The Journal of Supercomputing (2025) 81:105
https://doi.org/10.1007/s11227-024-06611-x

# LightUAV-YOLO: a lightweight object detection model for unmanned aerial vehicle image

Yifan Lyu[1] · Tianze Zhang[2] · Xin Li[1] · Aixun Liu[1] · Gang Shi[1]

Accepted: 10 October 2024

## Abstract

Object detection in unmanned aerial vehicle (UAV) images presents challenges such as high altitudes, small object sizes, and complex backgrounds. Additionally, many deep learning object detection algorithms require substantial computational resources, making them difficult to deploy on embedded devices with limited memory and processing power, which affects the effectiveness of drones in task execution. To tackle these issues, we propose the LightUAV-YOLO algorithm which is a lightweight object detection algorithm for UAVs based on YOLOv8n. We modified the neck structure of YOLOv8, enhancing the network's capability to detect small objects. To further optimize features fusion at different scales, we designed the orthogonal feature enhancement module (OFEM) which replaces simple concatenation for better feature representation. We also designed the local attention module (LAM) to effectively filter out irrelevant interference. The module enables our model better focus on important areas and further enhancing the model's robustness. Results demonstrate that our proposed LightUAV-YOLO algorithm achieves a 6.4 and 3.9% improvement in mAP50 and mAP50:95, respectively, on the VisDrone test dataset compared to the YOLOv8-nano. Meanwhile, the model maintains a low parameter count and computational complexity. Furthermore, we conducted extensive experiments on the UAVDT dataset, and our method consistently exhibited favorable results. This model not only meets accuracy requirements but also considers the lightweight requirements.

**Keywords** UAV remote sensing images · Feature fusion · YOLOv8 · Attention mechanism · Object detection

Yifan Lyu and Tianze Zhang have contributed equally to this work.

Extended author information available on the last page of the article

Published online: 30 October 2024

## 1 Introduction

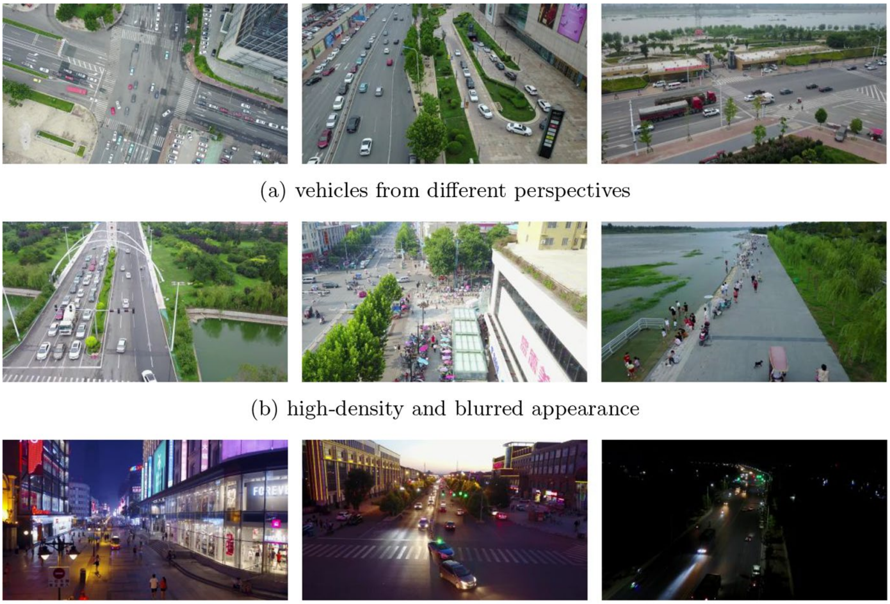
Object detection has become a fundamental task in computer vision, with wide-ranging applications [1–4] including autonomous driving, surveillance systems, and UAV aerial analysis, playing a crucial role in real-world scenarios. In recent years, the convenience of unmanned aerial vehicles (UAVs) has led to a significant increase in their usage across various fields. UAVs have unique advantages in capturing large-scale high-resolution images and video data, making them highly applicable in the field of computer vision.

Deep learning, especially the advent of convolutional neural networks (CNNs), has revolutionized the field of computer vision. Deep learning technologies have brought significant breakthroughs in the domain of object detection. By automatically learning multilevel feature representations from raw data, deep learning models can autonomously extract complex visual features from images without relying on manually designed feature extraction algorithms. There are two main types of object detection networks. Two-stage object detection networks, represented by R-CNN [5–7], first generate candidate regions and then classify and regress these regions. In contrast, single-stage networks, such as YOLO [8–12] and SSD [13], perform end-to-end classification and regression directly on the objects, achieving faster detection speeds. Recent methods typically build on these foundational detection frameworks to improve accuracy and performance. Generally, two-stage networks offer higher detection accuracy, while single-stage networks provide faster detection speeds. Therefore, it is essential to balance accuracy and real-time performance and choose an appropriate method to meet the needs of different application scenarios.

Different from conventional images, UAV-captured images present several challenges: (1) small object detection: large-scale objects, due to their size and rich feature representation, are typically easier to detect, which has led to more significant progress in large object detection. However, small objects, due to their limited area and vulnerability to noise, pose a major challenge for detection [14, 15]. (2) Dense object clusters: in many scenarios, UAV images contain densely arranged objects, where numerous similar objects are grouped together, often resulting in significant occlusion. (3) Diverse object characteristic: due to the varied shooting angles of drones, objects exhibit a diversity of characteristics. (4) Illumination and environmental issues: the broad scene coverage of UAV images can cause various illumination issues, such as insufficient lighting, blurriness, and objects confused with the background, further exacerbating the challenge of UAV object detection (Fig. 1).

These challenges underscore the research value of developing and optimizing deep learning-based detection algorithms to enhance the capability of UAVs in detecting targets in complex environments. This has become a prominent and highly challenging research topic in the academic field.

Feature pyramid network [18] (FPN) has been a focal module in object detection, attracting significant attention from numerous scholars who have iteratively improved upon it [19–22], producing promising outcomes. FPN offers



(a) vehicles from different perspectives

(b) high-density and blurred appearance

(c) various lighting conditions

**Fig. 1** Sample images taken from UAVDT [16] and Visdrone2021 [17]. (**a**), (**b**), and (**c**) describes the main problems of object detection in UAV images

two key benefits [23]: (1) divide and conquer: it detects objects at different levels based on their scales. (2) Feature fusion: FPN integrates shallow-level fine-grained details with deep-level high-level semantic information, yielding more discriminative features and significantly enhancing feature expressiveness. In deep convolutional networks, shallow-level features contain richer fine-grained information but have a poorer grasp of global relationships, which makes them more suitable for detecting densely packed small objects. Conversely, deep-level features encompass holistic semantic information but exhibit a weaker perception of details, making them more suitable for detecting large-sized objects. Small objects often suffer from weak features due to their small size, even to the extent of being lost in the background after multiple down-sampling stages. However, the P2 feature map, which contains the richest texture information for small objects, does not participate in the information transmission or detection within the FPN. To enhance the detection performance of small objects, a common approach is to incorporate the P2 layer into the FPN for feature fusion [24–27]. While this method effectively improves the detection performance of small objects, it also poses certain limitations. Firstly, introducing high-resolution feature maps into FPN increases model parameters and computational complexity significantly. Secondly, as features undergo continuous dimension expansion during the down-sampling process of the backbone network, deep-level features transmitted to FPN may exhibit redundancy in the process of small object detection.

In traditional FPN, each layer not only needs to focus on its corresponding scale objects but also has to incorporate information from other layers. This leads to receiving assistance from other layers while being interfered with by others [28]. Furthermore, the layers are concatenated together, which treats all feature layers equally. Each layer should prioritize its own information while using the information from other layers as supplementary. To optimize feature fusion across different scales and enhance feature robustness, various methods have been proposed. Wang et al. [29] introduced the spatial-aware module (SAM), which first uses deformable convolutions to adaptively learn the optimal convolution kernel structure and sampling points to individually optimize features at different scales before concatenation. Li et al. [30] proposed the bidirectional concatenation (BiC) module to integrate feature maps from three adjacent layers. Subsequently, Zhang et al. [25] improved the BiC module with depthwise (DW) convolutions to propose the sandwich module, both methods aiming to retain more accurate localization signals by fusing different layer features. The essence of the above methods is simple concatenation, treating all feature layers at different scales equally and not considering the relationship between different scales. Gong et al. [28] proposed a method based on fusion factors to control the information transfer from deep to shallow layers in the FPN, mitigating the negative impact of top-down connections. However, these algorithms still face challenges in accurately and efficiently detecting aerial targets in remote sensing images. Therefore, it is necessary to improve detection accuracy and enhance classification capabilities among similar categories. To address these issues, this study proposes the LightUAV-YOLO algorithm, with its contributions as follows:

1. Based on the analysis of the YOLOv8 model and UAV detection tasks, we propose a lightweight multi-scale information fusion network, LightUAV-YOLO, for aerial object detection using drones. Two newly proposed modules are embedded into the single-stage object detection algorithm YOLOv8.
2. The orthogonal feature enhancement module is designed to integrate into the network to further improve the fusion of features at different scales and enhance the model's detection performance.
3. To further augment the network's ability to focus on and recognize small target areas, we analyzed the advantages and disadvantages of commonly used attention mechanisms based on scene analysis and designed a local attention module accordingly. Comparative experiments demonstrate that our module achieves the best results.
4. A series of experiments were conducted on the UAV aerial datasets Visdrone2021 and UAVDT, and the experimental results were analyzed. The results indicate that this network significantly improves the detection performance.

