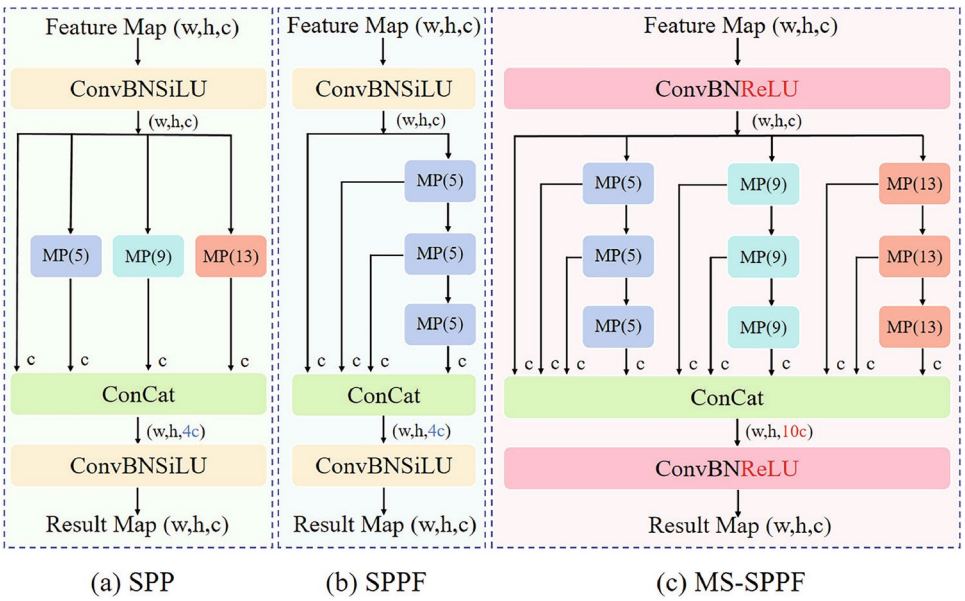

Fig. 3 The structure of the spatial pyramid pooling. **c** is our proposed multi-scale spatial pyramid pooling-fast module (MS-SPPF)

feature extraction by applying this pooling kernel multiple times. MS-SPPF fuses the ideas of SPP and SPPF by using numerous pooling kernels with different sizes $k = [5, 9, 13]$ and applying multiple times to each pooling kernel to enrich multi-scale feature extraction further. First, MS-SPPF performs an initial transformation of the input feature maps through a SimConv layer. Subsequently, the original feature maps are spliced with the feature maps processed by different pooling kernels in the channel dimension to integrate the multi-scale information. Finally, the spliced feature maps are again passed through a SimConv layer for further feature extraction and integration to generate the final output feature maps.

The backbone network extracts features at different scales in a bottom-up fashion. MS-SPPF enhances computational efficiency through SimConv and captures richer small objects and multi-scale features through repeated multi-scale pooling operations. Specifically, we apply SimConv to convolute the feature vector $x \in \mathbb{R}^{(c1,w,h)}$ extracted from the backbone network to obtain:

$$x1 = SimConv(x) \in \mathbb{R}^{(\tilde{c},w,h)}, \tag{1}$$

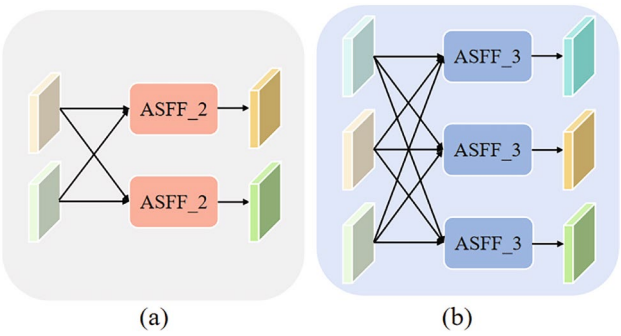

Fig. 4 Adaptive spatial feature fusion procedure. **a** Feature fusion at two different resolutions; **b** feature fusion at three different resolutions; but we can use the method with more levels as needed

where $\tilde{c} = c1/2$. Compared with the traditional convolutional, SimConv adopts ReLU as the default activation function, and the rest remains unchanged.

Then maxpooling is performed on $x1$ with kernel sizes of $k = 5, 9, 13$, respectively. Three pooling operations are conducted on each branch to obtain feature maps of different scales. This results in features with different receptive fields:

$$\begin{cases} p_i = mp_k(x1), \\ q_i = mp_k(p_i), \\ r_i = mp_k(q_i), \end{cases} \tag{2}$$

where $i = 1, 2, 3$ and mp is maxpooling operation. Finally, all the pooled features p_i, q_i, r_i and the retaining original information $x1$ are concatenated, then the SimConv operation is performed to obtain the feature vector y:

$$y = SimConv(Concat(p_i, q_i, r_i, x1) \in \mathbb{R}^{(c2,w,h)}. \tag{3}$$

The detailed procedure for the MS-SPPF structure is shown in Algorithm 1.

Algorithm 1 MS-SPPF algorithm

Require: Input feature map (x), shape $[c1, h, w]$;
Ensure: Output feature map (y), shape $[c2, h, w]$, where $c2, h, w$ are channel, height and width;
1: Pass input x through convolutional layer, output is $x1$, shape is $[\tilde{c}, h, w]$ where $\tilde{c} = c1/2$;
2: Define three maxpooling layers with kernel sizes $k = (5, 9, 13)$ and stride 1, padding $k/2$;
3: **for** $i = k$ **do**
4: The maxpooling layer is applied three times on $x1$, with one feature vector output for each pooling operation;
5: Obtain nine outputs, namely $p_i, q_i, r_i, i = 1, 2, 3$;
6: **end for**
7: Concatenate $p_i, q_i, r_i, x1$ along channel dimension, output is pooled_features, shape is $[\tilde{c} \times 10, h, w]$;
8: Pass pooled_features through convolutional layer, output feature map y with shape $[c2, h, w]$;
9: **return** Output feature map (y)

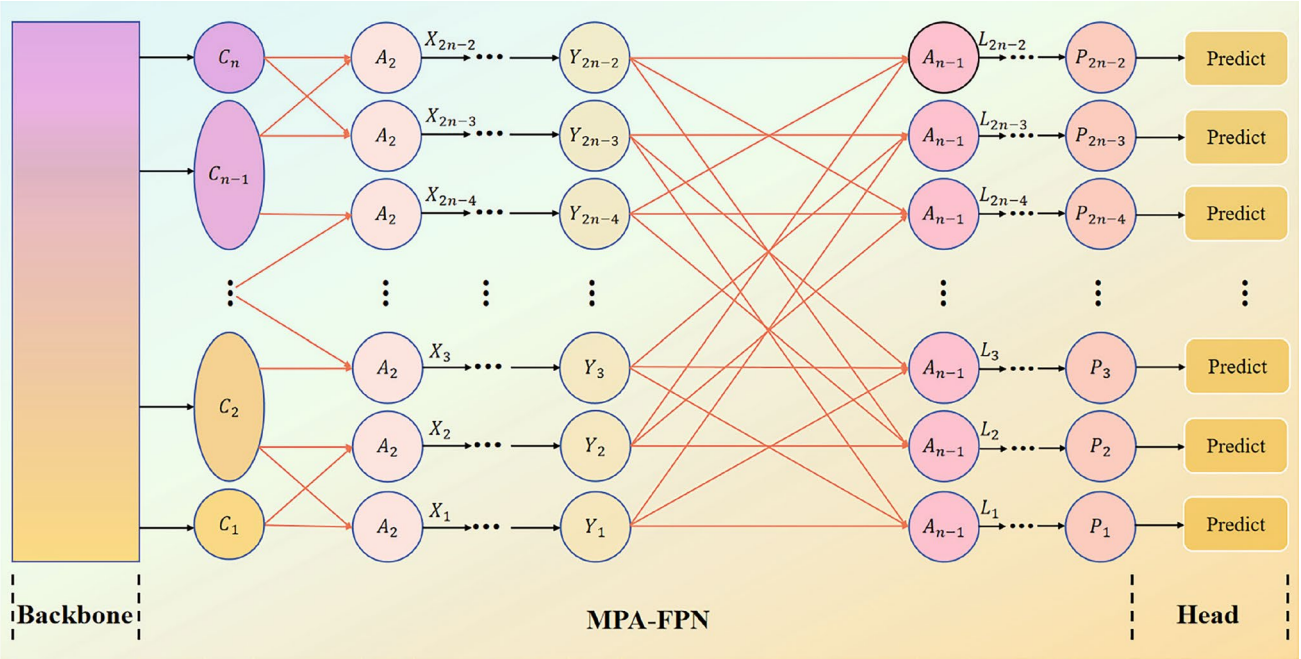

Fig. 5 The framework of our proposed MPA-FPN. For n multi-scale features, feature interaction is performed level by level, finally generating $2n-2$ detection feature maps

3.3 Adaptive spatial feature fusion

In the field of image processing and computer vision, multi-scale feature fusion has been a research direction that has attracted much attention. With the rapid progress of deep learning technology, how to efficiently integrate feature information of different scales to enhance the performance of complex tasks such as target detection and image segmentation has become a hotspot that researchers are competing to explore. Feature pyramid representation, as a typical means to solve the problem of scale variation in object detection, still has obvious limitations despite certain achievements. In particular, the inconsistency between different feature scales becomes a significant challenge for feature pyramid-based single-shot detectors. To solve this problem, Liu et al. [45] propose the adaptive spatial feature fusion (ASFF) method. ASFF effectively suppresses the inconsistency between different feature scales by learning spatial filtering of conflicting information, which significantly improves feature-scale invariance.

We introduce ASFF to fuse features from different spatial scales or levels to improve the performance of image analysis and understanding. During multi-scale feature fusion, ASFF assigns different spatial weights to features at different scales, enhancing the importance of key levels and alleviating the influence of information from features across scales.

Let $x_{ij}^{m\longrightarrow n}$ denote the feature vector at position (i, j) from level m to level n. The resulting feature vector is denoted as y_{ij}^{n}, obtained by adaptive spatial fusion of multi-scale features, defined by a linear combination of feature vectors $x_{ij}^{1\longrightarrow n}, x_{ij}^{2\longrightarrow n}, \ldots, x_{ij}^{m\longrightarrow n}$ as follows:

$$y_{ij}^{n} = \alpha_{1_{ij}}^{n} \cdot x_{ij}^{1\longrightarrow n} + \alpha_{2_{ij}}^{n} \cdot x_{ij}^{2\longrightarrow n} + \cdots + \alpha_{m_{ij}}^{n} \cdot x_{ij}^{m\longrightarrow n}, \tag{4}$$

where $\alpha_{1_{ij}}^{n}, \alpha_{2_{ij}}^{n}, \ldots, \alpha_{m_{ij}}^{n}$ represent the spatial weights of the m different scale features at level n, respectively, subject to the constraint by:

$$\alpha_{1_{ij}}^{n} + \alpha_{2_{ij}}^{n} + \cdots + \alpha_{m_{ij}}^{n} = 1; \alpha_{1_{ij}}^{n}, \alpha_{2_{ij}}^{n}, \ldots, \alpha_{m_{ij}}^{n} \in [0, 1]. \tag{5}$$

As shown in Fig. 4a and b, we integrated features from two and three scales, corresponding to the cases when $m = 2$ and $m = 3$ in Eq. 4. Considering the differences in the number of fused features at each stage, it is possible to implement an adaptive spatial fusion module for a specific stage based on the actual situation.

