Information Sciences 686 (2025) 121366

Contents lists available at ScienceDirect

Information Sciences

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

LUD-YOLO: A novel lightweight object detection network for unmanned aerial vehicle

Qingsong Fan [a,c], Yiting Li [b,c,*], Muhammet Deveci [d,e,f,*], Kaiyang Zhong [g], Seifedine Kadry [h,i]

[a] State Key Laboratory of Intelligent Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[b] College of Big Data Statistics, Guizhou University of Finance and Economics, Guiyang 550025, China
[c] Key Laboratory of Advanced Manufacturing Technology, Ministry of Education, Guizhou University, Guiyang 550025, China
[d] Department of Industrial Engineering, Turkish Naval Academy, National Defence University, 34942 Tuzla, Istanbul, Turkey
[e] Royal School of Mines, Imperial College London, South Kensington Campus, SW7 2AZ, London, UK
[f] Department of Information Technologies, Western Caspian University, Baku 1001, Azerbaijan
[g] College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou 310058, China
[h] Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon
[i] MEU Research Unit, Middle East University, Amman 11831, Jordan

ARTICLE INFO

Keywords:
Small object detection
YOLOv8
UAV
Deep learning
Feature fusion

ABSTRACT

Autonomous execution of tasks by unmanned aerial vehicles (UAVs) relies heavily on object detection. However, object detection in most images presents challenges such as complex backgrounds, small targets, and obstructions. Additionally, the limited computing speed and memory of the UAV processor affects the accuracy of conventional object detection algorithms. This paper proposes LUD-You Only Look Once (YOLO), a small and lightweight object detection algorithm for UAVs based on YOLOv8. The proposed algorithm introduces a new multiscale feature fusion mode that solves the degradation in feature propagation and interaction through the introduction of upsampling in the feature pyramid network and the progressive feature pyramid network. The application of the dynamic sparse attention mechanism in the Cf2 module achieves flexible computing allocation and content awareness. Furthermore, the proposed model is optimized to be sparse and lightweight, making it possible to deploy on UAV edge devices. Finally, the effectiveness and superiority of LUD-YOLO were verified on the VisDrone2019 and UAVDT datasets. The results of ablation and comparison experiments show that compared with the original algorithm, LUDY-N and LUDY-S have shown excellent performance in various evaluation indexes, indicating that the proposed improvement strategies make the model have better robustness and generalization. Moreover, compared with multiple other popular competitors, the proposed improvement strategies enable LUD-YOLO to have the best overall performance, providing an effective solution for UAVs object detection while balancing model size and detection accuracy.

* Corresponding authors.
E-mail addresses: yitingli.cn@mail.gufe.edu.cn (Y. Li), muhammetdeveci@gmail.com (M. Deveci).

https://doi.org/10.1016/j.ins.2024.121366
Received 20 March 2024; Received in revised form 10 July 2024; Accepted 15 August 2024
Available online 16 August 2024

1. Introduction

In recent years, due to the advantages of small size, long endurance, high concealment and simple operation, unmanned aerial vehicles (UAVs) can replace human beings to perform more complex or dangerous specific tasks. By adjusting the flight position and height, UAVs can detect and track moving targets at a distance, accurately and quickly capture target information from different angles, achieve coverage and monitoring of large areas within a short timeframe, it is very popular in military and civilian fields [1,2]. In the field of agriculture, Bhadra et al. achieved accurate and efficient estimation of corn biophysical properties using airborne hyperspectral images from drones [3]; In the field of electricity, Duo et al. utilized drones to achieve maintenance and repair of distribution lines [4]; In the field of geological survey, Liu et al. used drones to achieve aeromagnetic measurements in high-altitude mountainous areas [5]; In the field of urban inspection, Wan et al. used drones to achieve target tracking of pedestrians, vehicles, and other objects in smart cities [6]. The use of drones to achieve object recognition and detection in complex environments has greatly reduced the workload of human beings, but the traditional methods have slow running speed, low precision, computational redundancy, etc., which is difficult to meet the current needs of UAV development. Therefore, it is imperative to design an efficient and accurate UAV-based object detection method.