## 2 Related work

### 2.1 Multi-scale feature representations

Multi-scale feature fusion detection is crucial for improving traditional object detection performance. FPN combines the multi-scale feature maps from deep convolutional networks, which make each layer rich in information. PANet [19] based on the feature pyramid network adds a bottom-up path augmentation and progressively enhances low-level feature maps to higher levels through a series of lateral connections. NAS-FPN [31] achieves better detection results through neural architecture search. TridentNet [32] constructs a parallel multibranch network using dilated convolutions with different dilation rates, enabling the network to efficiently generate specific feature maps for objects of varying scales. FPG [33] introduces a deep multipath feature pyramid network, representing feature scale space as a regular grid of parallel paths with multidirectional lateral connections for feature fusion. BiFPN [22] removes nodes with only one input edge and adds extra edges between input and output nodes at the same level. It treats each bidirectional path as a feature network layer to optimize cross-scale connections and introduces learnable weights to determine the importance of different input features, thereby enhancing feature fusion effectiveness. DAMO-YOLO [34] designs an efficient re-parameterized generalized feature pyramid network (RepGFPN), improving on FPN by more effectively fusing multi-scale features. AFPN [35] introduces a progressive fusion strategy, which gradually integrates features from the bottom, middle, and top layers into the object detection process. This progressive fusion approach helps reduce the semantic gap between different levels of features, improving feature fusion effectiveness and enabling the detection model to better adapt to varying levels of semantic information.

Due to the significant scale variance among objects in drone imagery, different-sized objects exhibit notable differences in visual features. This scale variation poses challenges for traditional object detection methods when dealing with objects of different scales. Consequently, multi-scale feature fusion techniques are also frequently employed in drone object detection to effectively integrate feature information from various scales. This approach allows the model to better capture the characteristics of both small and large targets, thereby significantly enhancing detection performance and improving the model's adaptability and accuracy in complex scenarios.

In the context of drone image object detection, traditional multi-scale feature fusion methods struggle to effectively address this challenge. To mitigate this issue, SPD-YOLO [26] introduces a small object detection layer to improve the detection performance of small-sized objects. Drone-YOLO [25] incorporates a small object detection layer with richer contextual information into FPN and designs a sandwich-fusion module to optimize the network head's ability to identify and classify object positions. SCA-YOLO [27] and DMA-YOLO [24] integrate small object detection layers and bidirectional skip connections into their models to obtain richer feature information and enhance the model's sensitivity. FE-YOLO [29] proposes a feature enhancement module (FEM), which is integrated into the FPN to leverage deep contextual information from the FPN to guide fine-grained shallow-resolution features, thus enhancing the representation of object features. LUD-YOLO [36] introduces AFPN to alleviate the feature degradation problem during feature propagation and interaction. LODNU [37] presents an adaptive scale-weighted feature fusion method to achieve optimal combinations of different feature layers in PAN, enhancing detection performance for multi-scale targets.

## 2.2 Attention mechanism

In recent years, attention mechanisms have been widely applied in computer vision and have been proven to perform excellently in many computer vision tasks. The main idea of the SE module [38] is to enhance the network's ability to model interchannel dependencies through the squeeze-and-excitation mechanism. The core innovation lies in adaptively recalibrating the feature responses of each channel, significantly improving the network's feature representation capability. This effectively enhances the model's generalization across different datasets. The SE module brings a significant performance boost with minimal computational cost.

SENetV2 [38] introduces an improved SENet architecture by incorporating a new module called squeeze aggregated excitation (SaE) to enhance the network's representation capabilities. This module, combined with the operations of the SE module, strengthens the network's global representation learning through a multibranch fully connected layer. Experiments show that this module outperforms SE module.

GAM [39] aims to address the issue of insufficient information retention in traditional attention mechanisms across channel and spatial dimensions by designing a mechanism that reduces information loss and amplifies global dimension interaction features.

The core idea of CBAM [40] is to enhance the network's representation capability by focusing on important features and suppressing unnecessary ones. The module first applies channel attention to focus on "important" features and then applies spatial attention to focus on the "important locations" of these features. In this way, CBAM effectively helps the network focus on crucial information in the image, enhancing feature representation.

ECA [41] avoids the dimensions reduction operation in channel attention modules by adopting a local cross-channel interaction strategy, using 1D convolution to achieve efficient channel attention computation. This approach maintains performance while significantly reducing model complexity. By adaptively selecting the convolution kernel size, it determines the coverage of local cross-channel interaction. The ECA module offers higher efficiency and better performance compared to other attention modules with minimal parameters and low computational cost.

SimAM [42] proposes a conceptually simple yet highly effective attention module that optimizes an energy function to determine the importance of each neuron.

MLCA [43] introduces a hybrid local channel attention mechanism that combines local and global features, as well as channel and spatial feature information. Overall, the MLCA module enhances the network's ability to capture useful features while maintaining computational efficiency by combining channel and spatial attention at both local and global levels to improve accuracy.

## 3 Method

In this section, we provide a detailed description of the proposed method. Our objective is to design an lightweight and efficient object detection framework, primarily aimed at UAV-based object detection. We have made improvements on YOLOv8n. And the model is improved based on the characteristics of UAV images, focusing on the following aspects: (1) to enhance the detection performance for small objects, we incorporated shallow features, allowing small targets to be better detected and optimized by the shallow detection head. (2) Based on an analysis of the characteristics of UAV images and relevant datasets, we improved the PAFPN structure of YOLOv8. This modification not only reduces the model's parameter count and computational load but also increases detection accuracy. (3) The OFEM is proposed to integrate features from different scales for better feature representation. (4) To increase the network's focus on regions of interest, we introduced a novel region spatial-channel attention mechanism. The structure of the algorithm is shown in Fig. 2.

### 3.1 Small-size object detection layer

Shallow feature maps typically have smaller receptive fields and less overlapping receptive field regions, providing better positional and detail information. This ensures that the network can capture more details, making it more suitable for detecting smaller objects in images. Additionally, shallow features can effectively compensate for information lost during down-sampling in the neck part, ensuring the preservation of contextual information. On the other hand, features extracted by deeper layers are closer to the output and contain more semantic information, capturing more holistic information about the image. However, due to the low resolution of small objects, their ability to perceive details is poor. Therefore, optimizing convolutional networks and improving the utilization of feature maps of different sizes are crucial.

To improve the performance of the YOLOv8 model in small object detection, we integrate shallow features. The standard YOLOv8 model's feature maps have a maximum resolution at the P3 layer, with a size of 80×80, which is an 8x down-sampling result of the original size, used to detect targets larger than 8×8. This may unintentionally filter out some small objects, preventing them from being matched, thereby reducing the model's ability to learn small objects. To increase the learning capacity for small objects and enhance the network's learning ability for small objects, we adopted a method of adding the P2 feature map output to the

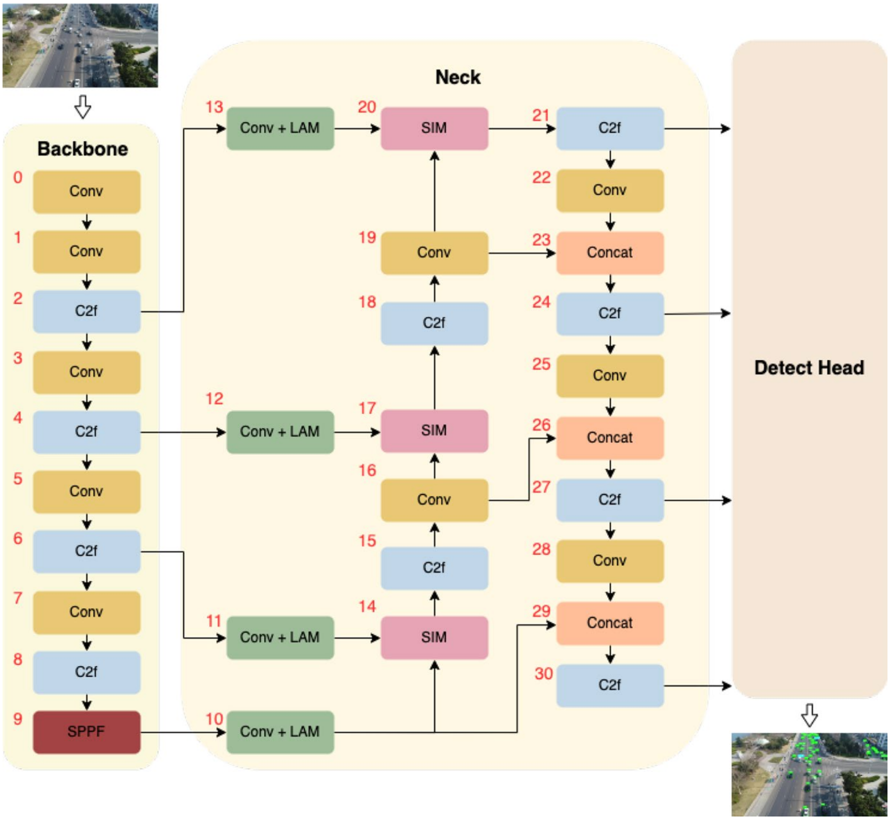
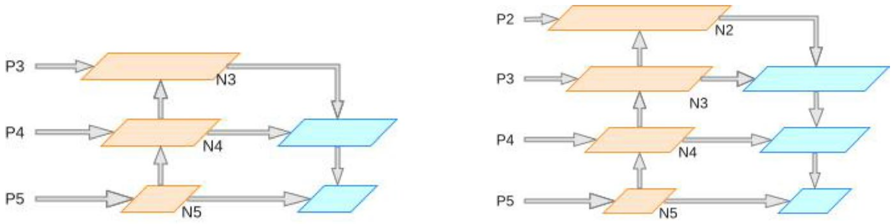


**Fig. 2** The main architecture of the algorithm LightUAV-YOLO in this paper

neck structure. Assuming the input image size is 640×640, the P2 layer feature map is obtained through 4x down-sampling, resulting in a size of 160×160. In this case, each feature point corresponds to a 4×4 receptive field in the input image, which can better detect small objects and provide information to other layers during subsequent feature fusion.