3.4 Multi-path adaptive feature pyramid network

The paradigm framework of MPA-FPN is shown in Fig. 5. Like many feature pyramid network based object detection methods, multi-scale features are extracted from the

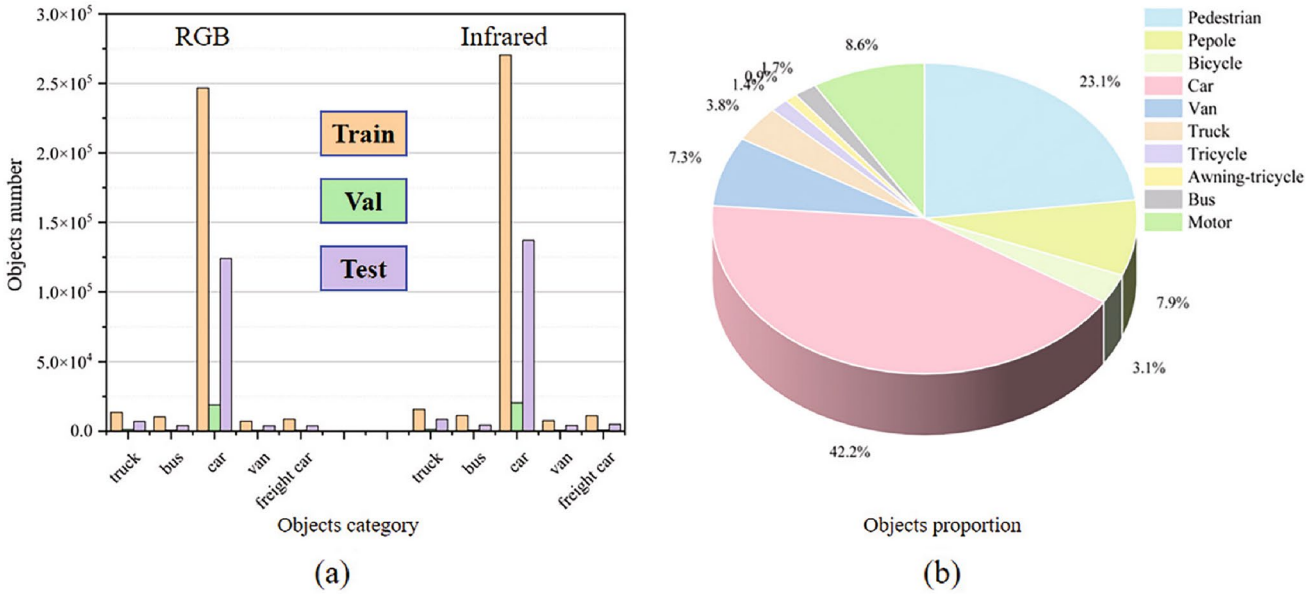

Fig. 6 **a** is the number of category labels in the training, validation and test datasets in the RGB and Infrared modes in the DroneVehicle dataset. **b** is the proportion of labels in a category in the VisDrone dataset

backbone before feature fusion. The last layer features are extracted from each feature level of the backbone to obtain a set of multi-scale features denoted as $\{C_1, C_2, \ldots, C_n\}$.

For feature fusion, a set of features of different scales obtained in the previous step, each of the two adjacent features is integrated into two levels of adaptive spatial, generating X_{2n-2} feature representations:

$$\begin{aligned}&\{X_1, X_2, \ldots, X_{2n-2}\}\\&\quad = \{A_2(C_1, C_2), A_2(C_2, C_1), \ldots, A_2(C_n, C_{n-1})\},\end{aligned} \tag{6}$$

where A_2 is the case of the adaptive spatial teature fusion operation shown in Fig. 4a and n = 2 in Eq. 4. Then extract the $\{X_1, X_2, \ldots, X_{2n-2}\}$ feature to get Y_i:

$$Y_i = Blackbox(X_i), i = 1, 2, \ldots, 2n-2, \tag{7}$$

where *Blackbox* is a series of feature extraction operations, such as convolution, *C*3 and *C*2*f*, or an adaptive spatial feature fusion at the next level. Next, the feature fusion at level $n-1$ is performed on $Y_i (i = 1, 2, \ldots, 2n-2)$ to obtain feature maps $L_i (i = 1, 2, \ldots, 2n-2)$:

$$\begin{aligned}\{L_1, L_2, \ldots, L_{2n-2}\} = \{&A_{n-1}(Y_1, Y_3, \ldots, Y_{2n-3}),\\&A_{n-1}(Y_2, Y_4, \ldots, Y_{2n-2}), \ldots,\\&A_{n-1}(Y_{2n-2}, Y_{2n-4}, \ldots, Y_2)\}.\end{aligned} \tag{8}$$

It is obvious that $\{L_1, L_2, \ldots, L_{2n-2}\}$ contains all the features in $\{C_1, C_2, \ldots, C_n\}$, but it is not directly fused. Due to the semantic gap between non-adjacent hierarchical features being greater than between adjacent hierarchical features, especially for the bottom and top features, directly fusing non-adjacent hierarchical features leads to poor fusion effects. Therefore, we first fuse adjacent features from different scales, then gradually fuse the features in steps, and finally generate feature maps rich in semantic information. Some operations on the feature representations $\{L_1, L_2, \ldots, L_{2n-2}\}$ before detection finally yield P_i:

$$P_i = f(L_i), i = 1, 2, \ldots, 2n-2, \tag{9}$$

where f is the method to perform the feature extraction, generating $2n-2$ feature maps rich in high- and low-level semantic information. The detailed procedure of MPA-FPN is shown in Algorithm 2.

Table 1 Experimental parameters

Parameters	Value
Image size	640 × 640
Batch size	16
Mosaic	1.0
Fliplr	0.5
HSV-H	0.015
HSV-S	0.7
HSV-V	0.4
Learning rateα	0.001
Weight decay	0.0005
Warmup epochs	3.0
Momentum	0.937
Warmup momentum	0.8

These are specific variables that remain unchanged throughout the experiment

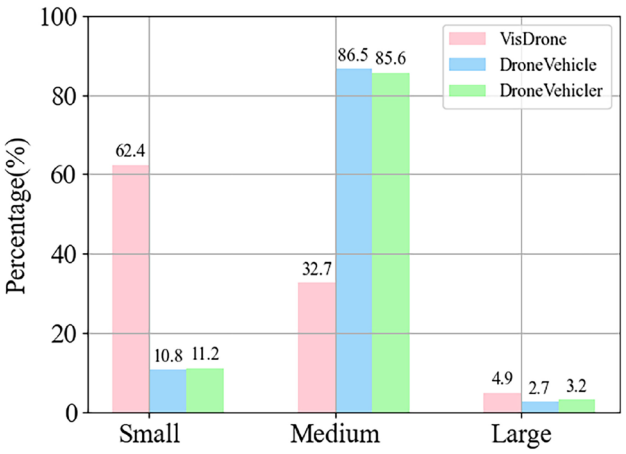

Fig. 7 VisDrone and DroneVehicle (RGB and infrared) datasets comprise objects categorized based on their sizes into small objects (area $< 32 \times 32$), medium objects ($32 \times 32 <$ area $< 96 \times 96$), and large objects (area $> 96 \times 96$)

Algorithm 2 MPA-FPN algorithm

Require: Enter the feature $\{C_1, C_2, \cdots, C_n\}$ extracted from the backbone network.;

Ensure: Get predictive feature maps $\{P_1, P_2, \cdots, P_{2n-2}\}$;

1: Deliver $\{C_1, C_2, \cdots, C_n\}$ to the MPA-FPN network;
2: Carry out adaptive spatial feature fusion at two levels, as illustrated in Equation 6. Gets a set of feature representations $\{X_1, X_2, \cdots, X_{2n-2}\}$.
3: As shown in Equation 7, the next level of spatial adaptive feature fusion or feature extraction is performed on X_i to obtain feature Y_i, where $i = 1, 2, \cdots, 2n-2$.
4: Perform feature fusion at level $n-1$ on $\{Y_1, Y_2, \cdots, Y_{2n-2}\}$ using Equation 8 to obtain $\{L_1, L_2, \cdots, L_{2n-2}\}$.
5: The predictive feature maps $P_i (i = 1, 2, \cdots, 2n-2)$ is obtained by Equation 9.
6: **return** Output feature maps $\{P_1, P_2, \cdots, P_{2n-2}\}$

3.5 Loss function

Considering the characteristics of images from the UAV viewpoint, which often contain many small targets. We design six detection heads to improve the detection accuracy of these small targets. DoubleM-Net achieves accurate target detection by using the decoupling head to detect different scales of feature maps generated by the neck network. The decoupling head is delicately conceived to decompose the detection task into two mutually independent branches: one specializes in classification prediction to identify the target class accurately. In contrast, the other branch focuses on regression prediction to accurately locate the target's position.

The loss function plays a crucial role in model training by quantifying the difference between the model predictions and the actual values, providing a clear guideline for model optimization. For the DoubleM-Net model, the loss calculation covers classification loss and regression loss. The classification loss is calculated using the binary cross entropy (BCE) loss function to ensure the model's accuracy in the classification task. In contrast, the regression loss combines the complete IoU (CIoU) loss and the distribution focus loss (DFL), further improving the model's target localization accuracy. The BCE loss function is defined as shown in Eq. (10),

Fig. 8 The first two rows and the last two rows are some visualization results of DoubleM-Net(x) on the VisDrone-DET-test-dev and VisDrone-DET-test-challenge datasets, respectively

$$\mathcal{L}_{BCE} = \frac{1}{n}\sum_{i=1}^{n} -\left[y_i \cdot \log_e(p_i) + (1-y_i)\cdot \log_e(1-p_i)\right], \quad (10)$$

where y_i denotes the label value of the i-th sample, which takes the value of 0 or 1. p_i denotes the predicted probability of the ith sample. Then the CIoU loss function is calculated by Eq. 11,

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b^p, b^{gt})}{c^2} + \varepsilon\nu, \quad (11)$$

where ε is the weight coefficient. b^p and b^{gt} represent the centroids of the predicted and actual boxes. ρ is the Euclidean distance calculated between the two centroids, and c denotes the diagonal distance between the closed regions of the two rectangular frames. ν is used to measure the similarity of the aspect ratios and is defined as in Eq. 12,

$$\nu = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w^p}{h^p}\right)^2, \quad (12)$$

where (w^{gt}, h^{gt}) and (w^p, h^p) are the width and height of the actual and predicted boxes, respectively. Finally, Class imbalances in data sets are a common challenge. This can cause the model to favor more numerous categories over less numerous ones during training. To mitigate this problem, we introduce distributed focus loss (DFL) to optimize the classification task, as shown in Eq. 13,

$$\mathcal{L}_{DFL}(F_i, F_{i+1}) = -\left[\left(y_{i+1} - y\right)\log\left(F_i\right) + \left(y - y_i\right)\log\left(F_{i+1}\right)\right], \quad (13)$$

where y is the target label. The global minimum solution of DFL, i.e, $F_i = \frac{y_{i+1}-y}{y_{i+1}-y_i}$, $F_{i+1} = \frac{y-y_i}{y_{i+1}-y_i}$, can guarantee the estimated regression target $\bar{y}$ infinitely close to the corresponding label y. DFL helps the model focus more quickly and accurately on the output distribution near the accurate label by explicitly enlarging the two probability values y_i and y_{i+1} adjacent to the target label y. This allows the model to give proper attention to the categories even when unbalanced.