At present, the mainstream object detection algorithms are two-stage detection and one-stage detection algorithms based on deep learning. The two-stage detection algorithms first produce a range of sample candidate boxes, then employ a convolutional neural network to classify these samples, commonly used Region-based convolutional neural networks (R-CNN)[7], Spatial Pyramid Pooling Networks (SPP-Net)[8] and Fast R-CNN[9]. Liu et al. proposed a denoising feature pyramid network to improve R-CNN to achieve high-precision object detection [10]; Han et al. proposed a pure Sparse R-CNN to realize object detection in traffic scenes [11]. These algorithms have a high level of accuracy in detecting objects, but their main disadvantage is that they are slow in processing the data. However, the one-stage detection algorithms directly convert the problem of object bounding box positioning into a regression problem and thus has good reasoning speed. The most representative one is the 'you only look once' (YOLO) series of algorithms proposed in recent years. Chen et al. proposed a YOLO-v4 target detection method based on drone images to achieve rapid detection and statistics of armei trees in large orchards [12], Xie et al. used YOLO V5s to detect UAV thermal imaging images and realized rapid identification of animal targets with an accuracy of 94.1 % [13]. However, due to the characteristics of UAV aerial images, complex background information, small object scale, and sparse and uneven distribution, the detection model is too large, have low accuracy, and cannot achieve real-time detection in the existing work [14,15]. Hence, creating a UAV object detection model that maintains a balance between "speed and accuracy" as seen in YOLO holds significant theoretical and practical importance.

This paper proposes a YOLOv8-based lightweight UAV small object detection algorithm, LUD-YOLO, which not only overcomes the ubiquitous problems of aerial images such as occlusion, density, and light changes but also achieves lightweight object detection with stronger generalization. The following are the primary contributions:

(1) To address the issue of image feature degradation during propagation and interaction, and to enhance the detection accuracy of UAVs for small objects. A new multiscale feature fusion model is proposed. This model uses adaptive spatial fusion operations to filter the multistage fusion process, which can effectively avoid large semantic gaps between non-adjacent levels.
(2) To address the problem of large memory occupation and high computational cost when extracting samples, this paper proposes the C2f-BiLevel Routing Attention (C2f-BRA) feature extraction module, which introduces the sparse representation of features into Backbone and takes advantage of the self-attention mechanism to capture long-distance context semantic connections. Flexible computational allocation and content awareness are achieved through the utilization of dynamic sparse attention.
(3) To achieve model lightweighting, the proposed model selects and adjusts convolution channels that demonstrate low sensitivity. This can reduce the number of model parameters without changing the detection network, which not only helps the generalization of the model and improves the robustness, but also overcomes the problem of software and hardware adaptation in the application of UAV small object detection.

The rest of this paper has been thoughtfully organized as follows. Section 2 reviews the relevant literature on UAV object detection. Section 3 provides a detailed explanation of the proposed LUD-YOLO model, and the principles of the proposed feature fusion, feature extraction, and model weight reduction are introduced. In Section 4, a series of ablation and comparison experiments showed the superiority of the proposed LUD-YOLO. Finally, Section 5 summarizes and prospect.

2. Related works

Many traditional UAV object detection methods rely on digital signals such as audio, radar, and RF signal analysis. However, these methods generally have low accuracy in complex tasks involving many or overlapping targets and poor detection robustness in complex environments. Consequently, there's a growing trend among experts and scholars to utilize deep learning algorithms for the extraction of object detection features from UAV aerial images. Xu et al. introduced a Faster R-CNN-based UAV ground object detection method that uses deep learning and polarized hyperspectral imaging technology. The proposed feature map and object classification were obtained through the pooling operation of the area of interest [12]. Zhang et al. introduced a method for object detection in UAV images, employing a detailed approach that starts with broad detection and refines to specifics. This method utilizes lightweight convolutional neural networks (CNNs) to extract deep features for preliminary object detection in key frames. They also used LiteFlowNet and object prior knowledge to refine the detection results [13]. Darehnaei et al. proposed a Faster R-CNN-based deep transfer learning model for vehicle detection in UAV images, which adaptively optimized six basic weights and the final decision threshold

through a genetic algorithm to maximize the improvement in UAV detection [14]. Dai et al. suggested a hybrid model combining convolutional neural networks with transformers to enhance object detection in UAV images efficiently. They employed a cross-shaped window transformer as the core structure to capture image features across multiple scales. The hybrid patch embedding module was used to extract image edge and corner information, and finally, the inference results of the original image and the sliced image were fused to improve the detection accuracy of small UAV targets without modifying the original network [15]. Ren et al. proposed a method that integrated MobileNetV3 into Mask-RCNN to process UAV thermal infrared video to improve the processing speed of object detection while ensuring high detection accuracy [16]. Chen et al. proposed an improved Mask R-CNN algorithm for the detection of cracks on the exterior walls of buildings by UAVs. They used DenseNet to retain more information about cracks than residual network (ResNet), and by changing the number of connections of each dense block, they realized the efficient identification of cracks on building exterior walls by UAVs [17]. Du employed UAV hyperspectral imaging to identify subterranean natural gas leaks, integrating a crop growth model with a CNN. This innovative approach offers a novel method for nondestructively detecting minute natural gas leaks [18].