### 3.2 Lightweight and efficient neck (LW neck)

As depicted in Fig. 3, the neck layer of the traditional YOLOv8 adopts the PAFPN structure consisting of three detection layers. To enhance the detection of small objects in the dataset, we introduce the feature maps from shallow networks. The feature extraction process of the YOLOv8 backbone undergoes four down-sampling processes, with the number of channels doubling after each down-sampling. The deepest layer's channels are four times that of the shallow layers, and these feature layers are directly fed into the FPN structure. Consequently, this unavoidably complicates the subsequent multi-scale feature fusion process and significantly increases



(a) original YOLOv8 neck structure

(b) improved YOLOv8 neck structure

**Fig. 3** Diagram of the Object Detection Layer Structure. (**a**) is example diagram of YOLOv8 neck information fusion structure. (**b**) is the YOLOv8 network structure after adding a small object detection layer

the network's parameter count and computational overhead. Therefore, in order to strike a balance between model performance and accuracy, we analyze the model structure from the following aspects.

Shallow features have a smaller receptive field and higher spatial information resolution, and deep features have a larger receptive field and contain rich semantic information. Additionally, deep features correspond to lower-resolution feature maps, with each pixel having a larger receptive field, allowing them to capture more information about medium and large objects. As shown in Fig. 9, our analysis for common UAV images reveals a fact that these images contain a significant proportion of small-sized objects. This indicates that, under the current circumstances, the role of deep features in capturing targets is significantly diminished. However, they require substantial computational cost, which severely impacts the model's performance.

To address the aforementioned issues, as illustrated in Fig. 2, convolutional layers are added at the 10th, 11th, 12th, and 13th positions. By adjusting the channel dimensions using 1 × 1 *Conv*2*d*, the dimensions (number of channels, denoted as d) of the feature map are fixed. Referring to [18], we set d = 256. Since the network width hyperparameter of YOLOv8n is 0.25, the actual value of d is 64. Although the feature dimension is same as [18], our purpose differs. We expand the dimensions of the P2 layer from 32 channels to 64 channels and reduce the P5 and P4 layers to 64 dimensions, maintaining a reasonable number of channels for the feature pyramid network input to balance the importance of different feature layers. This allows the model to better capture detection details from shallow features to improve detection performance while retaining sufficient deep semantic features.

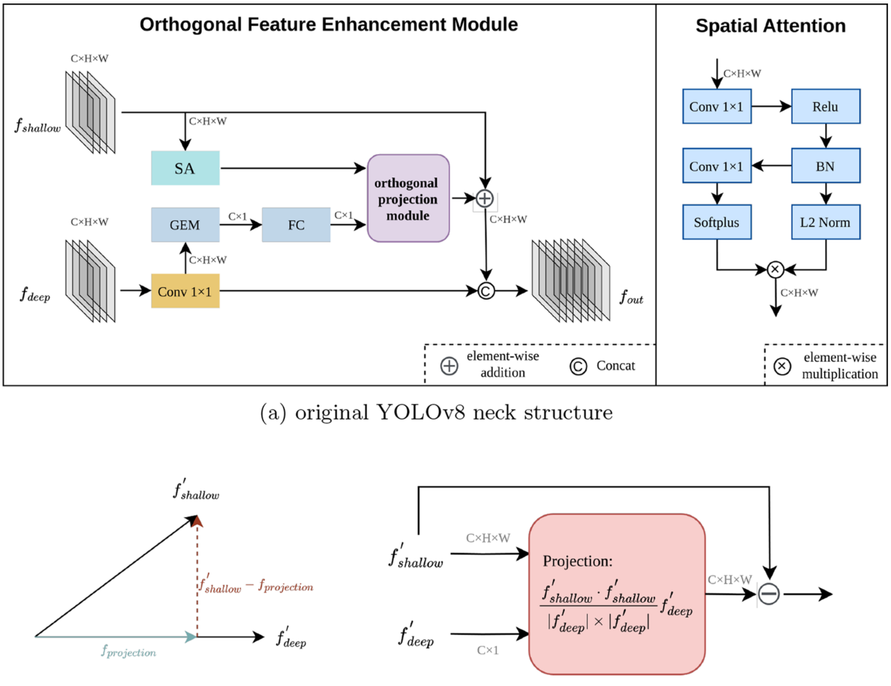
### 3.3 Orthogonal feature enhancement module (OFEM)

In the current FPN structure, feature fusion between different feature layers is achieved through the concatenation operation. As shown in Fig. 3, the P5 feature is derived from the P4 layer through deep sampling, and the N4 feature is obtained by concatenating the upsampled P5 feature with the P4 layer feature.

The FPN simply upsamples deep features and concatenates them with shallow features. This approach has the two limitations. Firstly, it fails to adequately express multi-scale features and introduces a large number of irrelevant contextual information. Secondly, this process can degrade significant local information during subsequent propagation and interaction, which is detrimental to the detection of targets. To address this issue, this paper proposes an orthogonal feature enhancement module (OFEM). We replace the concatenation operation with the OFEM for fusion between different feature layers, ensuring that the fusion of features at different scales in the feature pyramid is no longer a simple concatenation.

The OFEM inspired by DOLG [44] but different from DOLG [44]. Its purpose is to eliminate the redundant part of global information and magnify the difference between local features and global features. But our method focuses on enhancing the expression of feature fusion through orthogonal fusion. The structure of the OFEM is illustrated in Fig. 4a. The OFEM is composed by two branch.

On the one hand, the shallow features are input to the first branch and undergo processing through a *SA* module. The structure of the SA module is shown in Fig. 4a. Firstly, the input features $f_{\text{shallow}} \in \mathbb{R}^{C \times H \times W}$ undergo a $1 \times 1$ convolution for feature extraction. Then, the output features are divided into two branches: one is used for L2 normalization as the activation features, while the other is passed through a $1 \times 1$ convolution again, followed by a *Softplus* activation to generate a



(a) original YOLOv8 neck structure

(b) orthogonal projection theory

(c) orthogonal projection module

**Fig. 4** The structure of OFEM. $f_{\text{shallow}}$ represents shallow features, $f_{\text{deep}}$ represents deep features

attention map. Finally, the activation features and the attention map are multiplied element-wise to serve as the output features $f'_{\text{shallow}} \in \mathbb{R}^{C\times H\times W}$.

On the other hand, the deep features $f'_{\text{deep}} \in \mathbb{R}^{C\times H\times W}$ are input to the second branch and then generates a deep feature descriptor by incorporating a generalized mean pooling (GEM) mechanism [45] on the features extracted by the corresponding $1 \times 1$ convolution. And then, we employed a fully connected layer (FC) to perform a mapping of the same dimensionality to get $f'_{\text{deep}} \in \mathbb{R}^{C\times H\times W}$.

Next, $f_{\text{projection}}$ is calculated by $f'_{\text{shallow}}$ and $f'_{\text{deep}}$ as shown in Fig. 4c. The formula is as follows:

$$f_{\text{projection}}\left(f'_{\text{deep}}, f'_{\text{shallow}}\right) = \frac{f'_{\text{shallow}} \times f'_{\text{deep}}}{|f'_{\text{deep}}|^2} \times f'_{\text{deep}} \tag{1}$$

We can separate components of the shallow features orthogonal to the deep features from the shallow features, as shown in Fig. 4b. Through the above steps, the components of shallow features that eliminate redundant components are obtained. Then, the information contained in the shallow features is enhanced by adding the information back into the shallow features. The formula is as follows:

$$f_{\text{out}} = Concat\left[f_{\text{deep}}, \left(f'_{\text{shallow}} - f_{\text{projection}}\right) + f_{\text{shallow}}\right] \tag{2}$$

### 3.4 Local attention module (LAM)

In the FPN structure, while the fusion of multi-scale features enables the network to acquire rich information, it also introduces some redundant contextual information. This redundant context may include many regions unrelated to objects, leading to a lack of accurate attention to key features by the model and resulting in performance degradation when dealing with complex scenes. Therefore, we alleviate this issue by introducing attention mechanisms. These attention mechanisms aid in enhancing the network's focus on important features, thereby strengthening the model's performance and robustness.

Traditional channel attention mechanisms like SE and ECA focus solely on interchannel relationships, neglecting spatial information within each channel. CBAM models information across both channel and spatial dimensions. However, UAV images often contain numerous small objects, and the pixels corresponding to these small object features are very limited, making it difficult for standard spatial attention mechanisms to focus on specific key areas. MLCA introduces the concept of local channel attention and incorporates spatial information into each local area through a partitioning approach. Although this improves upon some limitations of SE, there are still certain constraints.