The overall training loss is a weighted combination of these three losses, as shown in Eq. 14,

$$\mathcal{L}_{sum} = \lambda_1\mathcal{L}_{BCE} + \lambda_2\mathcal{L}_{CIoU} + \lambda_3\mathcal{L}_{DFL}. \quad (14)$$

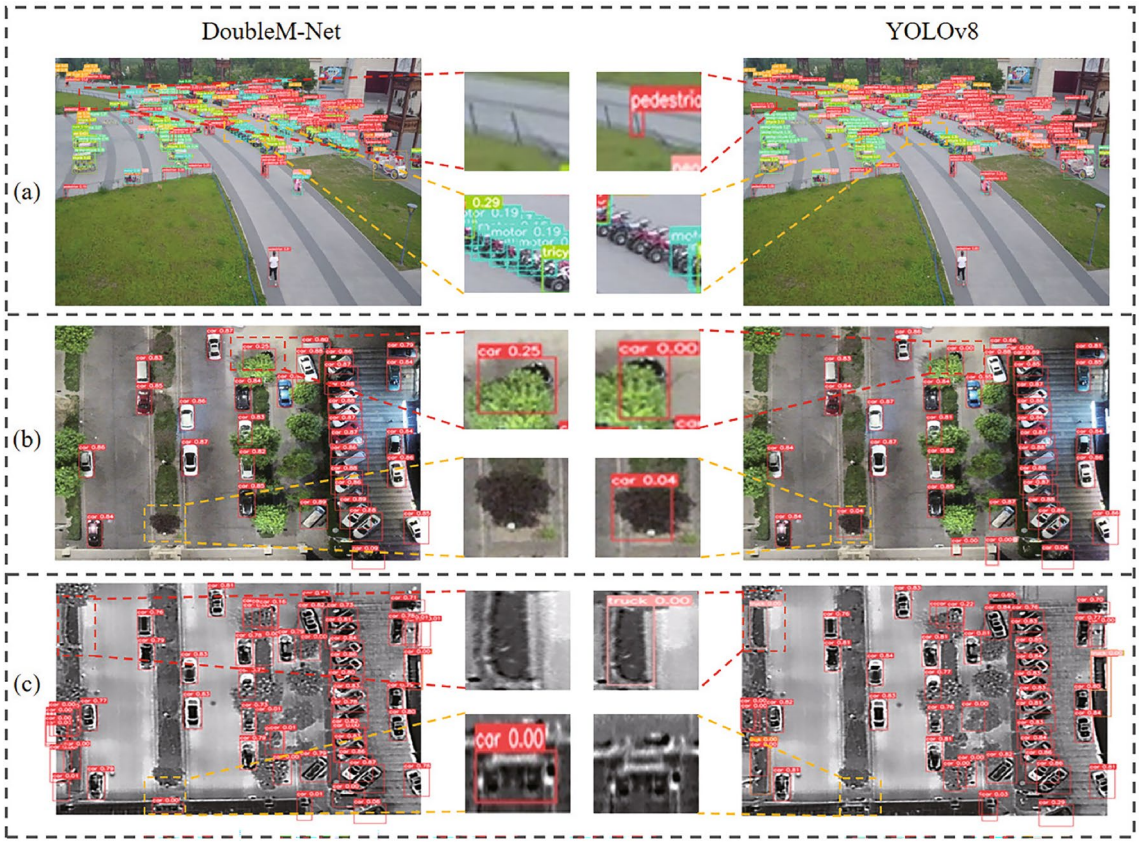

Fig. 9 We compare the effects of DoubleM-Net and YOLOv8, where the red and green dashed boxes indicate some differences in the detection effects of these two models. **a** is the VisDrone dataset; **b** and **c** the RGB and infrared patterns of the DroneVehicle dataset, respectively. It is seen from the figure that there are some omissions in YOLOv8, but DoubleM-Net can be detected

4 Experiments

In this section, we provide a comprehensive overview of the implementation steps and conduct a detailed analysis of the results obtained from our experiment. We evaluate the object detection performance using standard metrics, including average precision (AP) and the mean average precision (mAP). To present these results, we utilize graphs and tables. The experimental data presented in this section offer valuable insights into the performance and effectiveness of our proposed model.

4.1 Dataset and analysis

The VisDrone [46] and DroneVehicle [47] datasets contain many annotated UAV-view images and videos, providing strong support for algorithm performance evaluation and optimization. These datasets specifically focus on target detection and tracking in real-world environments with UAV viewpoints, providing an ideal testbed for researchers. By covering a wide range of environments, lighting conditions, and UAV models, these datasets are closer to real-world application scenarios, which helps to improve the algorithm's generalization ability in real-world applications. In existing research, VisDrone and DroneVehicle datasets have been widely used in various fields such as intelligent transportation, disaster response, urban planning, etc., providing essential data support for developing related applications.

The VisDrone dataset is a comprehensive benchmark designed explicitly for visual object detection and tracking in UAV platforms. It consists of images captured from various UAV platforms across 14 cities in China. The dataset provides ten categories: pedestrian, people, bicycle, car, truck, tricycle, awning-tricycle, bus, and motor, and their instance proportions, as shown in Fig. 6b. These images showed a high object density, averaging 53 instance objects per image. This dataset contains 8629 images, of which 6471 are used for training, 548 for validation, and 1610 for testing.

The DroneVehicle dataset comprises 56,878 images collected by the drone. Out of this total, 50% are RGB images, while the remaining 50% are infrared images. This dataset has five categories: car, truck, bus, van, freight car. The number of objects in the train, validation, and test dataset for

Table 2 We compare the detection effects of some classical single-and two-stage networks across ten classes on the VisDrone dataset

Model	Ped	Peo	Bic	Car	Van	Truck	Tri	A-Tri	Bus	Mo	mAP50-95
FSAF [48]	22.1	14.1	5.6	53.8	29.3	21.5	8.9	5.8	30.9	16.5	20.9
ATSS [49]	19.7	6.5	7.4	54.4	31.0	24.5	14.2	8.7	37.0	18.1	22.1
TridentNet [50]	16.9	10.5	5.9	50.8	28.8	22.4	14.4	7.1	33.0	16.8	20.7
CenterNet [51]	17.3	10.5	5.6	48.3	26.3	18.5	8.4	5.5	30.9	15.5	18.7
FCOS [52]	17.3	9.3	3.3	51.3	26.7	22.5	8.6	7.0	34.1	9.6	19.0
DDOD [53]	21.9	11.9	7.4	55.3	31.4	25.4	14.5	8.6	37.2	19.8	23.3
TOOD [54]	21.9	13.0	8.6	56.2	33.0	26.0	16.1	9.1	38.8	21.4	24.4
VFNet [55]	20.6	9.1	6.7	55.3	32.5	25.3	14.7	8.3	39.0	19.2	23.1
Cascade-RCNN [56]	19.9	12.3	8.4	54.1	35.3	26.4	17.4	9.2	42.2	19.6	24.5
Faster-RCNN [11]	17.6	12.0	7.2	50.5	30.1	23.3	14.4	8.9	37.2	18.2	21.9
YOLOX [57]	19.6	14.3	6.6	53.4	28.6	22.7	14.7	8.0	34.4	21.0	22.2
YOLOv3 [15]	22.0	13.7	7.1	53.9	30.4	25.0	14.9	7.6	39.8	18.8	23.3
YOLOv4 [16]	21.0	13.3	6.0	**59.1**	34.4	30.5	17.1	11.0	44.5	20.0	25.7
YOLOv5(x) [17]	**23.4**	15.4	7.8	58.9	34.6	**30.9**	19.7	12.4	**46.5**	23.7	27.3
YOLOv7 [19]	24.4	**18.1**	8.4	57.5	34.2	28.1	19.2	11.2	42.9	**24.9**	26.9
YOLOv8(x) [20]	23.1	15.5	7.9	58.9	**35.4**	30.4	19.8	12.8	44.9	23.7	27.2
YOLOv8-p6(n) [20]	12.6	8.7	1.9	50.0	24.0	15.7	9.9	6.8	28.2	13.5	17.1
DoubleM-Net-p6(n)	14.0	10.0	3.2	51.9	26.2	17.7	12.3	8.0	32.0	15.1	19.0
YOLOv8-p6(s) [20]	17.5	11.3	4.2	55.0	29.9	22.3	14.1	9.8	38.4	17.6	22.0
DoubleM-Net-p6(s)	18.1	12.9	5.6	54.6	30.1	23.2	15.8	10.0	41.8	19.8	23.2
YOLOv8-p6(m) [20]	20.6	13.3	6.0	57.3	32.9	25.9	18.1	11.6	42.8	21.2	25.0
DoubleM-Net-p6(m)	20.9	14.7	7.0	57.3	32.2	27.7	19.6	11.8	45.2	22.8	25.9
YOLOv8-p6(l) [20]	22.4	14.3	7.2	58.2	33.5	29.2	18.9	12.1	45.7	23.3	26.5
DoubleM-Net-p6(l)	21.9	15.2	8.0	58.2	33.8	29.2	20.2	12.9	44.8	23.8	26.8
YOLOv8-p6(x) [20]	22.8	15.4	8.0	58.6	35.2	30.8	20.6	12.2	44.8	23.2	27.2
DoubleM-Net-p6(x)	22.6	16.1	**8.8**	58.6	34.0	28.3	**22.1**	**13.7**	46.4	24.7	**27.5**

Results in the *n*, *s*, *m*, *l*, and *x* modes of YOLOv8-p6 and DoubleM-Net-p6 are also shown

both RGB and infrared images is displayed in Fig. 6a. The images in the DroneVehicle dataset are divided into three scenarios: day, night, and dark night, with 14,478, 5,468 and 8493 images, respectively. Since the infrared images have a higher contrast in low-light conditions, they have more annotation than the RGB images, as evident in Fig. 6a. The number of photos in this dataset for training, validation, and testing the dataset in RGB and infrared images is 17,990, 1469, and 8980, respectively. These images show a high density of objects, with an average of 17 instance objects per image, with a maximum number of 206.

As shown in Fig. 7, the VisDrone and DroneVehicle datasets have more than 95% of small (area $< 32 \times 32$) and medium ($32 \times 32 <$ area $< 96 \times 96$) objects. On the contrary, large (area $> 96 \times 96$) targets account for less than 5%. This data distribution score reflects the challenges of UAV target detection in practical applications, especially the urgent need for small and dense target detection. Therefore, these two datasets not only enrich the data resources in the field of UAV visual inspection but also provide strong support for the optimization of algorithms and practical applications, which is of great significance in promoting the development of UAV visual inspection technology.