The one-stage lightweight high-precision UAV object detection method has been highly favored compared to the two-stage object detection methods applied to UAVs in some fields and achieved high detection accuracy. The YOLO series algorithms are representatives of one-stage object detection [19,20]. Researchers have tried to ensure accuracy while meeting the UAV object detection efficiency. Li et al. developed an enhanced version of YOLOv4 tailored for detecting small objects in UAV imagery. They implemented an ultralight quantum space attention mechanism to generate unique attention feature maps for each feature map subspace. Then, they introduced non-maximum suppression to reduce lost detection targets due to occlusion [21]. Luo et al. enhanced YOLOv5 with the introduction of three asymmetric convolutional feature extraction modules. They utilized the K-Means++ algorithm for more precise anchor box determination and adopted EIOU for advanced non-maximum suppression, significantly boosting the model's post-processing capabilities. Their methodology yielded exceptional outcomes across various UAV aerial imaging applications [22]. Concurrently, Zhao et al. refined the YOLOv5 framework for improved drone object detection, incorporating a converter encoder into the backbone network to better focus on significant areas. They leveraged both the global attention mechanism (GAM) and the coordinated attention mechanism (CA) to intensify feature interaction and better capture details of smaller objects [23]. Furthermore, Liu's team devised the DBF-YOLO model, specifically designed for UAV object detection, which includes a module for extracting shallow features. This addition enhances the preservation of semantic details for smaller objects. By integrating a feature fusion network that combines shallow feature maps with detection outputs into the FPN+PAN layer, they effectively reduced the rate of missed detections [24]. In addition, Huang et al. introduced BLUR-YOLO, a unique algorithm for UAV image object detection that employs the h-swish activation function in both the backbone and neck networks to augment the model's expressive capabilities. By implementing the CoordAttention mechanism, they effectively minimized background noise, and using BlurPool rather than traditional downsampling, they developed a feature pyramid network called Blur-PANet, which efficiently amalgamates multi-layer features [25]. Zhao et al. incorporated a prediction head based on YOLOv7 to detect miniature people or objects. By integrating a simple attention module that is parameter-free, they managed to achieve object detection in maritime UAV images by locating attention areas in the scene [26].

The improvement of two-stage detection algorithms mostly focuses on image feature extraction. Although these models have achieved high detection accuracy in complex scene detection, they have the disadvantage of slow detection speed after fusion. However, the improvement of one-stage detection algorithms mostly focuses on improving the effect of each stage of the framework, such as sampling effect, feature mapping, activation function, etc., ignoring the problem of the size of the entire detection model and running computing power. In general, many current research studies only focus on object detection accuracy and do not consider the limited computing power and small storage space of UAV embedded devices. Therefore, it is crucial to propose a UAV object detection algorithm that takes both processing speed and detection accuracy into account.

3. Proposed LUD-YOLO

3.1. Improvements in feature fusion

In the object detection task, a reasonable fusion mode of multiscale features is critical for objects with scale variance. We set the feature map sizes extracted by Backbone in YOLOv8 [16,17] to be C1-C5. The model first uses the feature pyramid network (FPN) structure to upsample the C5 features to the same size as C3 in a "bottom-up" manner. After that, the features are down-sampled to C5 size in a "top-down" manner to complete the Path Aggregation Network (PAN) process. In this process, the features of each level are fully generalized, and the features of FPN and PAN are fused to achieve concise and effective feature engineering. However, for the accurate detection of multiscale targets in UAV images, this method still has the following limitations. On the one hand, images from the perspective of UAVs, the targets are generally very small and difficult to identify. Therefore, an appropriate small target improvement strategy and feature reuse method are necessary. On the other hand, the High-Level at the top of the FPN is passed to the Low-Level at the bottom of the feature and has experienced the propagation of multiple intermediate scales. The features of each scale only interact with features of adjacent scales before feature fusion. Throughout this propagation and interaction phase, semantic information from high-level features might become diminished or compromised. In other words, a top-down approach in PAN may bring about the opposite problem; meaning, intricate details from low-level features might become diminished or compromised throughout the propagation and interaction phases. To tackle the issues mentioned earlier, this paper suggests implementing the following strategies for improvement:

Initially, to enhance the precision of object detection by UAVs, we integrated an upsampling technique within the Feature Pyramid

Network (FPN). Specifically, this modification involves upsampling features from C5 to C2 size in the FPN-PAN, followed by a subsequent upsampling back to C5 size. Throughout this process, the focus was on maximizing feature reuse and interaction between the Backbone and FPN-PAN. Consequently, feature maps of identical sizes are concatenated, boosting feature utilization efficiency without the need for additional features and mitigating feature degradation to some extent. Moreover, the architecture of the three detection heads remains constant throughout, avoiding the introduction of a detection mechanism for the C2 feature size. This approach helps in eliminating parameter redundancy and computational excess.