By conducting a visual analysis of the UAV image dataset, we observed that although the targets are small, they often cluster together to form local regions. As shown in Fig. 5, the yellow grid divides the image into several equally sized areas,

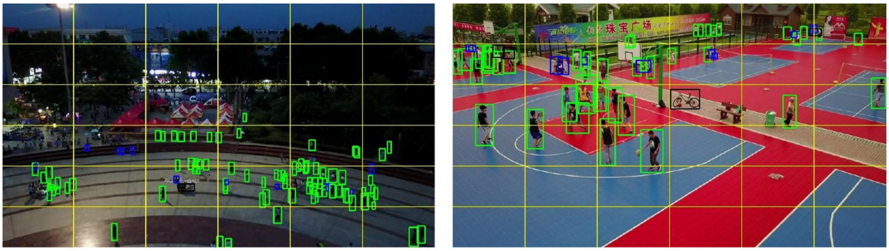


**Fig. 5** Visualization of local areas

with some regions containing a large number of detection instances. In this case, we divide the image into 36 local areas here. In the application scenarios of drones where small targets occupy a larger proportion, pixel-level spatial attention is inaccurate. Therefore, the model should to focus on more important areas rather than individual pixels. Based on the above analysis, we designed the local attention module. As illustrated in Fig. 6, this module initially employs a local channel block, and then followed by a local coordination attention block.

### 3.4.1 Local channel attention block

The algorithm introduced a local channel attention called MLCA [43], which divides images into multiple regions to calculate the importance of each region at the channel level. The module initially acquires local average and global average information through *AdaptiveAvgPool2d*, respectively. Then, it employs convolutional operations to learn dependencies among local channels and computes weights for different regions across channels via the *Sigmoid* function. The weight coefficient be defined as $W_l$. Given the feature $f_{\text{in}} \in \mathbb{R}^{C\times H\times W}$, the formula for $W_l$ is as follows:

$$W_l\left(F_{in}\right) = \sigma\left(UnAvgPooling\left(Conv\left(AvgPooling\left(F_{in}\right)\right) \oplus Conv\left(AvgPooling\left(F_{in}\right)\right)\right)\right) \tag{3}$$

Afterward, the obtained weight coefficients $W_l$ are multiplied with the features to derive scaled features $F'$. The attention mechanism is illustrated in Fig. 7. ⊗ denotes multiplication by element and ⊕ denotes multiplication by element.

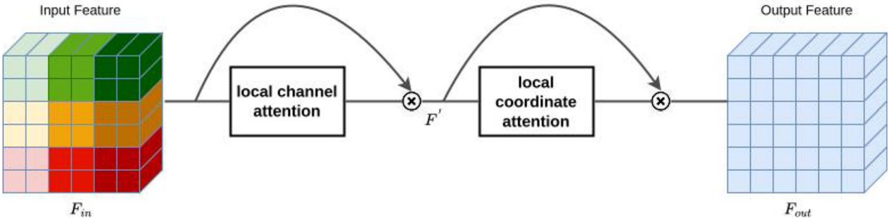


**Fig. 6** Flowchart of the LAM

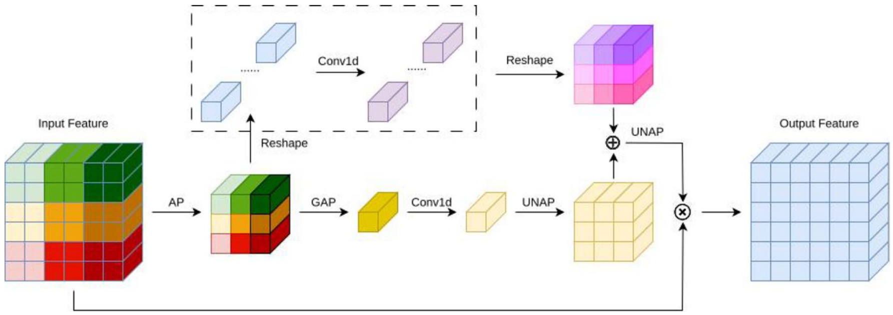


**Fig. 7** Flowchart of the local channel attention

$$F^{'} = W_l(F_{in}) \otimes F_{in}$$

### 3.4.2 Local coordination attention block

The local coordination attention block captures features along vertical and horizontal directions by performing 1D global pooling on the local feature tensor in two spatial directions, preserving precise positional information and capturing long-range dependencies. The feature maps in these two directions are independently encoded into direction-aware and position-sensitive attention maps. These attention maps are then multiplied with the input feature map to highlight the representation of objects of interest. Local coordination attention block allows the model to more accurately locate and recognize objects of interest.

Given the feature $f^{`} \in \mathbb{R}^{C' \times H' \times W'}$, which is derived by $F^{`} \in \mathbb{R}^{C \times H \times W}$ through *AdaptiveAvgPool2d*, features from the horizontal and vertical directions are obtained through pooling operations with kernels of size (H, 1) and (1, W), respectively. Thus, the output of the $c$-th channel at height $h$ and the $c$-th channel at width $w$ can be written as in Eqs. (4), (5)

$$z_c^h = \frac{1}{W} \sum_{0 \le i \le W} x_c(h, i) \tag{4}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j \le H} x_c(j, w) \tag{5}$$

Secondly, the two features be concatenated before learning positional dependencies through *Conv*2*d*, which contributes to more accurate object localization.

$$f_{intermediate} = \delta\left(Conv2d\left(Concat\left[z^h, z^w\right]\right)\right) \tag{6}$$

Next, $f_{intermediate}$ is split into two separate tensors $F_h$ and $F_w$ to get the weight coefficient $w_h$ and $w_w$ in both horizontal and vertical directions.

$$w_h = \sigma\left(Conv2d\left(F_h\right)\right) \tag{7}$$

$$w_w = \sigma\left(Conv2d\left(F_w\right)\right) \tag{8}$$

Finally, the weights are *UnAvgPooling* and multiplied with $F'$ to obtain the scaled feature map.

$$F_{out} = F' \otimes UnAvgPooling\left(w_h\right) \otimes UnAvgPooling\left(w_w\right) \tag{9}$$

The local coordination attention structure is shown in the Fig. 8

## 4 Experiments

The proposed model was evaluated on the Visdrone2021 and UAVDT datasets. This section describes the datasets and experimental setup used to evaluate the performance of our algorithm compared to other state-of-the-art methods. Finally, we conducted a series of ablation studies and visualized the experimental results.

### 4.1 Datasets

We evaluated our model on widely used benchmarks, including the Visdrone2021 and UAVDT datasets, which are designed for object detection in aerial drone photography. The Visdrone2021 dataset, developed by a team at the Machine Learning and Data Mining Laboratory of Tianjin University, captures various scenes from daily life and includes 10 categories. This comprehensive benchmark dataset consists of drone-captured images from 14 different cities across China, making it one of the most extensive and complex datasets for aerial drone photography in the country. It covers various altitudes, weather, and lighting conditions, and includes numerous objects with varying degrees of occlusion and deformation. The dataset contains 6471 training images, 548 validation images, and 3190 testing images, with 1580 images in the challenging test subset. The image categories include cars, pedestrians, buses, bicycles, tricycles, awning

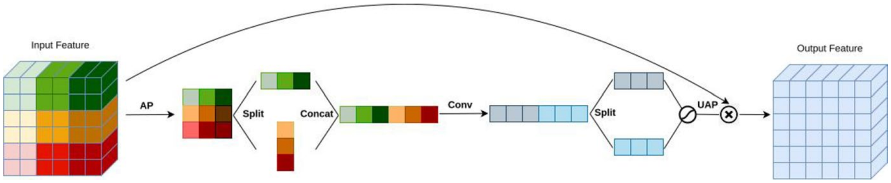


**Fig. 8** Flowchart of the local coordination attention

tricycles, trucks, vans, and people, totaling 2.6 million labels. The Visdrone2021 dataset reflects the general scenarios encountered in real-world drone applications, aligning well with the research context and objectives of this study. Consequently, we conducted real-time testing and ablation studies on this dataset.

The UAVDT dataset is suitable for vehicle detection and tracking tasks. It consists of 50 videos, with 23,829 training images and 16,580 testing images for detection tasks, encompassing three vehicle categories. Following prior work [32, 33, 49], the training and testing sets are derived from different videos, ensuring that all images from a single video are included in only one of these sets. Specifically, the training set includes images from 31 videos, while the test set includes images from 19 different videos.

Small objects are defined as having an area smaller than 32 × 32 pixels, medium objects have an area between 32 × 32 pixels and 96 × 96 pixels, and large objects have an area greater than 96 × 96 pixels. As shown in Fig. 9, we plotted the proportions of different-sized instances in both datasets, revealing that small objects constitute more than half of the instances.

## 4.2 Implementation details

All our models are trained and tested using NVIDIA A40 GPU, equipped with 48GB of memory. Our model implementation is based on the Pytorch 1.12.1 deep learning framework, using Python 3.9.18 as the programming language, and the operating system is Ubuntu 22.04. We use the YOLOv8n configuration to set hyperparameters. During the training process, input images are uniformly resized to 640 × 640, and the optimization is performed using the SGD optimizer. The initial learning rate is set to 0.01, with a weight decay coefficient of 0.0005. Throughout all experiments, we do not use pretrained weights.