4.2 Experimental parameters setting

This section presents an overview of the experimental parameters employed in our study. The experiments are performed on a system equipped with an NVIDIA GeForce RTX 3090 GPU and a 15 vCPU Intel(R) Xeon(R) Platinum 8358P CPU @ 2.60GHz processor. Table 1 summarizes the critical parameters for the DoubleM-Net model. All models in this paper are trained from scratch on the VisDrone and DroneVehicle datasets, with each model trained for 100 epochs.

4.3 Evaluation criterion

The primary evaluation metric used in object detection is the average accuracy (AP), which is calculated based on four possible outcomes: true positive (TP), false positive

Table 3 The following table shows the effects of the *n*, *s*, *m*, *l* and *x* models of YOLOv5, YOLOv8 and DoubleM-Net on the VisDrone dataset, and the experimental results show the detection accuracy of our model for small targets

Model	Ped	Peo	Bic	Car	Van	Truck	Tri	A-Tri	Bus	Mo	mAP50
YOLOv5(n)	29.1	23.2	5.7	72.4	33.3	22.7	18.1	9.3	38.5	31.1	28.3
YOLOv8(n)	30.6	24.8	5.6	73.5	35.6	24.5	19.4	10.1	41.9	32.6	29.9
DoubleM-Net(n)	32.3	27.4	8.9	75.1	37.4	28.5	21.6	12.4	47.3	36.5	32.7
Improvement	**+1.7**	**+2.6**	**+3.2**	**+1.6**	**+1.8**	**+4.0**	**+2.2**	**+2.3**	**+5.4**	**+3.9**	**+2.8**
YOLOv5(s)	38.6	29.6	9.4	78.1	42.6	32.4	24.2	14.5	49.7	40.2	35.9
YOLOv8(s)	39.7	31.0	10.3	76.4	40.8	32.8	24.5	14.0	51.5	40.9	36.2
DoubleM-Net(s)	41.4	34.1	13.9	77.5	42.7	35.8	30.4	17.4	56.2	45.4	39.5
Improvement	**+1.7**	**+3.1**	**+3.6**	−0.6	**+0.1**	**+3.0**	**+5.9**	**+2.9**	**+4.7**	**+4.5**	**+3.3**
YOLOv5(m)	44.1	33.9	13.2	80.4	45.9	37.3	27.8	16.9	57.7	46.2	40.3
YOLOv8(m)	44.5	35.1	14.2	78.8	46.0	38.7	31.1	18.2	58.6	46.8	41.2
DoubleM-Net(m)	45.0	36.9	15.9	81.3	45.9	42.1	35.1	21.0	62.1	49.6	43.5
Improvement	**+0.5**	**+1.8**	**+1.7**	**+0.9**	−0.1	**+3.4**	**+4.0**	**+2.8**	**+3.5**	**+2.8**	**+2.3**
YOLOv5(l)	46.7	36.6	15.1	80.1	47.2	42.2	34.1	18.2	63.0	48.0	43.1
YOLOv8(l)	46.4	36.4	15.7	79.8	48.1	42.6	34.7	18.9	60.9	48.8	43.2
DoubleM-Net(l)	46.1	37.7	18.3	79.6	47.4	42.6	35.0	21.1	62.5	50.0	44.0
Improvement	−0.6	**+1.1**	**+2.6**	−0.5	−0.7	=	**+0.3**	**+2.2**	−0.5	**+0.2**	**+0.8**
YOLOv5(x)	48.8	37.5	17.7	80.5	48.5	45.1	34.2	19.9	60.9	50.2	44.3
YOLOv8(x)	48.4	37.6	17.2	80.8	49.0	44.3	34.9	19.8	61.9	50.3	44.4
DoubleM-Net(x)	48.1	38.5	19.0	80.0	48.2	43.0	35.4	21.5	62.4	51.6	44.8
Improvement	−0.7	**+0.9**	**+1.3**	−0.8	−0.8	−2.1	**+0.5**	**+1.6**	**+0.5**	**+1.3**	**+0.4**

Ped, Peo, Bic, Tri, and A-Tri are the abbreviations for Pedestrian, People, Bicycle, Tricycle, and Awning-Tricycle, respectively

The underline highlights the performance gap between our model and the best results

(FP), true negative (TN), and false negative (FN). The classification of these outcomes is determined by the predicted category of the detection model and the actual category of the object being detected. The AP metric provides valuable insights into the performance and accuracy of object detection. The precision rate is calculated using the following equation:

$$Precision = \frac{TP}{TP + FP}. \tag{15}$$

The recall rate is defined as:

$$Recall = \frac{TP}{TP + FN}. \tag{16}$$

The AP metric plays a crucial role in assessing the effectiveness of a learned model for each category. The formula for calculating AP is shown in Eq. 17.

$$AP = \int_0^1 Precision(Recall) d(Recall), \tag{17}$$

where $P(R)$ is a curve based on recall and precision. The AP value indicates the performance of the model in a particular category. On the other hand, mAP is the average of all AP values in all categories. It provides an assessment of the overall learning performance of the model and can be defined as Eq. 18,

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP(i), \tag{18}$$

where n is the total number of classes or categories.

4.4 Main results

This section aims to verify the effectiveness of the proposed DoubleM-Net. We conduct model training and validation from scratch using the Visdrone and DroneVehicle datasets to accomplish this. All experimental results based on YOLOv5 in this paper are conducted under the framework of YOLOv8.

4.4.1 Experimental results on VisDrone dataset

To validate the object detection performance of DoubleM-Net technology in UAV scenarios, we compare several classical single-stage and two-stage object detection methods

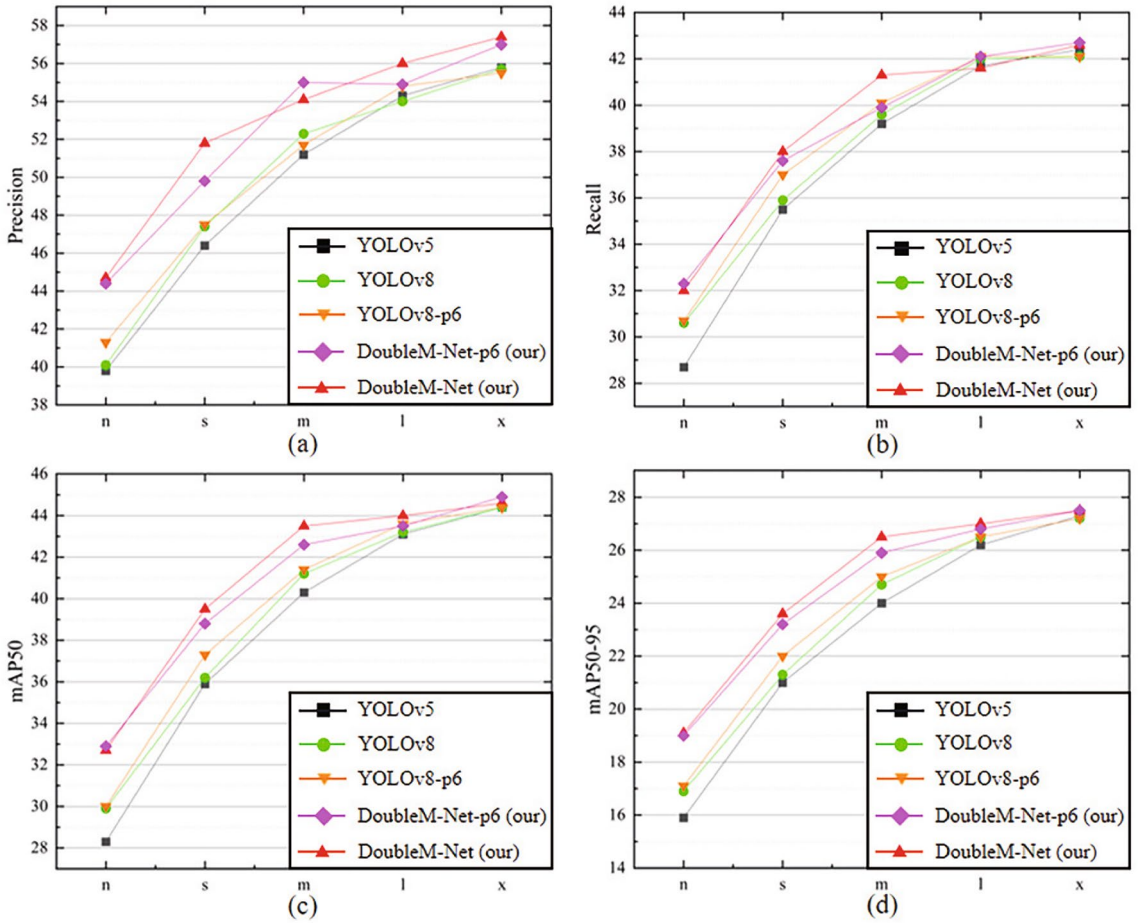

Fig. 10 **a–d** are the Precision, Recall, mAP 50, and mAP 50-95 for YOLOv5, YOLOv8, and DoubleM-Net on the VisDrone dataset, respectively. It can be seen that our model achieves significant results on *n*, *s*, *m*, *l*, and *x*

on the VisDrone dataset. It is worth noting that all models are trained from scratch without utilizing any pre-trained weights. The detection results of DoubleM-Net on the VisDrone dataset are shown in Fig. 8. The first two rows depict the visualization of detection results on the VisDrone-DET-test-dev dataset. In comparison, the last two rows represent the results on the VisDrone-DET-test-challenge dataset. From the figure, it can be observed that DoubleM-Net can accurately recognize and localize medium to large vehicles under different lighting and weather conditions. It also achieves satisfactory detection results for smaller objects that can be distinguished by the human eye, such as pedestrians and motors. DoubleM-Net also competes in challenging scenarios with targets occluded and densely areas. Figure 9a compares the detection performance between DoubleM-Net and YOLOv8 on the VisDrone dataset. The image shows that YOLOv8 fails to detect the densely packed motors indicated by the orange dashed box, while DoubleM-Net accurately recognizes them. Furthermore, the red dashed box highlights a false detection produced by YOLOv8, which DoubleM-Net avoids.

Table 2 presents the comparative results of our proposed model with several classical single-stage and two-stage networks, demonstrating the advantages of DoubleM-Net in terms of accuracy across different categories. Moreover, the detection performance varies across different categories, with bicycle, tricycle, and awning-tricycle showing the most significant improvements. Hence, our method exhibits remarkable effectiveness in detecting small targets such as bicycles and tricycles. Although the results may not be the best for other categories, they still achieve competitive performance on par with the competing models. Table 3 compares the performance of YOLOv5, YOLOv8, and DoubleM-Net on five scales: *n*, *s*, *m*, *l*, and *x*. It is evident from the table that both large and small models achieve excellent results for small targets such as bicycles, tricycles, and motorcycles. DoubleM-Net's *n*, *s*, and *m* models demonstrate respective improvements of 2.8%, 3.3%, and 2.3% in the mAP50

Fig. 11 We present the visualization results of the DroneVehicle dataset. **a–d** are the results in the RGB and Infrared mode, respectively

metric. Although the accuracy improvements for the *l* and *x* models are less significant, a slight enhancement is still observed. Overall, these findings highlight the effectiveness of DoubleM-Net in capturing small objects across different scales. In Fig. 10, we contrast the precision, recall, mAP50, and mAP50-95 of YOLOv5, YOLOv8, YOLOv8-p6, DoubleM-Net, and DoubleM-Net-p6 across five different scales. Our model demonstrates significant performance improvements at the *n*, *s*, and *m* scales, with slight improvements observed at the other two scales. Across these four metrics, our model exhibits noticeable enhancements at the *n*, *s*, and *m* scales, indicating its effectiveness in detecting small targets with higher precision and recall. While the improvements are less pronounced for the more significant *l* and *x* scales, some enhancement is still observed. Our model consistently performs well across different scales, showcasing good detection capability and performance levels.