Secondly, the introduction of the Asymptotic Feature Pyramid Network (AFPN) effectively mitigates the issues related to feature degradation during feature propagation and interaction. AFPN employs adaptive spatial fusion techniques to selectively blend features throughout the multi-level fusion process. Its progressive architecture ensures that semantic information across various feature levels becomes more aligned as the fusion progresses. The process unfolds in two key phases: Initially, two low-level features of differing resolutions are merged, initiating the fusion sequence while preserving essential information from the larger feature map in the context of object detection. Subsequently, the system gradually integrates higher-level features into the mix, culminating in a comprehensive fusion of features across all sizes, enhancing detection capabilities. This fusion method can avoid large semantic gaps between nonadjacent levels, as shown in Fig. 1.

However, this process may lead to a slower inference speed of the edge device, negatively affecting the detection speed. Hence, to tackle the issue of personalization in UAV object detection, this paper introduces the AFPN idea to address the problem of information attenuation during C2-C4 size changes, ensuring feature information from small targets to larger targets, and achieving a better balance between feature engineering effects and detection speed.

Suppose x_{ij}^{n-l} represents the feature vector at position $(i.j)$ from Level n to Level l, and the resulting feature vector is expressed as y_{ij}^{l}. This vector is obtained by adaptive spatial fusion of multilevel features, and is represented by the feature vectors x_{ij}^{1-l}, x_{ij}^{2-l} and x_{ij}^{3-l}, the linear combination is as follows:

$$y_{ij}^{l} = \alpha_{ij}^{l} \cdot x_{ij}^{1-l} + \beta_{ij}^{l} \cdot x_{ij}^{2-l} + \gamma_{ij}^{l} \cdot x_{ij}^{3-l} \tag{1}$$

where α_{ij}^{l}, β_{ij}^{l} and γ_{ij}^{l} indicate the spatial weight of the three Level features at Level l, and are subject to the constraints of $\alpha_{ij}^{l} + \beta_{ij}^{l} + \gamma_{ij}^{l} = 1$. Considering the difference in the number of features fused at each stage of AFPN, an adaptive spatial fusion module of C2-C4 size features is implemented here. The entire feature fusion process we propose can be represented by Fig. 2.

3.2. Improvements in feature extraction

For image feature extraction in the process of small object detection, first, good features should have the following characteristics: A single sample is inherently sparse, that is, each sample only needs very few nonzero values to describe it; There is also sparsity among multiple samples, that is, the features represented by each row in the feature matrix have only a small number of non-zero values; The feature distribution of the images is uniform and consistent, that is, the statistical properties of each feature are similar. Second, in

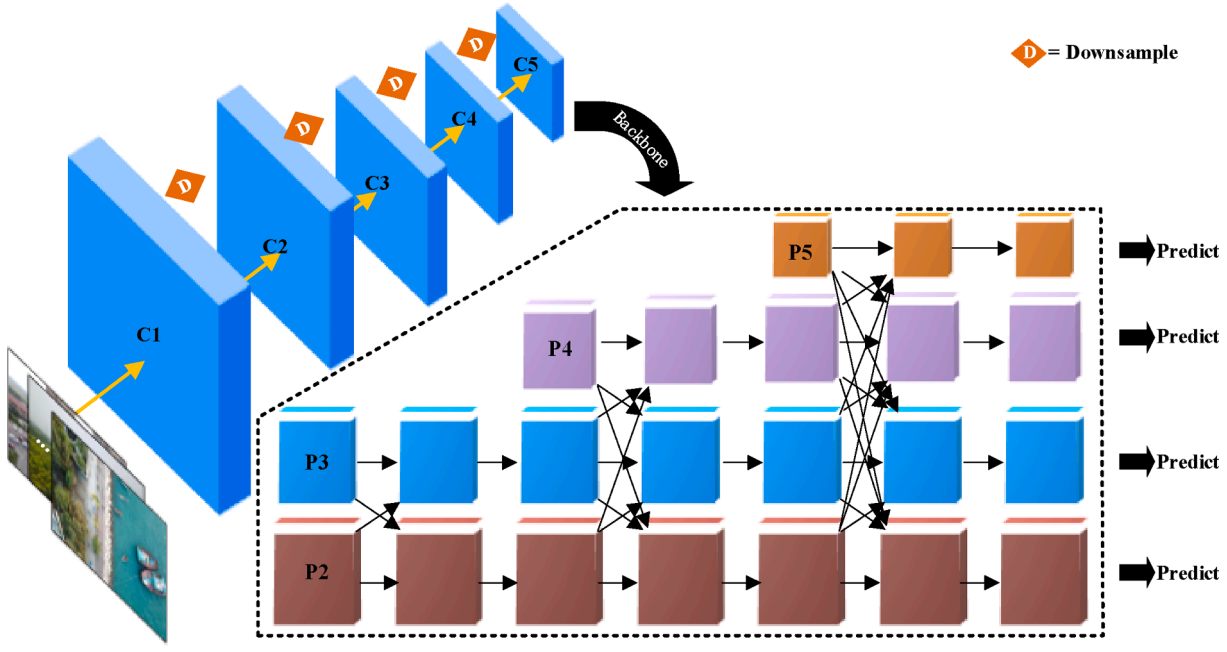


Fig. 1. Structure of the original AFPN.