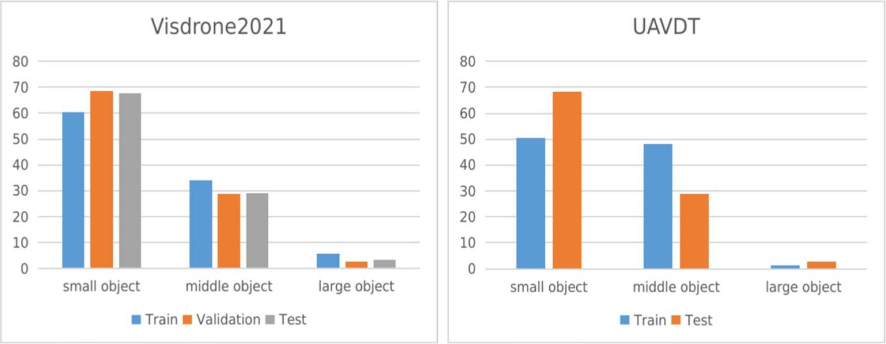


**Fig. 9** The proportion of instances of different sizes in the data set. The chart on the left shows the distribution of the Visdrone2021 dataset, and the chart on the right shows the distribution of the UAVDT dataset. It can be seen from the figure that the proportion of small objects is relatively high

### 4.3 Valuation index

The evaluation metrics include mean Average Precision (mAP), Average Precision (AP), Precision (P), and Recall (R). The formulas for calculating Precision (P) and Recall (R) are as follows:

$$P = \frac{TP}{TP + FP} \tag{10}$$

$$R = \frac{TP}{TP + FN} \tag{11}$$

In these equations, TP represents the number of correctly predicted positive samples, FP represents the number of incorrectly predicted positive samples, and FN represents the number of incorrectly predicted negative samples. The formulas for calculating Average Precision (AP) and mean Average Precision (mAP) are as follows:

$$AP = \int_0^1 p(x)dx \tag{12}$$

$$mAP = \frac{1}{K}\sum_{i=1}^{K} AP_i \tag{13}$$

The parameter $K$ represents the number of classes, and AP is the average precision for each class.

GFLOPs, a unit of measure for the calculating speed of a computer equal to one billion floating-point operations per second, is used to measure the computational complexity of training the model. The parameter value indicates the number of model parameters, which is used to assess the consumption of computational memory resources. FPS, or Frames Per Second, represents the number of images the model can detect per second, used to evaluate the real-time performance of the model. FPS is directly related to the resolution of the detected images. Generally, under the same model and operational environment, the higher the input image resolution during detection, the lower the FPS.

### 4.4 Experiments on Visdrone2021

To validate the effectiveness of the algorithm in detecting various targets in UAV images, we conducted a comparative analysis with various state-of-the-art object detection algorithms on the Visdrone2021 test dataset and the Visdrone2021 val dataset. The input image resolution was set to $640 \times 640$, and we performed 300 training epochs on the Visdrone2021 dataset. The comparison results of different methods on the Visdrone2021 dataset are presented in Table 1. Table 1 shows the detection results of our method and other detectors. Although our method did not achieve the highest precision in several individual categories, it achieves the highest

**Table 1** Comparison of different algorithms on the Visdrone2021 test dataset

| Method | Object category | | | | | | | | | | mAP50 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | PED | PER | BC | Car | Van | Truck | TRI | ATRI | Bus | MO | |
| Fast R-CNN [46] | 21.4 | 15.6 | 6.7 | 51.7 | 29.5 | 19 | 13.1 | 7.7 | 31.4 | 20.7 | 21.7 |
| Faster R-CNN [46] | 20.9 | 14.8 | 7.3 | 51 | 29.7 | 19.5 | 14 | 8.8 | 30.5 | 21.2 | 21.8 |
| Cascade R-CNN [46] | 22.2 | 14.8 | 7.6 | 54.6 | 31.5 | 21.6 | 14.8 | 8.6 | 34.9 | 21.4 | 23.2 |
| RetinaNet [46] | 13 | 7.9 | 1.4 | 45.5 | 19.9 | 11.5 | 6.3 | 4.2 | 17.8 | 11.8 | 13.9 |
| CenterNet [47] | 22.6 | 20.6 | 14.6 | 59.7 | 24 | 21.3 | 20.1 | 17.4 | 37.9 | 23.7 | 26.2 |
| DMNet [17] | 28.5 | 20.4 | 15.9 | 56.8 | 37.9 | 30.1 | 22.6 | 14 | 47.1 | 29.2 | 30.3 |
| HRDet+ [48] | 28.6 | 14.5 | 11.7 | 49.4 | 37.1 | 35.2 | 28.8 | 21.9 | 43.3 | 23.5 | 28 |
| MSC-CenterNet [17] | 33.7 | 15.2 | 12.1 | 55.2 | 40.5 | 34.1 | 29.2 | 21.6 | 42.2 | 27.5 | 31.1 |
| YOLOv3-LITE [49] | 34.5 | 23.4 | 7.9 | 70.8 | 31.3 | 21.9 | 15.3 | 6.2 | 40.9 | 32.7 | 28.5 |
| YOLOv8n | 20.8 | 11.3 | 5.3 | 65.8 | 30.9 | 30.3 | 11.5 | 11.6 | 48.1 | 21.4 | 25.7 |
| **Ours** | **30.5** | **18.9** | **9.9** | **73.5** | **37.9** | **33.9** | **16.6** | **16.6** | **52.7** | **30.3** | **32.1** |

The bold value highlight our experimental results

mAP. Notably, our method achieved the highest accuracy of 73.5% in the Car category, significantly surpassing other models, demonstrating outstanding advantages in UAV object detection tasks.

In addition, we also compared the performance of the proposed method with other lightweight UAV object detection algorithms on the Visdrone2021 val dataset, focusing on detection accuracy and model size. The results are shown in the Table 2, LODNU [37], Drone-YOLO(nano) [25], and LUDY [36] are also three lightweight drone object detection models. Due to the lack of open-source code for certain papers, we could only obtain a part of the experimental data from the papers. Our method achieves excellent detection results and has a smaller parameter count compared to both.

### 4.5 Ablation studies

To validate the effectiveness of the proposed improvements at each stage of object detection, a series of ablation experiments were conducted using the Visdrone2021 dataset. The ablation experiments utilized YOLOv8n as the baseline algorithm and employed mean Average Precision (mAP), model size, the number of parameters, and the number of floating-point operations as evaluation metrics. Additionally, frames per second (FPS) was measured to evaluate the model's speed performance. To ensure data accuracy, the FPS measurements were conducted under identical conditions.

Additionally, Fig. 10 visually presents the performance of various metrics of the model, as well as the overall performance of the entire model. A, B, C, D, and E correspond to the five models listed in Table 3.

**Table 2** Comparison of different algorithms on the Visdrone2021 val dataset

| Method | mAP50 | Object category | | | | | | | | | | Params (M) | FLops (G) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PED | PEO | BC | **Car** | Van | Truck | TRI | **ATRI** | Bus | MO | | |
| YOLOv4-tiny | 19.7 | 33.1 | 16.8 | 1.2 | 72.6 | 4.4 | 15.3 | 1.1 | 1.0 | 25.2 | 26.4 | 6.1 | 7.0 |
| YOLOv7-tiny | 31.3 | 37.6 | 33.9 | 4.7 | 75.2 | 34.9 | 21.9 | 16.4 | 7.8 | 40.1 | 40.4 | 6.2 | 13.9 |
| YOLOX-s | 36.5 | 40.5 | 35.6 | 11.6 | 76.8 | 39.4 | 32.1 | 24.4 | 11.5 | 48.5 | 44.6 | 5.1 | 15.4 |
| LODNU [37] | 31.4 | – | – | – | – | – | – | – | – | – | – | 8.7 | 9.3 |
| YoloV8-n | 31.4 | 33.6 | 28.1 | 6.7 | 75.5 | 37.8 | 24.8 | 20.4 | 11.4 | 40.6 | 35.4 | 3.0 | 8.1 |
| YoloV8-s | 39.7 | 43.3 | 34.2 | 12.5 | 80.3 | 46.6 | 36.8 | 28.5 | 15.2 | 54.8 | 45.1 | 11.1 | 28.5 |
| Drone–YOLOn [25] | 38.1 | – | – | – | – | – | – | – | – | – | – | 3.1 | – |
| LUDY-n [36] | 35.2 | 36.9 | 29.3 | 9.9 | 77.4 | 48.4 | 31.4 | 22.2 | 13.6 | 49.8 | 39.4 | 2.81 | – |
| LUDY-s [36] | 41.7 | 44.8 | 34.3 | 14.5 | 80.9 | 48.4 | 39.4 | 29.8 | 16.9 | 62.2 | 46.2 | 10.3 | – |
| **Ours** | **39.8** | **47.0** | **38.1** | **11.4** | **81.9** | **46.6** | **32.7** | **25.1** | **13.8** | **53.4** | **47.6** | **2.2** | **18.2** |

The bold value highlight our experimental results

**Table 3** The ablation study results of the algorithm on VisDrone2021 dataset

| Datasets | Method | mAP50 | mAP50:95 | mAP$_s$ | mAP$_m$ | mAP$_l$ | FLOPs(G) | Params(M) | FPS |
|---|---|---|---|---|---|---|---|---|---|
| Val | baseline | 31.4 | 18.4 | 8.7 | 26.2 | 37.5 | 8.2 | 3.0 | 219 |
| | baseline+P2 | 36.9 | 22.3 | 12.2 | 30.2 | 38.2 | 40.8 | 3.8 | 185 |
| | baseline+P2+LW Neck | 39.2 | 23.4 | 13.7 | 31.5 | 37.3 | 17.2 | 2.0 | 150 |
| | baseline+P2+LW Neck+OFEM | 39.8 | 23.7 | 14.1 | 32.1 | 38.1 | 18.1 | 2.2 | 151 |
| | baseline+P2+LW Neck+OFEM+LAM | 39.8 | 24.1 | 14.3 | 31.6 | 39.0 | 18.2 | 2.2 | 143 |
| Test | baseline | 25.7 | 14.5 | 5.5 | 21.2 | 32.7 | 8.2 | 3.0 | 219 |
| | baseline+P2 | 29.5 | 16.8 | 7.4 | 23.8 | 31.8 | 40.8 | 3.8 | 185 |
| | baseline+P2+LW Neck | 30.5 | 17.4 | 8.2 | 24.7 | 32.8 | 17.2 | 2.0 | 150 |
| | baseline+P2+LW Neck+OFEM | 31.7 | 18.1 | 8.6 | 25.7 | 33.9 | 18.1 | 2.2 | 151 |
| | baseline+P2+LW Neck+OFEM+LAM | 32.1 | 18.4 | 8.4 | 25.9 | 34.5 | 18.2 | 2.2 | 143 |