4.4.2 Experimental results on DroneVehicle dataset

Now, we further showcase the detection performance of our model on the DroneVehicle dataset. All our models are trained from scratch without using any pre-trained weights. This ensures that our model can independently perform accurate object detection on the DroneVehicle dataset and validate its generalization capability across multiple datasets.

The detection results of DoubleM-Net on the DroneVehicle dataset are shown in Fig. 11. The figure showcases the detection visualizations of both RGB and infrared images from the dataset, represented as (a) and (c), and (b) and (d) respectively, corresponding to different modalities. It is evident from the images that DoubleM-Net can accurately identify and localize objects, regardless of whether it is day or night. Even for objects located at the images' boundaries, DoubleM-Net can recognize

Table 4 We present the effects of the *n*, *s*and *m* models of YOLOv5, YOLOv8 and DoubleM-Net in both RGB and Infrared modes on the DroneVehicle dataset, and the experimental results show that our model significantly improves the accuracy of air-to-ground images detection

Model	Modality	Car	Truck	Bus	Van	Freight Car	mAP50-95
YOLOv5(n)	RGB	62.5	39.5	64.6	32.0	27.2	45.1
YOLOv8(n)	RGB	63.3	42.1	66.8	35.3	29.3	47.4
DoubleM-Net(n)	RGB	63.8	42.0	67.0	35.3	29.6	47.6
Improvement	–	**+1.3**	−0.1	**+0.2**	=	**+2.4**	**+0.2**
YOLOv5(s)	RGB	65.2	47.6	70.8	40.4	33.8	51.6
YOLOv8(s)	RGB	65.4	49.0	70.9	41.0	35.1	52.3
DoubleM-Net(s)	RGB	65.6	50.4	70.1	41.2	37.0	52.9
Improvement	–	**+0.2**	**+1.4**	−0.8	**+0.2**	**+1.9**	**+0.6**
YOLOv5(m)	RGB	65.8	51.4	71.8	42.8	37.2	53.8
YOLOv8(m)	RGB	66.5	50.8	72.1	44.0	37.6	54.2
DoubleM-Net(m)	RGB	66.6	53.4	73.3	44.5	37.4	55.0
Improvement	–	**+0.1**	**+2.0**	**+1.2**	**+0.5**	−0.2	**+0.8**
YOLOv5(n)	Infrared	68.0	42.0	71.8	37.7	42.9	52.5
YOLOv8(n)	Infrared	68.6	44.8	73.0	39.3	46.0	54.3
DoubleM-Net(n)	Infrared	69.0	47.1	72.5	42.5	46.2	55.5
Improvement	–	**+0.4**	**+2.3**	−0.5	**+3.2**	**+0.2**	**+1.2**
YOLOv5(s)	Infrared	70.1	50.7	74.7	45.5	50.9	58.4
YOLOv8(s)	Infrared	70.2	52.7	75.0	46.6	52.6	59.4
DoubleM-Net(s)	Infrared	70.4	52.5	74.9	46.8	52.8	59.5
Improvement	–	**+0.2**	−0.2	−0.1	**+0.2**	**+0.2**	**+0.1**
YOLOv5(m)	Infrared	70.7	52.0	76.2	47.8	52.3	59.8
YOLOv8(m)	Infrared	71.1	52.7	76.1	47.4	51.8	59.8
DoubleM-Net(m)	Infrared	71.2	54.8	76.2	48.3	51.6	60.4
Improvement	–	**+0.1**	**+2.1**	=	**+0.5**	−0.7	**+0.6**

The underline highlights the performance gap between our model and the best results

them, albeit with relatively lower confidence scores. Furthermore, DoubleM-Net demonstrates impressive performance in challenging scenarios such as occluded targets and dense regions, showcasing its competitiveness. In Fig. 9b, we compare the detection performance between DoubleM-Net and YOLOv8 on the DroneVehicle dataset in RGB mode. By examining the image, it is evident that YOLOv8 exhibits false positive detections, as indicated by the orange dashed bounding box, while DoubleM-Net avoids such false positives. Additionally, we observe that YOLOv8 performs poorly when dealing with occluded targets, as demonstrated by the red dashed bounding box, whereas DoubleM-Net provides accurate detections with a confidence score of 0.25. As depicted in Fig. 9c, both models demonstrate comparable performance in Infrared mode. Table 4 presents a comparative analysis of YOLOv5, YOLOv8, and DoubleM-Net in terms of different scales, namely *n*, *s*, and m. The table demonstrates that in RGB and Infrared modes, these models achieve excellent results in detecting small objects such as cars and vans. Remarkably, the *n*, *s*, and *m* models of DoubleM-Net show a slight improvement in the mAP50-95 metric compared to the other models. Overall, these results highlight the capture capability and generalization performance of DoubleM-Net for small targets on different datasets. In the radar Fig. 12, a comparison is made between YOLOv5, YOLOv8, and DoubleM-Net for precision, recall, mAP50, mAP50-95, and F1 scores on three different scales (*n*, *s*, and *m*). The results demonstrate improvements in all five metrics for our model in the RGB mode, indicating its effectiveness in detecting air-to-ground images. In the Infrared mode, DoubleM-Net remains competitive with slight improvements in recall and precision for the *s* and *m* model sizes. Overall, our model consistently exhibits excellent detection capabilities and performance levels across different scales.

Through the discussion in the above two sections, we can state that DoubleM-Net demonstrates competitive detection performance on both datasets, thereby showcasing the superior capabilities of this model in the field of aerial image detection. DoubleM-Net proves its effectiveness in detecting small objects, adapting to scale variations, and performing well in RGB and infrared modes. These findings highlight the remarkable performance and generalization ability of DoubleM-Net in aerial image detection, providing strong support for its application in related domains.

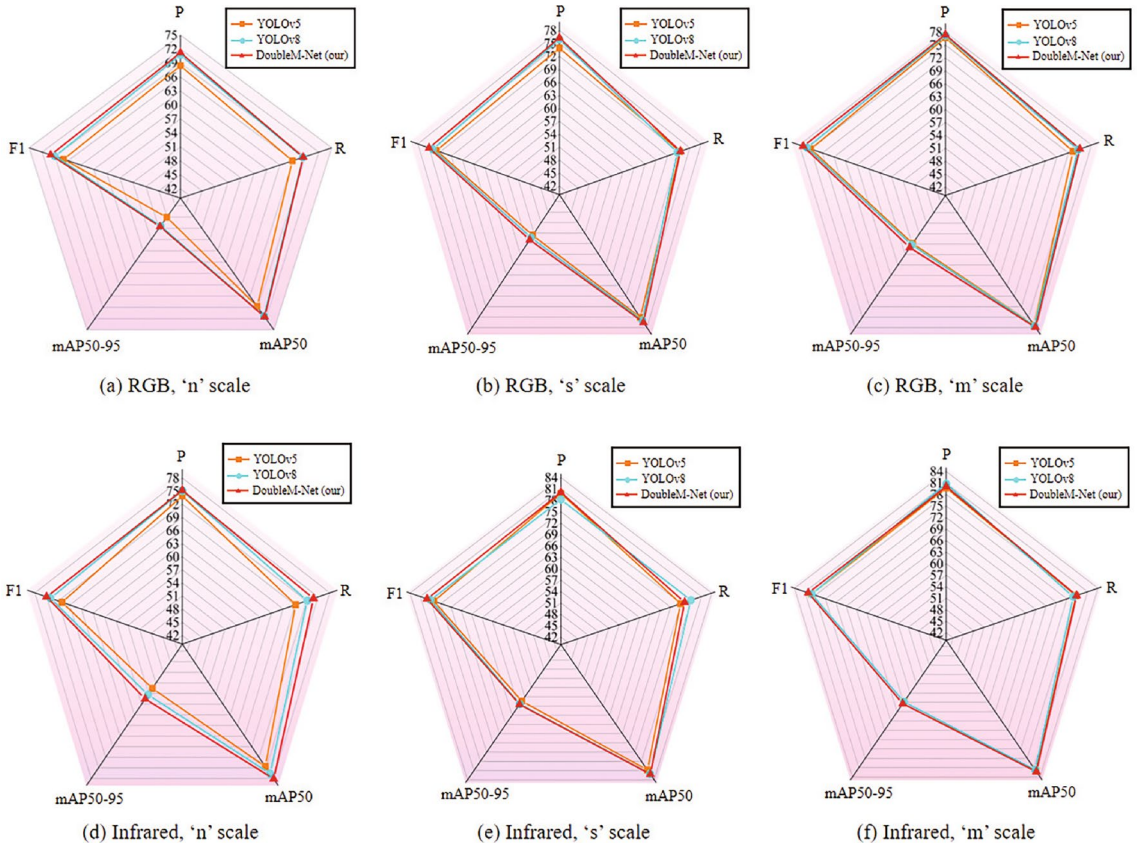

Fig. 12 Comparison of Precision, Recall, mAP 50, mAP 50-95 and F1 values for YOLOv5, YOLOv8 and DoubleM-Net, in both the RGB and Infrared modes of the DroneVehicle dataset

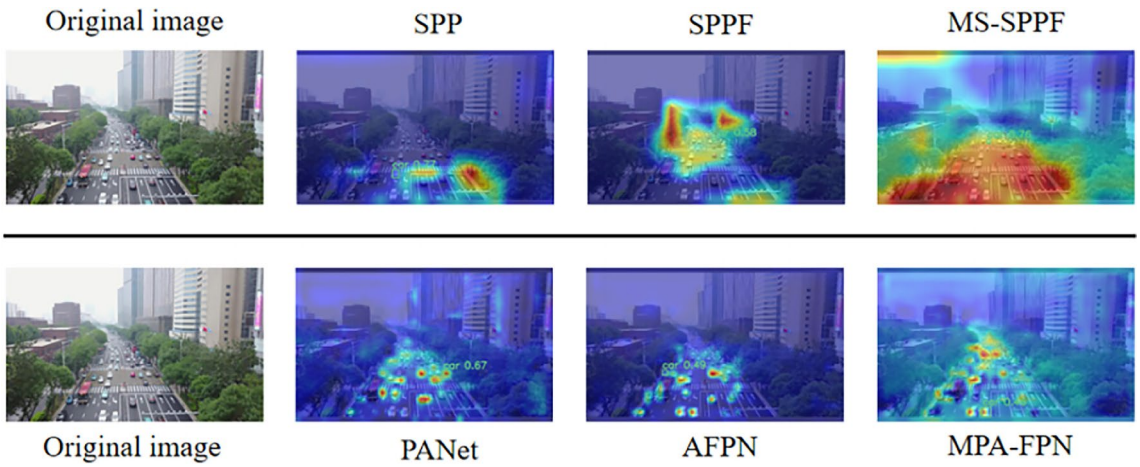

Fig. 13 Ablation visualization. The first row compares feature maps (visualization results) after SPP, SPPF, and MS-SPPF. The second row compares feature maps (visualization results) after PANet, AFPN, and MAP-FPN

4.5 Ablation studies

We systematically validated the contributions of the main modules included in DoubleM-Net to improve detection. The following experiments are conducted using the VisDrone dataset. To validate the performance of this method under different single-stage detectors, we conduct experiments based on YOLOv5 and YOLOv8.