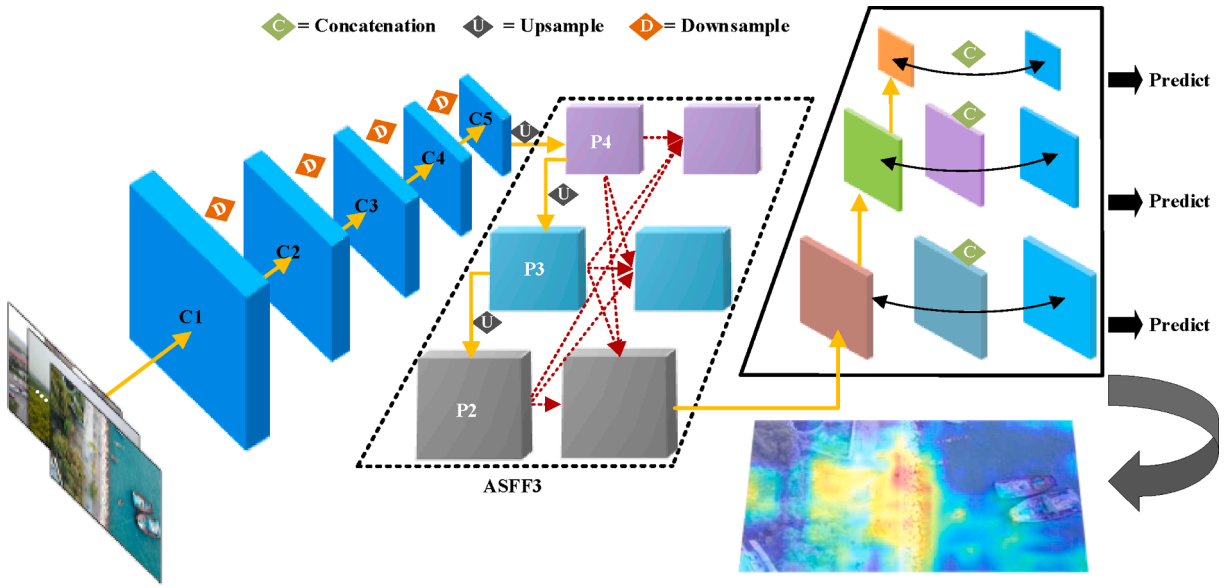


Fig. 2. The proposed structure for feature fusion.

YOLOv8, Backbone mainly uses a large number of C2f modules. Different from the C3 module in YOLOv5, the C2f module fully considers the rich gradient flow information and achieves better feature extraction ability. However, C2f is still composed of the convolutional neural network, which is a local operator in nature. In contrast, a key property of attention is the global receptive field, and the long-range dependency among features can be captured by using the attention mechanism.

Therefore, we introduce the sparse representation of features into Backbone and take advantage of the self-attention mechanism to capture long-distance contextual semantic connections, which improves the network's feature extraction capabilities. In this paper, we apply Biform module, an attention mechanism with sparse relationships, in the C2f module. While taking advantage of the above advantages, it reduces the problems of large memory usage and high computational cost of the attention mechanism. Overall, it can also reduce the size of the original YOLOv8 and improve the inference speed.

The core mechanism of the Biform module lies in the dynamic sparse attention structure, which implements flexible computation allocation and content awareness. The workflow is as follows:

First, $S \times S$ non-overlapping regions are divided by feature map $X \in R^{C,H,W}$, such that each region contains $H \times W/S^2$ feature vectors, that is, X is converted to $X^r \in R^{S^2 \times HW/S^2 \times C}$;

Second, set Query, Key, Value, that is, $Q, K, V \in R^{S^2 \times HW/S^2 \times C}$, with a linear projection:

$$Q = X^r W^q, \mathrm{K} = X^r W^k, V = X^r W^v \tag{2}$$

Third, a directed graph is constructed to represent the participation relationship. For each given region, the mean value of each region is applied to derive the region-level Q^r and K^r, and the matrix multiplication between them derives the adjacency matrix A^r of region-to-region semantic relations.

$$A^r = Q^r (K^r)^T \tag{3}$$

Prune the association graph by retaining only the $Top-k$ connections for each region.