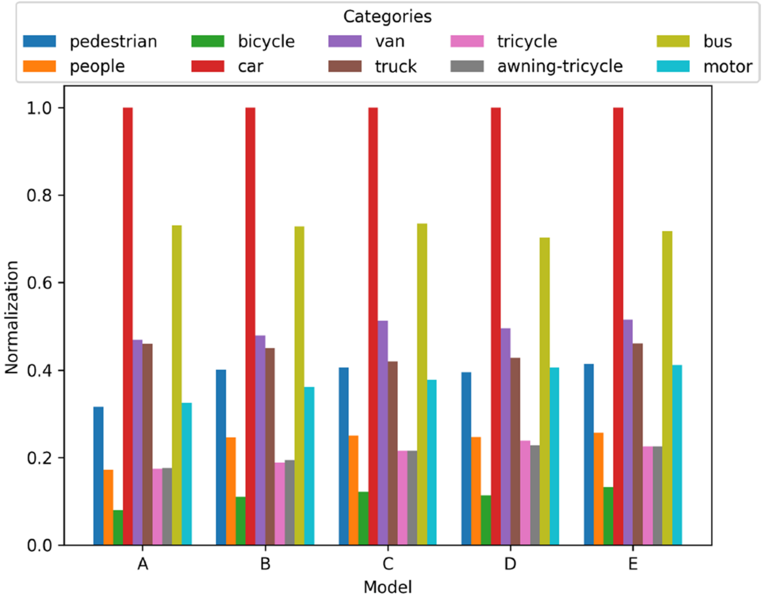


**Fig. 10** The ablation study results of LightUAV-YOLO at VisDrone2021 test dataset, from A to E, correspond sequentially to each row in Table 3

Firstly, adding a small object detection layer to the network significantly improves detection accuracy. However, this also considerably increases the number of parameters and computational load of the model.

Secondly, we designed the lightweight version of the FPN, which not only halved the computational load of the model but also improved the mAP50 by 1%. Additionally, as demonstrated in Fig. 10 and Table 3, the mAP for each category increased except for the truck category, and the mAP for large objects did not obviously decrease. This further confirms our hypothesis that this design allows for more efficient utilization of features.

Moreover, our designed OFEM enhanced the mAP50 by 1.2%. This module improves the model's feature fusion capability, allowing for better utilization of contextual information. Meanwhile, the computational burden of this module is small, with an increase in less than 1 GFLOPs and only 0.2M increase in the number of parameters.

We conduct a visual analysis by comparing the heatmap results generated with and without OFEM to intuitively demonstrate the effectiveness of OFEM. The visualization results are shown in Fig. 11, showcasing various scenarios. It is evident that OFEM not only effectively mitigates interference from complex environments but also enables the model to focus more on the location of the target.

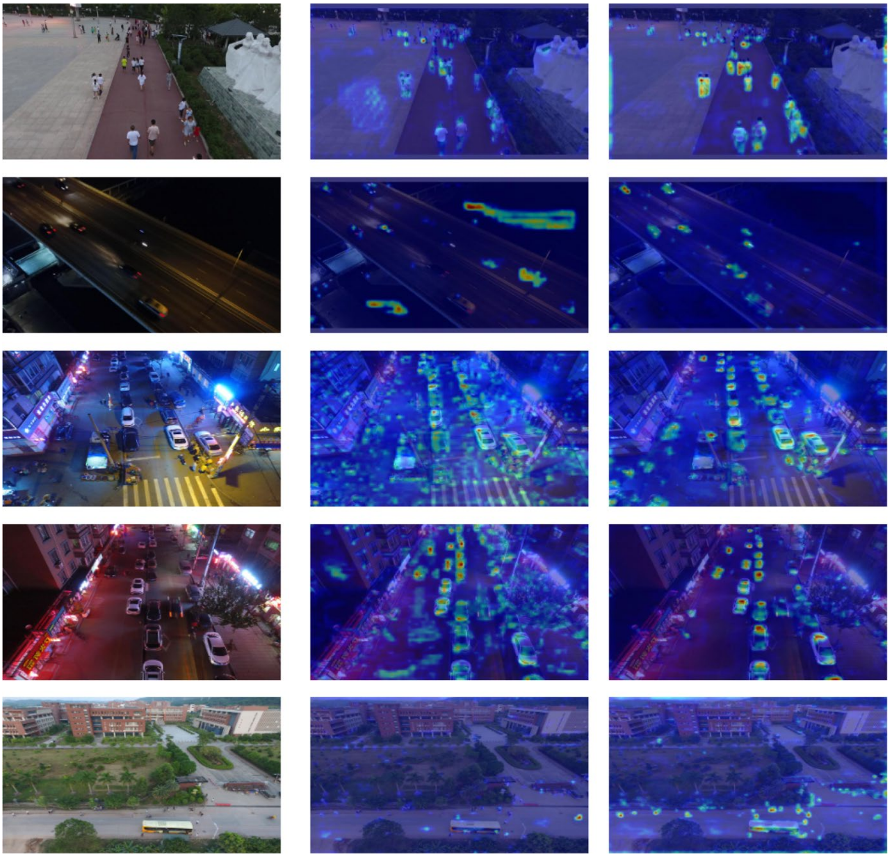


**Fig. 11** Visualization across various scenarios is presented. The first column displays the original images, the second column shows the results without the OFEM, and the third column illustrates the results with the inclusion of OFEM

Finally, our proposed LAM improved the mAP50 by 0.4% and the mAP50:95 by 0.3%, with almost no increase in computational load and parameter count. LAM reduces background interference, resulting in an improvement in mAP50 for most categories.

**Table 4** Comparative with different attention mechanisms

| Method | mAP50 | mAP50:95 | Precision | Recall |
|---|---|---|---|---|
| stage4 | 31.7 | 18.1 | 42.7 | 33.7 |
| stage4+se | 31.5(−0.2) | 18.1 (+0.0) | 41.6 | 33.5 |
| stage4+cbam | 31.8 (+0.1) | 18.2 (+0.1) | 43.3 | 33.9 |
| stage4+mlca | 31.6(−0.1) | 18.2 (+0.1) | 42.2 | 33.9 |
| stage4+lab | 32.1 (+0.4) | 18.4 (+0.3) | 43.5 | 33.8 |

### 4.6 Comparative experiments on different attention mechanisms

To verify the effect of our proposed LAM, we compared the results of three commonly used different attention mechanisms. We added these on the basis of the fourth stage mentioned above Table 3. The experimental results are shown in Table 4.

As shown in the Table 4, the SE module considers the importance of different feature layers solely from the channel dimension, which can easily lead to the loss of important features or the enhancement of some background interference. This results in a decrease in both accuracy and recall, with mAP50 dropping by 0.2%. The CBAM module models from both the channel and spatial dimensions, making its design complex, with large parameters and high computational overhead, rendering it unsuitable for lightweight networks. Additionally, UAV images contain a large number of small-sized targets, and CBAM's modeling of every pixel in the spatial dimension can lead to scattered attention, making it difficult to capture the truly critical areas. The MLCA integrates spatial information into the local channel attention, which partially mitigates the shortcomings of the SE module but does not fully exploit the importance of spatial information, resulting in performance similar to the SE module.

Figure 12 illustrates the comparison between the results of object detection networks using various attention mechanisms and the heatmap analysis on the VisDrone dataset. The heatmaps were generated using Grad-CAM [50]. The results demonstrate that the proposed attention method, LAM, is feasible and outperforms other attention methods in current cases.

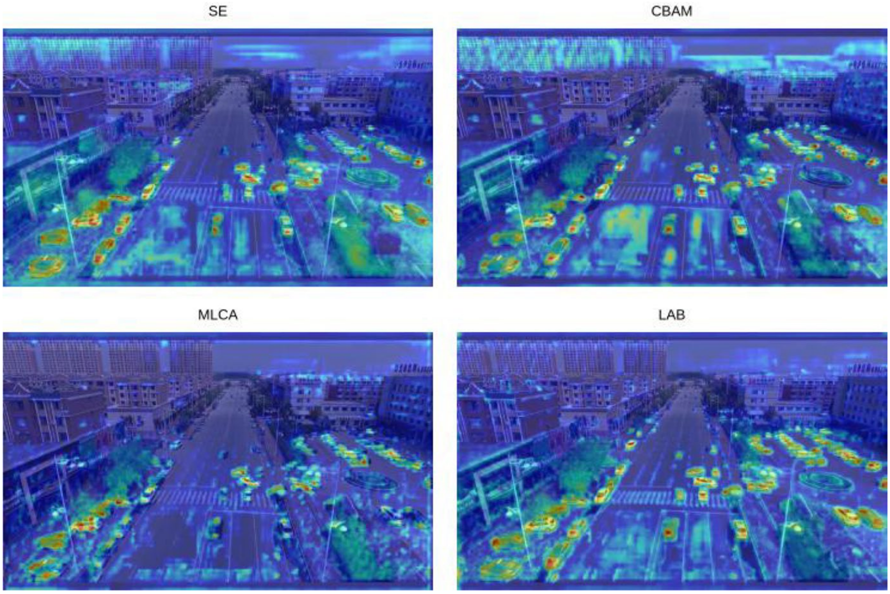


**Fig. 12** Visual analysis of SE, CBAM, MLCA, and LAM mechanism thermogramed by Grad-CAM

**Table 5** Comparison of different algorithms on UAVDT datasets

| Method | mAP50 | mAP50:95 |
|---|---|---|
| ClusDet [51] | 26.5 | 13.7 |
| DMNet [51] | 24.6 | 14.7 |
| GDFNet [15] | 26.1 | 15.4 |
| SODNet [15] | 29.9 | 17.1 |
| DSHNet [15] | 30.4 | 17.8 |
| CEASC [52] | 30.9 | 17.1 |
| DCGMF [53] | 31.4 | 18.5 |
| YOLOv8n | 29.4 | 17.5 |
| Ours | 31.3 | 19.4 |