Table 5 On the VisDrone dataset, we conducted ablation experiments on MS-SPPF module with YOLOv8 as the baseline

Model	Modules			Evaluation values										
Scales	SPP	SPPF	MS-SPPF	Ped	Peo	Bic	Car	Van	Tru	Tri	A-Tri	Bus	Mo	mAP50-95
n	✓			12.4	8.4	2.1	49.4	23.4	15.5	10.1	6.5	28.5	12.9	16.9
		✓		12.5	8.6	2.1	49.5	24.1	15.5	10.4	6.5	27.0	13.0	16.9
			✓	12.3	8.7	2.3	49.4	23.2	16.9	10.4	6.6	29.2	12.6	17.2
s	✓			17.7	11.8	3.9	54.5	29.2	22.2	13.8	9.4	37.5	17.3	21.7
		✓		17.4	11.5	4.1	53.5	28.3	21.4	13.5	8.9	37.4	17.5	21.3
			✓	17.5	11.3	4.2	53.9	29.0	22.4	14.2	9.5	38.2	17.5	21.8
m	✓			20.4	13.8	6.4	56.8	33.2	25.8	17.5	11.1	43.1	21.3	24.9
		✓		20.5	13.7	6.0	56.5	32.5	25.1	17.4	11.8	42.7	21.2	24.7
			✓	21.0	14.0	6.4	57.6	33.7	26.1	17.9	11.5	41.8	21.5	25.1
l	✓			22.2	15.0	7.5	58.3	34.8	30.3	19.7	11.9	43.4	22.8	26.6
		✓		21.9	14.9	6.8	57.9	34.3	29.0	19.8	12.1	44.8	23.0	26.5
			✓	22.4	15.2	7.0	58.2	34.5	29.8	19.0	12.0	45.3	23.5	26.7
x	✓			23.8	15.8	8.3	59.2	34.2	30.7	19.8	12.4	46.7	23.9	27.5
		✓		23.1	15.5	7.8	58.9	35.4	30.4	19.8	12.8	44.9	23.7	27.2
			✓	23.5	15.6	8.0	59.0	35.8	31.0	20.5	12.6	45.4	23.8	27.5

Table 6 On the VisDrone dataset, we conducted ablation experiments on MS-SPPF module with YOLOv5 as the baseline

Model	Modules			Evaluation values										
Scales	SPP	SPPF	MS-SPPF	Ped	Peo	Bic	Car	Van	Tru	Tri	A-Tri	Bus	Mo	mAP50-95
n	✓			11.5	8.0	1.8	48.3	22.9	14.6	9.5	6.3	26.4	11.8	16.1
		✓		11.6	7.9	2.2	48.2	22.7	14.5	9.2	5.5	25.0	12.0	15.9
			✓	11.6	7.8	2.0	48.6	23.0	13.8	10.0	6.7	26.4	12.0	16.2
s	✓			16.7	10.9	3.8	54.5	29.5	21.6	13.9	8.6	38.8	16.5	21.5
		✓		16.8	10.9	3.6	54.5	29.8	20.8	12.8	9.0	34.6	16.7	21.0
			✓	17.0	11.4	3.8	54.4	29.6	21.4	14.2	8.2	37.7	17.1	21.5
m	✓			20.1	13.3	5.5	57.6	32.6	26.0	16.6	10.5	41.6	20.0	24.4
		✓		20.1	13.0	5.4	57.2	32.3	25.0	15.6	10.4	41.3	20.1	24.0
			✓	20.2	13.4	5.7	57.6	33.4	25.4	16.3	11.4	41.5	20.6	24.6
l	✓			22.0	14.6	6.7	58.2	34.3	28.9	18.9	11.7	44.1	22.4	26.2
		✓		21.8	14.2	6.4	58.1	33.7	29.3	18.8	11.3	46.4	22.1	26.2
			✓	22.0	14.8	6.6	59.6	35.2	28.8	18.0	11.1	45.7	22.5	26.4
x	✓			23.4	15.1	7.5	59.3	35.5	30.7	19.7	12.2	46.8	24.1	27.4
		✓		23.4	15.4	7.8	58.9	34.6	30.9	19.7	12.4	46.5	23.7	27.3
			✓	23.4	15.1	8.4	59.2	35.5	29.9	20.4	12.5	46.5	23.9	27.5

4.5.1 Effect of MS-SPPF

The proposed MS-SPPF module in this paper leverages multi-scale pooling operations with repeated application of different-sized pooling kernels to extract critical information and multi-scale features of small targets. Ablation studies are conducted on the VisDrone dataset to evaluate the effectiveness of this module. Tables 5 and 6 present the advantages of MS-SPPF on baseline YOLOv8 and YOLOv5, respectively. According to the experimental results, both YOLOv8 and YOLOv5 showed a certain degree of improvement in mAP50-95 on different model sizes *n*, *s*, *m*, *l*, and *x*. These tables show that MS-SPPF outperforms SPP and SPPF in detecting small objects such as tricycles and awning-tricycles. Moreover, our model demonstrates competitive performance in other categories as well. These findings highlight the significant advantage of the MS-SPPF module in enhancing the detection of small objects and underscore the competitiveness of our model across multiple target categories. By comparing Tables 7 and 8, the influence of different pooling kernels (5, 5, 5), (9, 9, 9), (13, 13, 13), and (5, 9, 13) on precision, recall, mAP50, and mAP50-95 metrics in the MS-SPPF module can be observed. Tables 7 and 8 are based on YOLOv8 and YOLOv5, respectively. From the data in these tables, it can be concluded that, for different

Table 7 On the VisDrone dataset, we conduct ablation experiments on the MS-SPPF module with different pooling kernels using YOLOv8 as the baseline

Model	MS-SPPF				Evaluation values						
Scales	(5,5,5)	(9,9,9)	(13,13,13)	(5,9,13)	Param	GFLOPs	P	R	mAP50	mAP50-95	Time
n	✓				3.2	8.4	39.9	29.9	29.2	16.5	3.1
		✓			3.2	8.4	40.3	30.0	29.5	16.7	2.1
			✓		3.2	8.4	41.4	30.5	29.8	16.9	2.0
				✓	3.2	8.4	41.6	30.6	29.9	17.2	5.1
s	✓				11.9	29.3	48.0	35.8	37.1	21.8	9.0
		✓			11.9	29.3	48.2	36.3	37.0	21.8	8.9
			✓		11.9	29.3	46.6	36.5	36.8	21.7	8.7
				✓	11.9	29.3	47.6	36.4	37.3	22.0	8.8
m	✓				26.8	79.3	52.6	40.4	41.5	25.1	13.0
		✓			26.8	79.3	52.1	39.4	41.3	25.0	11.0
			✓		26.8	79.3	51.3	39.9	41.5	25.1	12.1
				✓	26.8	79.3	52.8	40.6	41.7	25.2	10.8
l	✓				44.4	166.1	54.7	41.9	43.6	26.7	24.6
		✓			44.4	166.1	55.1	41.3	43.5	26.7	27.6
			✓		44.4	166.1	54.6	41.8	43.4	26.4	31.5
				✓	44.4	166.1	54.7	42.3	43.7	26.9	24.4
x	✓				69.3	259.2	56.7	42.8	44.7	27.6	33.2
		✓			69.3	259.2	55.8	42.5	44.5	27.3	34.1
			✓		69.3	259.2	54.0	43.7	44.8	27.5	32.9
				✓	69.3	259.2	56.4	42.7	44.9	27.8	30.3

Param. in the table is the number of parameters (in *M*). P and R are precision and recall, respectively. Time is the inference time (in *ms*). The same is true for Tables 8, 10 and 11 below

Table 8 On the VisDrone dataset, we conducted ablation experiments on the MS-SPPF module with different pooling kernels using YOLOv5 as the baseline

Model	MS-SPPF				Evaluation values						
Scales	(5,5,5)	(9,9,9)	(13,13,13)	(5,9,13)	Param	GFLOPs	P	R	mAP50	mAP50-95	Time
n	✓				2.7	7.3	38.5	28.9	28.2	16.0	3.5
		✓			2.7	7.3	39.2	29.1	28.4	16.1	3.1
			✓		2.7	7.3	39.4	29.0	28.1	16.0	3.5
				✓	2.7	7.3	38.8	29.5	28.4	16.2	3.4
s	✓				9.9	24.7	46.7	36.0	36.3	21.3	4.1
		✓			9.9	24.7	46.5	36.1	36.4	21.2	4.0
			✓		9.9	24.7	46.0	35.6	35.8	21.1	4.3
				✓	9.9	24.7	46.7	36.1	36.5	21.5	4.6
m	✓				26.8	65.8	50.3	40.1	40.7	24.5	7.0
		✓			26.8	65.8	50.8	39.0	40.2	24.2	7.2
			✓		26.8	65.8	51.7	39.6	40.7	24.4	7.1
				✓	26.8	65.8	51.7	39.7	40.9	24.6	7.2
l	✓				56.3	137.8	53.8	41.1	42.9	26.1	15.8
		✓			56.3	137.8	54.8	41.9	43.8	26.3	14.4
			✓		56.3	137.8	53.5	41.7	43.2	26.1	15.9
				✓	56.3	137.8	54.6	41.8	43.4	26.6	13.5
x	✓				102.1	250.9	55.4	42.9	44.5	27.3	17.9
		✓			102.1	250.9	55.8	42.5	44.7	27.5	16.2
			✓		102.1	250.9	55.6	42.6	44.3	27.2	17.1
				✓	102.1	250.9	55.7	43.1	44.9	27.5	16.3

Table 9 Comparing the precision results of MS-SPPF and MPA-FPN ablation experiments on the VisDrone dataset for ten categories, with YOLOv8 as the baseline