$$I^r = TopkIndex(A^r) \tag{4}$$

Finally, based on the area-to-area routing index matrix I^r, applying fine-grained token-to-token attention:

$$K^g = gather(K, I^r), \quad V^g = gather(V, I^r) \tag{5}$$

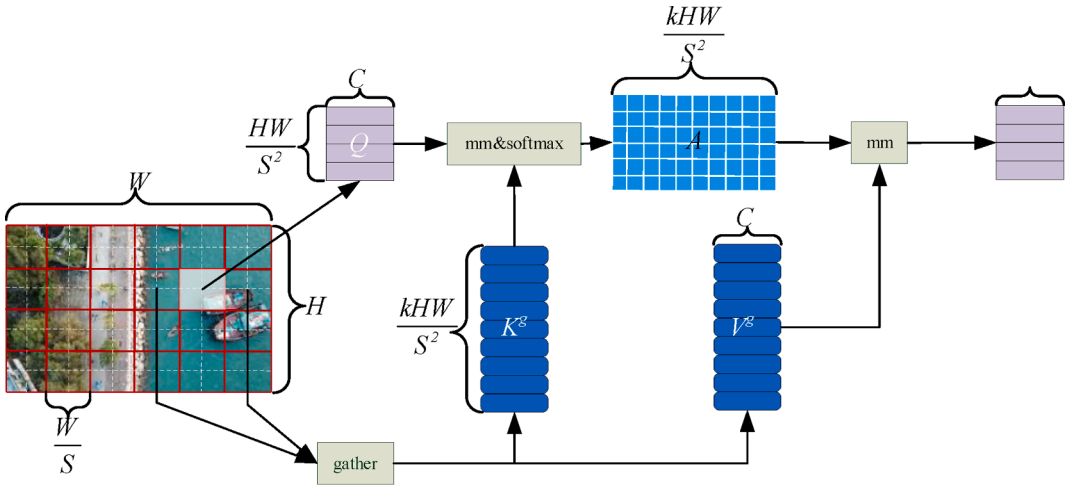
$$O = Attention(Q, K^g, V^g) + LCE(V) \tag{6}$$

where $LVE(\cdot)$ represents the use of deep convolution for parameterization, $Q, K, V \in R^{S^2 \times HW/S^2 \times C}$, $W^q, W^k, W^v \in R^{C \times C}$, $Q^r, K^r \in R^{S^2 \times C}$, $A^r \in R^{S^2 \times S^2}$, and $K^g, V^g \in R^{S^2 \times HW/S^2 \times C}$.

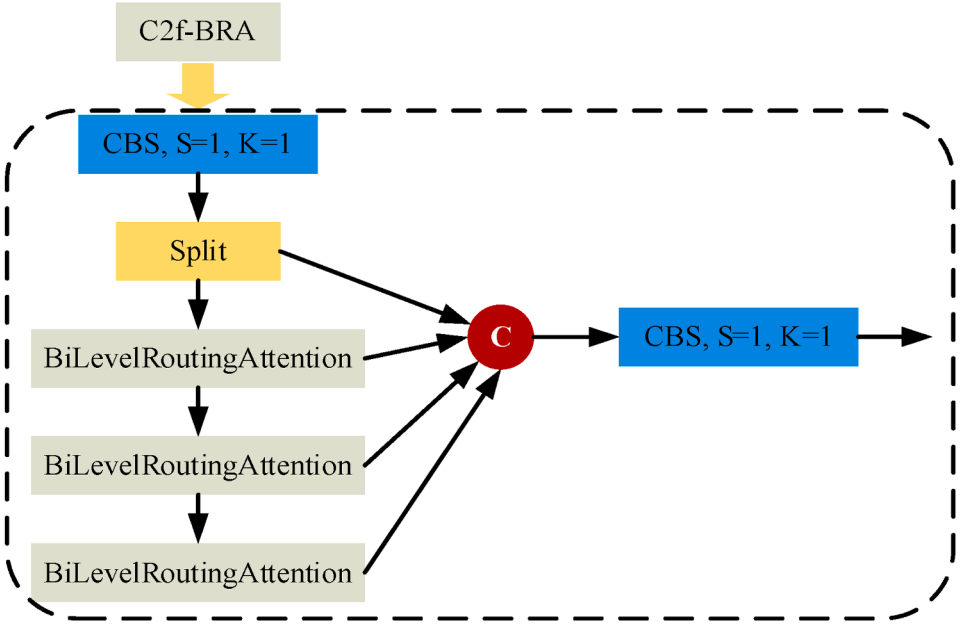
The above ideas are applied to the C2f module to form the C2f-BRA module, as shown in Fig. 3.

3.3. Lightweight adjustment of the LUD-YOLO

When the object detection model is deployed to a UAV, the embedded device inference model needs to be used. In general, the



(a)



(b)

Fig. 3. Improvements in feature extraction methods. (a) Schematic of the Biform module; (b) The structure of C2f-BRA module.

factors that influence the actual deployment of the model include the size of the model, the memory occupied during runtime, and the amount of calculation. When inferring the object detection model, the activation response process of the convolutional neural network will occupy a large amount of running memory, which is a great resource burden for embedded devices. In this paper, we used Network Slimming [18] to sparse the trained model, and use a simple and effective network retraining solution to implement UAV edge device deployment under limited resources.