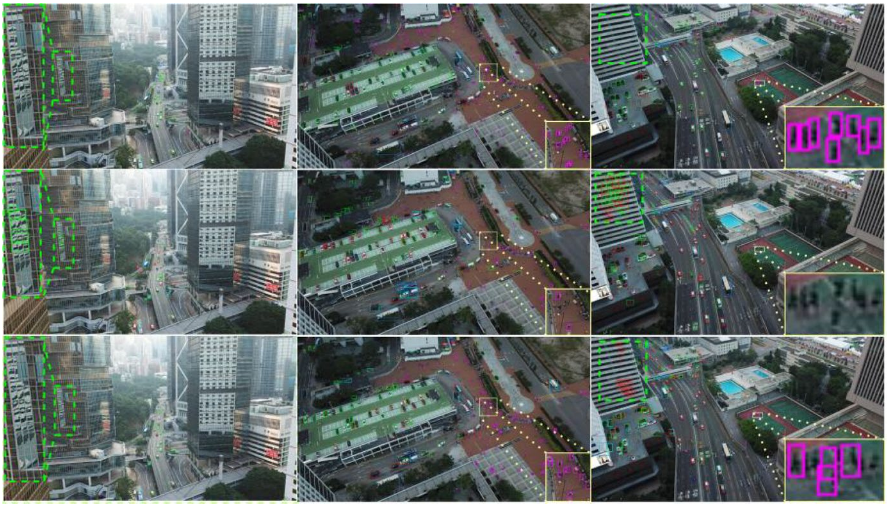


**Fig. 13** Comparison plot of experimental results on the Visdrone2021 test dataset

### 4.7 Experiments on UAVDT

We also conducted experiments on the UAVDT dataset, training for 30 epochs. As shown in the Table 5, our proposed model outperforms the baseline model YOLOv8n by 1.9% in both mAP50 and mAP50:95. Additionally, while our model shows a slight decrease in mAP50 compared to DCGMF, it achieves a 0.9% higher mAP50:95.

### 4.8 Visualization

#### 4.8.1 Visual representation of datasets

As shown in Fig. 13, the first row displays the ground truth results. The second row shows the recognition results of the YOLOv8n algorithm, and the third row presents

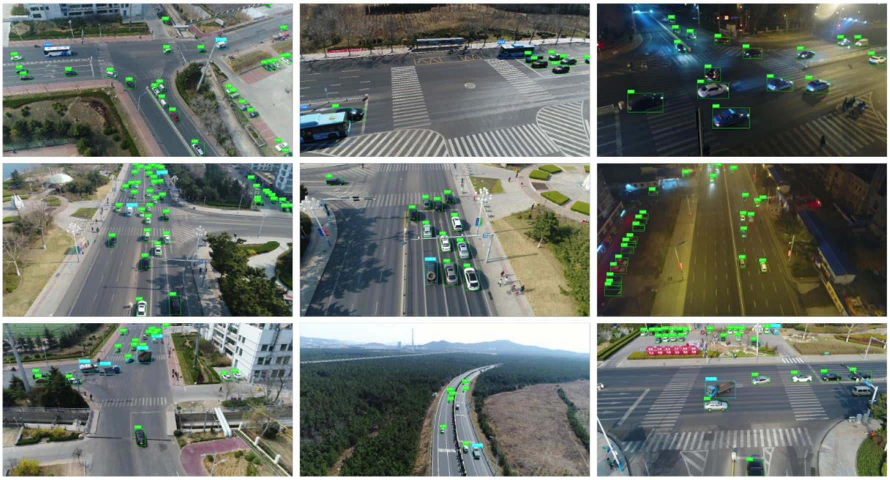
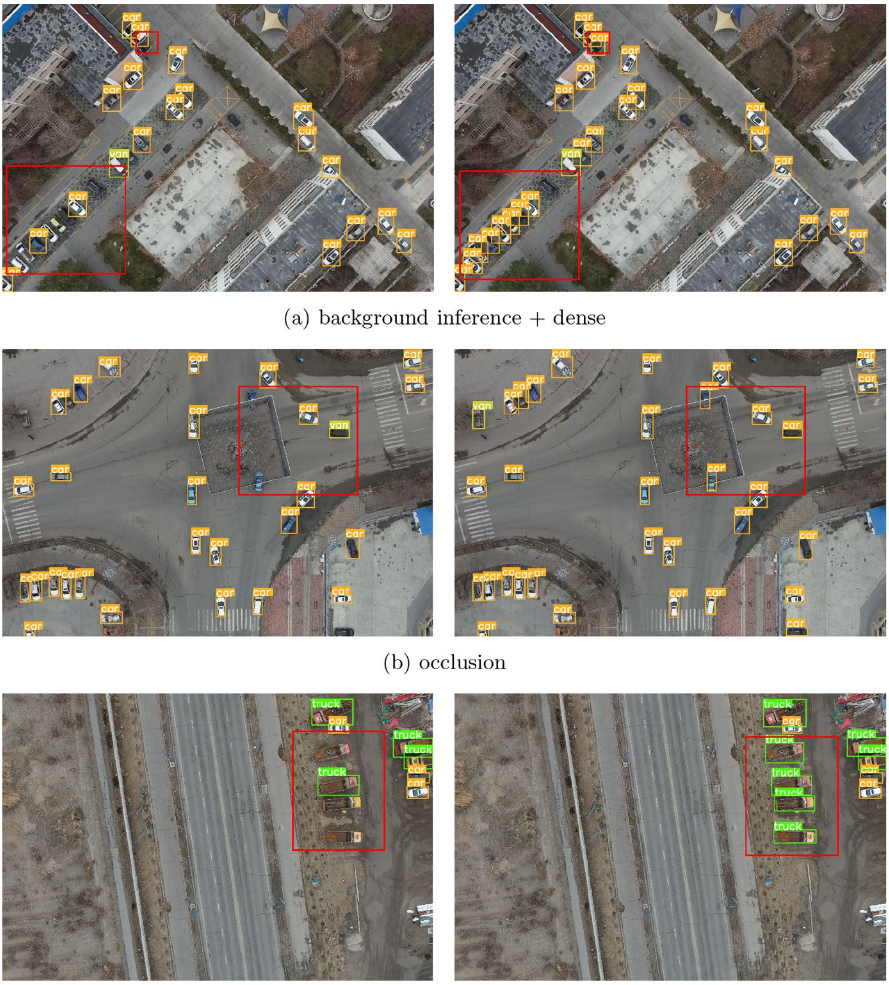


**Fig. 14** LightUAV-YOLO test results on UAVDT

**Fig. 15** Matrice 300 RTK drone shooting live scenes

the recognition results of our LightUAV-YOLO algorithm. The green dashed boxes represent background information. It can be observed that YOLOv8n mistakenly identifies many background elements as targets, whereas our algorithm better distinguishes between foreground and background. The yellow solid boxes indicate small target areas. YOLOv8n has instances of missed detections, while our improved algorithm alleviates this issue.

To more intuitively demonstrate the effectiveness of our proposed method in practical scenarios, we present the detection results on the UAVDT dataset. As shown in the Fig. 14, our method exhibits excellent performance in complex environments. Under various lighting conditions, the improved algorithm successfully detects vehicle targets, maintaining good detection capabilities even in dimly lit scenes. Additionally, our algorithm performs well across different shooting angles, without a decrease in accuracy due to angle variations. Moreover, as illustrated in



(a) background inference + dense

(b) occlusion

(c) background inference

**Fig. 16** Comparison of the detection performance of YOLOv8n and LightUAV-YOLO in a variety of different scenarios captured by the drone. The left column presents the results from YOLOv8n, while the right column displays the detection results from LightUAV-YOLO. The red bounding box shows more obvious contrast situation

the figure, our algorithm shows outstanding detection performance in the presence of occlusions and densely populated target areas.

### 4.8.2 Visual comparison in real world

In order to demonstrate the generality and practicability of the algorithm, we applied the object detection algorithm to images captured by drones. The image data were captured using a Matrice 300 RTK drone, as illustrated in Fig. 15, in Urumqi, Xinjiang. This drone is equipped with Zenmuse P1 which is a high-performance, multifunctional aerial surveying payload. By importing the digital surface model (DSM) in the drone, the drone is enabled to perform terrain-following flight operations at an altitude of 100 ms above ground. We selected several distinct scenarios, including occlusion, background interference and dense vehicle scenes, to compare the detection performance of YOLOv8n and LightUAV-YOLO. The models are trained by Visdrone2021 dataset. As shown in Fig. 16, the left image illustrates the results of YOLOv8n, while the right image displays the results of LightUAV-YOLO. The red bounding box shows more obvious contrast situation in this figures.

It is evident that LightUAV-YOLO demonstrates superior detection performance. Notably, in the presence of vehicles with varying colors and complex backgrounds, YOLOv8n exhibited significant missed detection issues in Fig. 16a, whereas LightUAV-YOLO accurately locates and classifies the targets. Additionally, as shown in Fig. 16b, the YOLOv8n model also encounters missed detection under partial occlusion, while LightUAV-YOLO successfully identifies the target. Finally, in scenarios with complex backgrounds, as shown in Fig. 16c, even for larger targets such as trucks, YOLOv8n fails to detect the objects. And LightUAV-YOLO accurately detects all targets present in the image. Experimental results across various complex scenarios indicate that the proposed LightUAV-YOLO significantly enhances the overall performance of the model.