YOLOv8	Modules	Ped	Peo	Bic	Car	Van	Truck	Tri	A-tri	Bus	Mo	Precision
n	Baseline	40.8	47.0	22.3	61.6	42.6	38.6	31.4	20.7	52.0	44.1	40.1
	+MS-SPPF	39.8	48.5	22.9	61.9	42.5	40.2	39.7	26.4	48.6	45.1	41.6
	+MPA-FPN	43.7	51.5	26.4	65.1	45.4	44.4	38.4	28.6	53.3	50.4	44.7
s	Baseline	48.1	53.6	27.9	70.6	46.6	46.5	39.5	27.8	60.4	52.6	47.4
	+MS-SPPF	47.2	52.7	23.9	69.9	48.1	48.4	39.5	26.4	57.4	52.3	46.6
	+MPA-FPN	53.2	58.4	29.8	72.6	48.5	49.6	46.0	35.2	66.6	58.1	51.8
m	Baseline	55.6	59.2	30.4	75.9	50.4	52.9	45.4	31.4	64.7	56.9	52.3
	+MS-SPPF	54.4	58.2	28.9	74.6	51.1	52.8	46.4	33.2	63.3	57.5	52.0
	+MPA-FPN	55.8	58.7	31.1	76.3	51.0	53.2	49.6	34.1	69.5	61.5	54.1
l	Baseline	57.1	59.3	31.1	77.8	52.5	50.5	47.5	33.7	71.7	58.7	54.0
	+MS-SPPF	56.7	60.3	32.8	76.1	53.8	52.5	47.0	34.0	70.1	57.7	54.1
	+MPA-FPN	58.9	60.3	38.1	78.6	52.3	56.4	49.8	34.4	68.3	62.8	56.0
x	Baseline	60.7	61.7	33.1	78.9	51.2	55.6	49.6	33.7	72.4	60.2	55.7
	+MS-SPPF	59.8	61.4	33.2	79.5	53.5	57.5	52.9	33.7	71.3	61.3	56.4
	+MPA-FPN	61.2	62.2	37.4	79.4	53.4	58.2	49.9	36.8	71.4	64.1	57.4

Table 10 Considering YOLOv5 as the baseline, the precision, recall, mAP50, and mAP50-95 metrics of MS-SPPF and MPA-FPN ablation experiments are compared on different model sizes (*n*, *s*, *m*, *l*, and *x*) on the VisDrone dataset

YOLOv5	Modules	Param	GFLOPs	P	R	mAP50	mAP50-95	Time
n	Baseline	2.5	7.2	39.8	28.7	28.3	15.9	3.6
	+MS-SPPF	2.7	7.3	38.8	29.5	28.4	16.2	3.4
	+MPA-FPN	4.9	21.7	43.4	30.8	31.5	18.2	5.3
s	Baseline	9.1	24.1	46.4	35.5	35.9	21.0	4.2
	+MS-SPPF	9.9	24.7	46.7	36.1	36.5	21.5	4.6
	+MPA-FPN	18.7	79.9	50.0	37.5	38.6	23.0	15.5
m	Baseline	25.1	64.4	51.2	39.2	40.3	24.0	7.6
	+MS-SPPF	26.8	65.8	51.7	39.7	40.9	24.6	7.2
	+MPA-FPN	47.8	201.7	54.0	40.5	41.9	25.6	17.9
l	Baseline	53.2	135.3	54.3	41.7	43.1	26.2	12.9
	+MS-SPPF	56.3	137.8	54.6	41.8	43.4	26.6	13.5
	+MPA-FPN	95.2	398.6	54.7	41.9	43.5	26.8	43.9
x	Baseline	97.2	246.9	55.8	42.4	44.4	27.3	18.7
	+MS-SPPF	102.1	250.9	55.7	43.1	44.9	27.5	16.3
	+MPA-FPN	164.7	685.9	55.8	43.3	44.7	27.6	58.1

model sizes, the pooling kernel (5, 9, 13) exhibits a significant advantage. All four metrics improve using the (5, 9, 13) kernel. This highlights the crucial role of selecting an appropriate pooling kernel size in enhancing detection performance. The first row of Fig. 13 shows that the feature maps processed by MS-SPPF have a more significant effect in presenting the contours and shapes of the objects compared to SPP and SPPF. Even when facing objects of irregular size and shape, MS-SPPF can still effectively capture and emphasize their key features.

4.5.2 Effect of MPA-FPN

In this section, we similarly explore the effectiveness of the MPA-FPN module on the VisDrone dataset. Taking YOLOv8 as the baseline, we gradually introduce the MS-SPPF and MPA-FPN module, comparing their precision on the ten categories, as shown in Table 9. Encouragingly, we observe a significant improvement in precision after incorporating the MPA-FPN module. Whether it is for small objects such as tricycles or easily confusable objects like cars, vans, and buses, impressive detection results are achieved. Continuing with the YOLOv5 baseline, we repeat the process by introducing the MS-SPPF module first and then incorporating the MPA-FPN module. We analyze the precision, recall, mAP50, and mAP50-95 metrics at each step, as presented in Table 10. It is evident from the results that the integration of the MPA-FPN module led to substantial improvements in precision, recall, and mAP values. This underscores the significance of the MPA-FPN module in enhancing object

Table 11 This table shows the object detection results of the other target detectors and the DoubleM-Net model on the VisDrone validation dataset

Model	Param	GFLOPs	P	R	mAP50	mAP50-95	Time
YOLOv5(n)	2.5	7.2	39.8	28.7	28.3	15.9	3.6
YOLOv5(s)	9.1	24.1	46.4	35.5	35.9	21.0	4.2
YOLOv5(m)	25.1	64.4	51.2	39.2	40.3	24.0	7.6
YOLOv5(l)	53.2	135.3	53.9	41.7	43.1	26.2	12.9
YOLOv8(n)	3.1	8.9	40.1	30.6	29.9	16.9	3.1
YOLOv8(s)	11.2	28.8	47.4	35.9	36.2	21.3	8.9
YOLOv8(m)	25.9	79.3	52.3	39.6	41.2	24.7	11.2
YOLOv8(l)	43.7	165.7	54.0	42.0	43.2	26.5	27.7
YOLOv9	60.5	264.0	54.5	41.7	43.8	26.7	31.6
YOLOv9-c	50.7	236.7	55.4	42.3	44.5	27.2	33.3
YOLOv9-e	68.8	240.8	54.5	42.2	43.9	26.9	41.5
DoubleM-Net(n)	5.5	23.7	44.7	32.0	32.7	19.1	5.2
DoubleM-Net(s)	20.9	88.5	51.8	38.0	39.5	23.6	15.2
DoubleM-Net(m)	53.5	238.8	54.1	41.3	43.5	26.5	25.7
DoubleM-Net(l)	104.0	493.7	56.0	44.6	44.7	27.5	49.7

All models in the table are trained from scratch using the original images.YOLOv5 is trained in the framework of YOLOv8

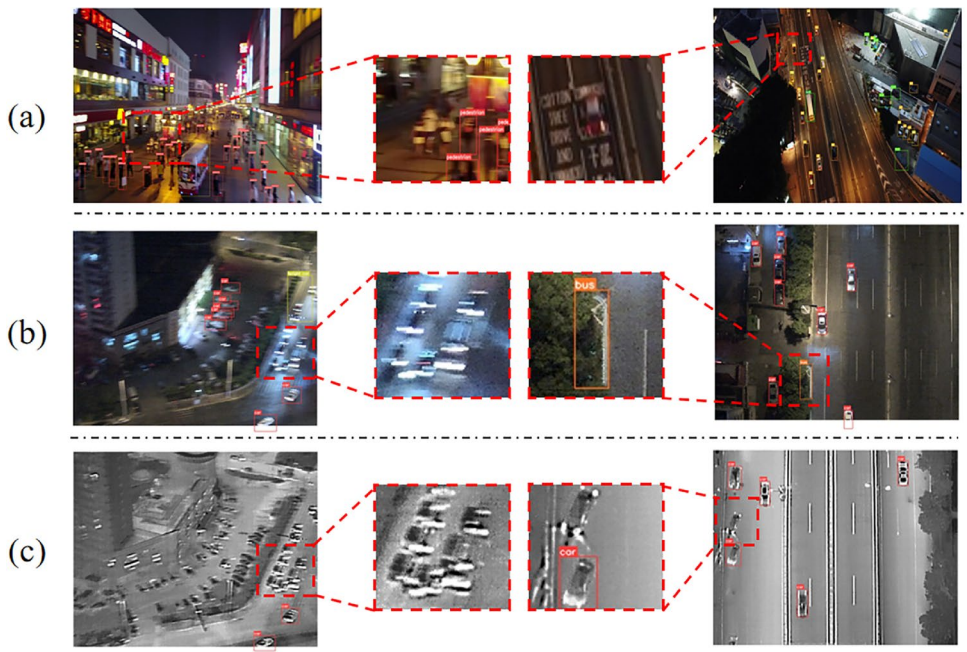

Fig. 14 DoubleM-Net has some limitations in its processing effectiveness in different datasets and modes. Specifically, **a** in the VisDrone dataset, DoubleM-Net is ineffective in processing images in blurred and nighttime environments; **b** in the RGB mode of the DroneVehicle dataset, DoubleM-Net also faces the problem of ineffective processing in blurred and nighttime environments; **c** for the infrared mode of the DroneVehicle dataset, DoubleM-Net also exhibits a lack of processing power in blurred and nighttime environments. These limitations are identified in the red dashed box in Fig.

detection performance. Based on the findings from these ablation experiments, it is evident that incorporating the MPA-FPN module, whether in YOLOv8 or YOLOv5, significantly improves the accuracy and effectiveness of object detection. This further substantiates the efficacy and competitiveness of the MPA-FPN module in multi-scale object detection. The second row of Fig. 13 shows that the feature maps processed by MPA-FPN show a more outstanding ability to highlight the contours and shapes of the objects compared with PANet and AFPN. MPA-FPN cannot only accurately locate the object's exact position in the image but also effectively identify the size and shape features of the object. In addition, for small targets, the effect of MPA-FPN is superior and can capture and present their detailed features more accurately.

4.6 Limitation analysis

Although the DoubleM-Net model exhibits good detection performance on the VisDrone and DroneVehicle datasets, it is still insufficient in complex scenarios such as blur and nighttime, which is visualized in Fig. 14. Specifically, as

shown in Fig. 14a, in the VisDrone dataset, DoubleM-Net's recognition ability is significantly affected when dealing with images in blurred and nighttime environments, resulting in poor detection results; in Fig. 14b, when facing RGB-mode images in the DroneVehicle dataset, DoubleM-Net similarly faces the problem of poor processing in blur and nighttime environments. Figure 14c further shows that DoubleM-Net's processing ability in blur and nighttime environments also appears to be stretched when confronted with the infrared mode of the DroneVehicle dataset. These limitations are clearly labeled in the red dashed box in Fig. 14.

The lack of information and unclear details in blurred images significantly challenge the model's recognition. Due to the blurred images, the model is limited in extracting features and performing recognition, making it challenging to accurately capture essential information in the images. Meanwhile, images in nighttime environments often suffer from insufficient lighting, which leads to reduced image contrast and detailed information becoming difficult to distinguish, thus further increasing the difficulty of model processing. In addition, the noise and interference factors that may exist in nighttime environments can also adversely affect the model's performance. Therefore, in-depth study and optimization of these limitations are needed in the following research work to improve further the DoubleM-Net model's detection effect in complex environments such as blurred and nighttime environments.