The essence of this approach involves applying L1 regularization constraints to the scaling factor within the batch normalization (BN) layer of the convolution block in the original network. This moves the value of the BN scaling factor closer to zero. This process does not macroscopically change the structure of the detection network, but only adds additional regularization items, which basically does not affect the overall performance of the model. Specifically, before pruning the model, the L1 regularization constraints are applied to the BN layer in the model and training begins. After the training is complete, the pruning ratio is set to select the reserved convolution channel by arranging the weight in descending order and using the pruning ratio. Since each weight corresponds to a

specific convolutional channel, less sensitive convolutional channels in the network will be pruned. In fact, in some cases, it helps the model generalize and improves its robustness. In addition, pruning unimportant channels may temporarily reduce model performance. Therefore, after pruning the model, the network retraining (fine-tuning) must be used to compensate for the loss of accuracy and to regain the same reasoning ability as before. Due to the feature fusion process at the C2-C5 layers of the backbone and neck in the proposed network, to reduce the number of parameters while ensuring the quality of feature engineering, the network involved in feature fusion and the C2f-BRA module do not participate in sensitivity analysis. The entire lightweight adjustment process includes four steps: adding regularization constraints, acquiring sensitivity, pruning convolution channels that do not reach the threshold, and fine-tuning, which can be expressed by the pseudocode given in Table 1.

After the completion of the Network Slimming process, compared with the initial detection model, the obtained new network is more compact and better in terms of model size, runtime memory, and computation time. The above lightweight process can usually be repeated multiple times to obtain a multipass network slimming scheme to make the network more concise. In this paper, the form of proportional pruning is chosen to implement the above process, and the scale factor is 0.8. Fig. 4 shows the relationship between convolutional layers and scaling factors.

Thus far, the schematic structure of the entire LUD-YOLO model can be obtained, as shown in Fig. 5. The figure shows the usage of the proposed improved feature fusion, C2f-BRA module, lightweight processing after model training, and finally adapted to the UAV to complete object detection. Similar to the N/S/M/L/X 5 variants of YOLOv8, this paper applies the improved strategies to YOLOv8-n and YOLOv8-s respectively, named LUDY-N and LUDY-S, to consider the lightweight model suitable for UAVs. The benchmark model YOLOv8-n is the minimum detection model specially suitable for edge devices, and the benchmark model YOLOv8-s is the model with the best detection effect on limited computing resources.

4. Experiment results and analysis

4.1. Datasets

The VisDrone2019 dataset [19] is a large-scale dataset for UAV vision research and algorithm evaluation, which is widely used internationally. This paper selects this dataset as the experimental object for UAVs target detection. This dataset contains more than 6,000 video sequences and more than 25,000 images. The dataset covers various scenarios, such as urban, rural, highway, and construction sites, as well as under different environments, weather conditions, and target scales, which enable the algorithm to adapt and perform better when faced with different challenges. These data define a variety of common target categories, including pedestrians, vehicles, bicycles, and motorcycles, etc., which is of great significance for tasks such as traffic monitoring, pedestrian identification, and vehicle tracking. In terms of information labeling, the VisDrone2019 dataset provides detailed labeling information, including object bounding boxes, object categories, motion state, and occlusion, etc. Detailed annotation information is particularly important for algorithm training, evaluation, and validation.

The release of this dataset aims to promote the development and performance evaluation of UAV vision algorithms. The proposed

Table 1
Pseudocodes of the lightweight process.

Pseudo-code of our network slimming	
01	Add L1 regularization constraint {
02	initialize original model and prune ratio
03	for module in original model
04	if module includes BN
05	detach module into w, b
06	Add L1 regularization $z_1 = \frac{z_{in} - \mu_\beta}{\sqrt{\sigma_B^2 + \xi}}, \quad z_{out} = \gamma z_1 + \beta$
07	}
08	Obtain Sensitivity
09	Calculate the importance score of each parameter
10	Sort the parameters in descending order of importance score
11	Prune according to pruning ratio, determine the number of parameters that need pruning
12	Prune according to pruning ratio_Ratio, determine the number of parameters that need pruning num_Pruned_Params
13	Trim convolutional channels
14	Select the top num from the sorted parameter list_Pruned_Params parameters for pruning by following rules:
15	for each layer
16	if it isinstance C2f (m1)
17	trim convolutional channels as TopConv (conv2d + BN+action)
18	else if it not isinstance list (m2)
19	for I in m2
20	If I is instance C2f, SPPF or C2f-BRA
21	trim convolutional channels as one module (conv2d)
21	Fine-Tuning
22	Reinitialize the remaining parameters to keep the model available
23	Return the pruned model
24	Retrain the model to restore accuracy