## 5 Conclusion

This paper proposes the LightUAV-YOLO algorithm to address the challenges of object detection in UAV applications while reducing computational pressure. We made several improvements to the neck structure of YOLOv8. First, we add a small object detection layer to improve the network's learning ability for small targets to address the problem of small object detection difficulty. Secondly, in order to enrich the semantic information of shallow features and improve the fusion effect of features of different scales, designed and implemented the OFEM. Thirdly, the LAM we introduced effectively filters out irrelevant interference information and further improves the robustness of the model. The test results show that LightUAV-YOLO outperforms YOLOv8n with better detection accuracy and fewer parameters. Future work will focus on further improving the effectiveness of the model in actual scenarios.

**Author contributions** Conceptualization, Y.L. and X.L.; methodology, Y.L.; software, X.L. and T.Z.; validation, Y.L. and T.Z.; formal analysis, Y.L. and T.Z.; investigation, T.Z.; resources, G.S.; writing---original draft preparation, Y.L.; writing---review and editing, Y.L. and T.Z.; visualization, Y.L. and T.Z.; supervision, G.S.; project administration, G.S.; funding acquisition, G.S. All authors have read and agreed to the published version of the manuscript.

**Data availability** The datasets on which the study is based were accessed from websites and are available for downloading through the following link: Visdrone2021 dataset (http://aiskyeye.com/visdrone-2020/), UAVDT dataset (https://sites.google.com/view/grli-uavdt).

## Declarations

**Conflict of interest** No potential conflict of interest was reported by the authors.

## References

1. Jia X, Tong Y, Qiao H, Li M, Tong J, Liang B (2023) Fast and accurate object detector for autonomous driving based on improved yolov5. Sci Rep 13(1):9711
2. Teja Y (2023) Static object detection for video surveillance. Multimed Tools Appl 82(14):21627–21639
3. Zhao H, Zhang H, Zhao Y (2023) Yolov7-Sea: Object Detection of Maritime UAV Images Based on Improved yolov7. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 233–238
4. Zhai X, Huang Z, Li T, Liu H, Wang S (2023) Yolo-drone: an optimized yolov8 network for tiny uav object detection. Electronics 12(17):3664
5. 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
6. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1440–1448
7. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. Adv Neural Inform Process Syst 28
8. 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
9. Redmon J, Farhadi A(2017) Yolo9000: Better, Faster, Stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7263–7271
10. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767
11. Bochkovskiy A, Wang C-Y, Liao H-YM(2020) Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934
12. Wang C-Y, Bochkovskiy A, Liao H-YM(2023) Yolov7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7464–7475
13. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: Single Shot Multibox Detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp 21–37 . Springer
14. Diwan T, Anirudh G, Tembhurne JV (2023) Object detection using yolo: challenges, architectural successors, datasets and applications. Multimed Tools Appl 82(6):9243–9275
15. Zhao Q, Liu B, Lyu S, Wang C, Zhang H (2023) Tph-yolov5++: boosting object detection on drone-captured scenarios with cross-layer asymmetric transformer. Remote Sens 15(6):1687
16. Du D, Qi Y, Yu H, Yang Y, Duan K, Li G, Zhang W, Huang Q, Tian Q (2018) The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. In: Proceedings of the European Conference on Computer Vision (ECCV)
17. Cao Y, He Z, Wang L, Wang W, Yuan Y, Zhang D, Zhang J, Zhu P, Van Gool L, Han J (2021) Visdrone-det2021: The Vision Meets Drone Object Detection Challenge Results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 2847–2854
18. Lin T-Y, Dollár 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

19. 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
20. Kong T, Sun F, Tan C, Liu H, Huang W (2018) Deep Feature Pyramid Reconfiguration for Object Detection. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 169–185
21. Pang J, Chen K, Shi J, Feng H, Ouyang W, Lin D (2019) Libra r-cnn: Towards Balanced Learning for Object Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 821–830
22. 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
23. Chen Q, Wang Y, Yang T, Zhang X, Cheng J, Sun J (2011) You Only Look One-Level Feature. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 13039–13048
24. Li Y-l, Feng Y, Zhou M-l, Xiong X-c, Wang Y-h, Qiang B-h (2024) Dma-yolo: multi-scale object detection method with attention mechanism for aerial images. The Visual Comput 40(6):4505–4518
25. Zhang Z (2023) Drone-yolo: an efficient neural network method for target detection in drone images. Drones 7(8):526
26. Zhong R, Peng E, Li Z, Ai Q, Han T, Tang Y (2024) Spd-yolov8: an small-size object detection model of uav imagery in complex scene. The J Supercomput 1–21
27. Zeng S, Yang W, Jiao Y, Geng L, Chen X (2024) Sca-yolo: a new small object detection model for uav images. The Visual Comput 40(3):1787–1803
28. Gong Y, Yu X, Ding Y, Peng X, Zhao J, Han Z (2021) Effective Fusion Factor in fpn for Tiny Object Detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 1160–1168
29. Wang M, Yang W, Wang L, Chen D, Wei F, KeZiErBieKe H, Liao Y (2023) Fe-yolov5: feature enhancement network based on yolov5 for small object detection. J Vis Commun Image Rep 90:103752
30. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, Ke Z, Li Q, Cheng M, Nie W, et al (2022) Yolov6: a single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976
31. Ghiasi G, Lin T-Y, Le QV (2019) Nas-fpn: Learning Scalable Feature Pyramid Architecture for Object Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7036–7045
32. 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
33. Chen K, Cao Y, Loy CC, Lin D, Feichtenhofer C (2020) Feature pyramid grids. arXiv preprint arXiv:2004.03580
34. Xu X, Jiang Y, Chen W, Huang Y, Zhang Y, Sun, X (2022) Damo-yolo: a report on real-time object detection design. arXiv preprint arXiv:2211.15444
35. Yang G, Lei J, Zhu Z, Cheng S, Feng Z, Liang R (2023) Afpn: Asymptotic Feature Pyramid Network for Object Detection. In: 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp 2184–2189 . IEEE
36. Fan Q, Li Y, Deveci M, Zhong K, Kadry S (2024) Lud-yolo: a novel lightweight object detection network for unmanned aerial vehicle. Inform Sci 121366
37. Chen N, Li Y, Yang Z, Lu Z, Wang S, Wang J (2023) Lodnu: lightweight object detection network in uav vision. The J Supercompu 79(9):10117–10138
38. Hu J, Shen L, Sun G (2018) Squeeze-and-Excitation Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 7132–7141
39. Liu Y, Shao Z, Hoffmann N (2021) Global attention mechanism: Retain information to enhance channel-spatial interactions. arXiv preprint arXiv:2112.05561
40. Woo S, Park J, Lee J-Y, Kweon IS (2018) Cbam: Convolutional Block Attention Module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19
41. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q (2020) Eca-net: Efficient Channel Attention for Deep Convolutional Neural Networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 11534–11542

42. Yang L, Zhang R-Y, Li L, Xie X (2021) Simam: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks. In: International Conference on Machine Learning, pp 11863–11874 . PMLR
43. Wan D, Lu R, Shen S, Xu T, Lang X, Ren Z (2023) Mixed local channel attention for object detection. Eng Appl Artif Intell 123:106442
44. Yang M, He D, Fan M, Shi B, Xue X, Li F, Ding E, Huang J (2021) Dolg: Single-Stage Image Retrieval With Deep Orthogonal Fusion of Local and Global Features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 11772–11781
45. Radenović F, Tolias G, Chum O (2018) Fine-tuning cnn image retrieval with no human annotation. IEEE Trans Pattern Anal Mach Intell 41(7):1655–1668
46. Yu W, Yang T, Chen C (2021) Towards Resolving the Challenge of Long-Tail Distribution in UAV Images for Object Detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 3258–3267
47. Albaba BM, Ozer S (2021) Synet: An Ensemble Network for Object Detection in UAV Images. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp 10227–10234 . IEEE
48. Du D, Zhu P, Wen L, Bian X, Lin H, Hu Q, Peng T, Zheng J, Wang X, Zhang Y (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
49. Zhao H, Zhou Y, Zhang L, Peng Y, Hu X, Peng H, Cai X (2020) Mixed yolov3-lite: a lightweight real-time object detection method. Sensors 20(7):1861
50. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-Cam: Visual Explanations from Deep Networks Via Gradient-Based Localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp 618–626
51. Zhang Y, Wu C, Guo W, Zhang T, Li W (2023) Cfanet: efficient detection of uav image based on cross-layer feature aggregation. IEEE Transactions on Geoscience and Remote Sensing
52. 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
53. Shi Y, Wang C, Xu S, Yuan M-D, Liu F, Zhang L (2024) Deformable convolution-guided multiscale feature learning and fusion for uav object detection. IEEE Geoscience and Remote Sensing Letters

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

### Yifan Lyu[1] · Tianze Zhang[2] · Xin Li[1] · Aixun Liu[1] · Gang Shi[1]

✉ Gang Shi
shigang@xju.edu.cn

Yifan Lyu
107552204042@stu.xju.edu.cn

Tianze Zhang
zhangtianze.unimelb@gmail.com

Xin Li
107552204023@stu.xju.edu.cn

Aixun Liu
107552204029@stu.xju.edu.cn

[1] College of Computer Science and Technology, Xinjiang University, Shengli Road,
Urumqi 830017, Xinjiang, China

[2] Faculty of Science, The University of Melbourne, Parkville, Melbourne VIC 3010, Australia