Table 11 compares the DoubleM-Net model with other target detectors on key performance metrics, such as the number of parameters, GFLOPs, precision, recall, mAP50, mAP50-95, and inference time. By analyzing these data in depth, we can find that although the DoubleM-Net model exhibits notable detection performance, it is also accompanied by some significant limitations. First, the number of parameters of the DoubleM-Net model is relatively large, which means that the model requires more computational resources and storage space during training and deployment. Second, the increase in GFLOPs also indicates that the model requires higher computational effort in performing forward propagation, which may lead to slower inference in practical applications, especially in scenarios with high real-time requirements. In addition, the extended inference time further limits the application of the DoubleM-Net model in real-time scenarios.

Despite these limitations, the DoubleM-Net(m) model is still comparable to YOLOv8(l), YOLOv9, YOLOv9-c, and YOLOv9-e in terms of detection effectiveness, which to some extent proves the superiority of its detection performance. Nevertheless, while pursuing high detection accuracy, there is also a need to weigh the number of parameters and computational complexity. Reducing the number of parameters and computational complexity of the model under the premise of guaranteeing the detection accuracy is the current direction of further optimization and improvement of the DoubleM-Net model.

5 Conclusion

Dynamic environments and numerous small targets often lead to low object detection accuracy in aerial scenes. In this paper, we propose an innovative approach called DoubleM-Net to optimize the detection performance in UAV scenarios. The method consists of two key modules we designed, MS-SPPF and MPA-FPN. Among them, MS-SPPF performs multiple pooling operations using pooling kernels of different sizes (k = 5, 9, 13), effectively capturing spatial features at different scales. Second, to overcome the limitations of feature pyramid networks in solving scale-varying problems, we construct an original MPA-FPN structure. By optimizing the feature fusion method, MPA-FPN effectively reduces the information contradiction between non-adjacent features and enhances the interaction between low-level and high-level semantic information. A new approach is provided to solve the scale change problem in object detection. The experimental results show that the mAP50-95 of DoubleM-Net is 27.5% on the VisDrone dataset. In contrast, on the DroneVehicle dataset, the mAP50-95 of DoubleM-Net is 55.0% and 60.4% in RGB and IR modes, respectively. In addition, our model performs well in the air-to-ground image detection task and excels in detecting small objects.

Improving the detection accuracy comes at the cost of significantly increasing the computational requirements. doubleM-Net puts pressure on computational resources, which will be a significant challenge for future work. Therefore, we will carry out the following work in the future:

1. To maintain high accuracy, mitigate the number of detector parameters and computations.
2. Explore distillation and pruning techniques to optimize the lightweight detector design.
3. Explore the detection effect in complex situations such as blurring and nighttime.

Acknowledgements The research is supported by the National Natural Science Foundation of China under Grant No. 62376114, the National Natural Science Foundation of China under Grant No.12101289, the Natural Science Foundation of Fujian Province under Grant Nos.2020J01821 and 2022J01891. And it is supported by the Institute of Meteorological Big Data-Digital Fujian, and Fujian Key Laboratory of Data Science and Statistics (Minnan Normal University), China.

Author contributions Zhongxu Li, Qihan He, Hong Zhao and Wenyuan Yang contribute equally to this work.

Data availability and access All data, models, and code generated and utilized in this study are available upon reasonable request from the corresponding author. The codes will upload on https://github.com/yangwygithub/PaperCode.git, Branch: DoubleM-Net_Zhongxu-Li2023.

Declarations

Conflict of 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.

Ethical and informed consent for data used This article does not contain any research conducted by any author on human participants or animals and informed consent is obtained from all individual participants included in the study.

References

1. Zou Z, Chen K, Shi Z, Guo Y, Ye J (2023) Object detection in 20 years: a survey. Proc IEEE 111(3):257–276
2. Wang X, Zhao Y, Pourpanah F (2020) Recent advances in deep learning. Int J Mach Learn Cybern 11:747–750
3. Cui J, Qin Y, Wu Y, Shao C, Yang H (2023) Skip connection yolo architecture for noise barrier defect detection using uav-based images in high-speed railway. IEEE Trans Intell Transp Syst 24(11):12180–12195
4. Li X, Wu J (2023) Developing a more reliable framework for extracting traffic data from a uav video. IEEE Trans Intell Transp Syst 24(11):12272–12283
5. Huang J, Jiang X, Jin G (2022) Detection of river floating debris in uav images based on improved yolov5. In: 2022 International Joint Conference on Neural Networks, pp 1–8
6. Sun L, Zhang Y, Ouyang C, Yin S, Ren X, Fu S (2023) A portable uav-based laser-induced fluorescence lidar system for oil pollution and aquatic environment monitoring. Opt Commun 527:128914–128928
7. Furusawa T, Premachandra C (2023) Innovative colormap for emphatic imaging of human voice for uav-based disaster victim search. In: 2023 IEEE Region 10 Symposium, pp. 1–5
8. Dorn C, Depold A, Lurz F, Erhardt S, Hagelauer A (2022) Uav-based localization of mobile phones for search and rescue applications. In: 2022 IEEE 22nd Annual Wireless and Microwave Technology Conference, pp. 1–4
9. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587
10. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
11. Ren S, He K, Girshick R, Sun J (2017) Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39:1–14
12. Lin TY, Dollar P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125
13. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788
14. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271
15. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement arXiv:1804.02767
16. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection arXiv:2004.10934
17. Jocher G (2020) YOLOv5 by Ultralytics
18. Li C, Li L, Geng Y, Jiang H, Cheng M, Zhang B, Ke Z, Xu X, Chu X (2023) Yolov6 v3.0: a full-scale reloading arXiv:2301.05586
19. Wang CY, Bochkovskiy A, Liao HYM (2022) Yolov7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors arXiv:2207.02696
20. Jocher G, Chaurasia A, Qiu J (2023) YOLO by Ultralytics
21. Wang CY, Yeh IH, Liao HYM (2024) Yolov9: learning what you want to learn using programmable gradient information arXiv:2402.13616
22. Xu X, Zhang X, Zhang T (2022) Lite-yolov5: a lightweight deep learning detector for on-board ship detection in large-scene sentinel-1 sar images. Remote Sens 14:1018–1030
23. Xu X, Jiang Y, Chen W, Huang Y, Zhang Y, Sun X (2023) Damo-yolo: a report on real-time object detection design arXiv:2211.15444
24. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
25. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
26. Liu S, Huang D, Wang a (2018) Receptive field block net for accurate and fast object detection. In: Proceedings of the European Conference on Computer Vision, pp. 385–400
27. Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2016) Inception-v4, inception-resnet and the impact of residual connections on learning. Proc AAAI Conf Artif Intell 31:11231–11245
28. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: Computer Vision—ECCV 2016: 14th European Conference, pp. 21–37
29. 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
30. Tan M, Pang R, Le QV (2020) Efficientdet: scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781–10790
31. Zhang T, Zhang X, Ke X (2021) Quad-fpn: a novel quad feature pyramid network for sar ship detection. Remote Sens 13:2771–2785
32. Jiang Y, Tan Z, Wang J, Sun X, Lin M, Li H (2022) Giraffedet: a heavy-neck paradigm for object detection arXiv:2202.04256
33. Xu X, Zhang X, Shao Z, Shi J, Wei S, Zhang T, Zeng T (2022) A group-wise feature enhancement-and-fusion network with dual-polarization feature enrichment for sar ship detection. Remote Sens 14:5276–5291
34. Yang G, Lei J, Zhu Z, Cheng S, Feng Z, Liang R (2023) Afpn: asymptotic feature pyramid network for object detection arXiv:2306.15988
35. Saqib M, Khan SD, Sharma N, Blumenstein M (2017) A study on detecting drones using deep convolutional neural networks. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 1–5
36. Chen C, Zhang Y, Lv Q, Wei S, Wang X, Sun X, Dong J (2019) Rrnet: a hybrid detector for object detection in drone-captured

images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 100–108
37. Khan SD, Alarabi L, Basalamah S (2022) A unified deep learning framework of multi-scale detectors for geo-spatial object detection in high-resolution satellite images. Arab J Sci Eng 47(8):9489–9504
38. Zhang R, Shao Z, Huang X, Wang J, Li D (2020) Object detection in uav images via global density fused convolutional network. Remote Sens 12(19):3140–3143
39. Tian G, Liu J, Yang W (2021) A dual neural network for object detection in uav images. Neurocomputing 443:292–301
40. Chen J, Wang Q, Peng W, Xu H, Li X, Xu W (2022) Disparity-based multiscale fusion network for transportation detection. IEEE Trans Intell Transp Syst 23(10):18855–18863
41. Li S, Chen J, Peng W, Shi X, Bu W (2023) A vehicle detection method based on disparity segmentation. Multimed Tools Appl 82(13):19643–19655
42. Ma B, Liu Z, Dang Q, Zhao W, Wang J, Cheng Y, Yuan Z (2023) Deep reinforcement learning of uav tracking control under wind disturbances environments. IEEE Trans Instrum Meas 72(5):1–13
43. Zhang R, Shao Z, Huang X, Wang J, Wang Y, Li D (2022) Adaptive dense pyramid network for object detection in uav imagery. Neurocomputing 489:377–389
44. Wang T, Ma Z, Yang T, Zou S (2023) Petnet: a yolo-based prior enhanced transformer network for aerial image detection. Neurocomputing 547:126384–126399
45. Liu S, Huang D, Wang Y (2019) Learning spatial fusion for single-shot object detection arXiv:1911.09516
46. Zhu P, Wen L, Du D, Bian X, Fan H, Hu Q, Ling H (2022) Detection and tracking meet drones challenge. IEEE Trans Pattern Anal Mach Intell 44(11):7380–7399
47. Sun Y, Cao B, Zhu P, Hu Q (2022) Drone-based rgb-infrared cross-modality vehicle detection via uncertainty-aware learning. IEEE Trans Circuits Syst Video Technol 32(10):6700–6713
48. Zhu C, He Y, Savvides M (2019) Feature selective anchor-free module for single-shot object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 840–849
49. Zhang S, Chi C, Yao Y, Lei Z, Li SZ (2019) Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection arXiv:1912.02424
50. Li Y, Chen Y, Wang N, Zhang Z (2019) Scale-aware trident networks for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6054–6063
51. Zhou X, Wang D, Krähenbühl P (2019) Objects as points
52. Tian Z, Shen C, Chen H, He T (2019) Fcos: fully convolutional one-stage object detection arXiv:1904.01355
53. Chen Z, Yang C, Li Q, Zhao F, Zha ZJ, Wu F (2021) Disentangle your dense object detector. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 4939–4948
54. Feng C, Zhong Y, Gao Y, Scott MR, Huang W (2021) Tood: task-aligned one-stage object detection. In: 2021 IEEE/CVF International Conference on Computer Vision, pp. 3490–3499
55. Zhang H, Wang Y, Dayoub F, Sünderhauf N (2020) Varifocalnet: an iou-aware dense object detector arXiv:1200.81336
56. Cai Z, Vasconcelos N (2019) Cascade r-cnn: high quality object detection and instance segmentation. IEEE Trans Pattern Anal Mach Intell 43:1–15
57. Ge Z, Liu S, Wang F, Li Z, Sun J (2021) Yolox: exceeding yolo series in 2021 arXiv:2107.08430