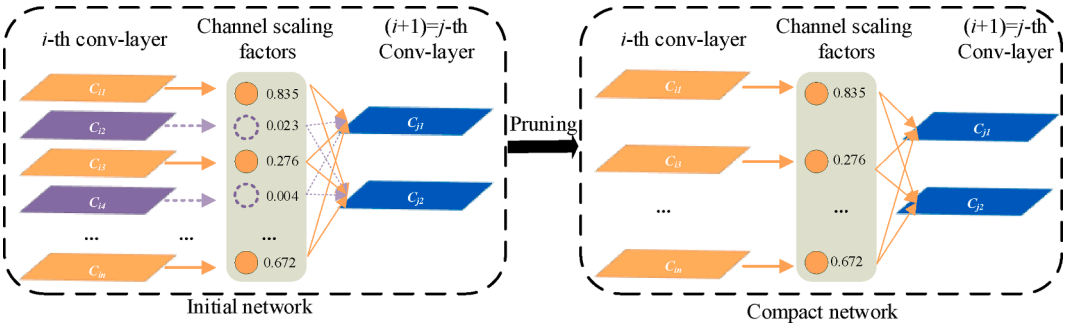


Fig. 4. The relationship between the convolutional layers and the scale factors.

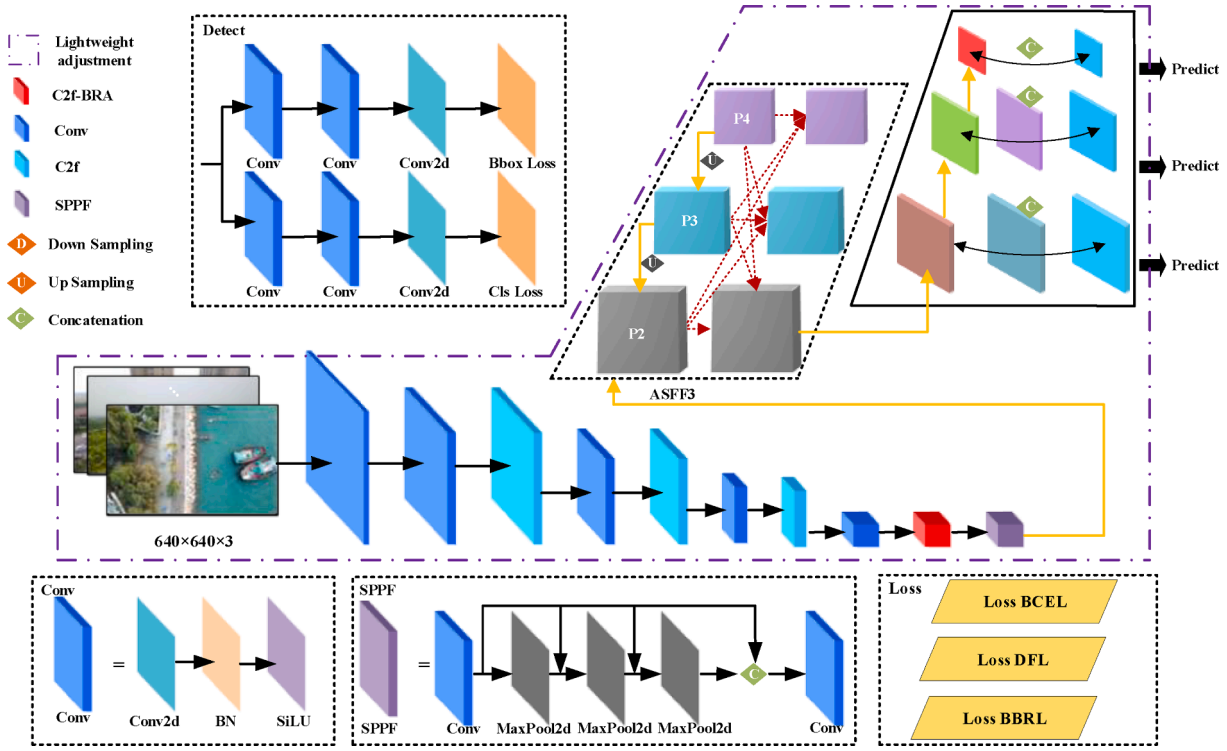
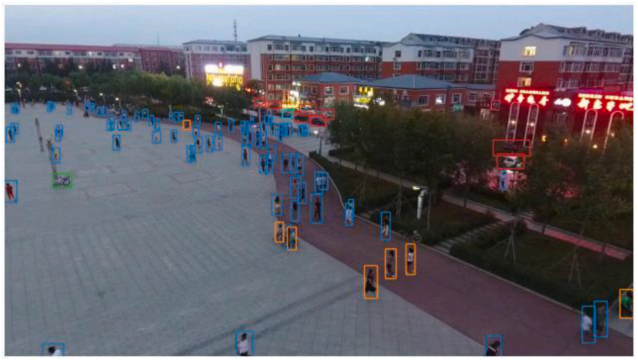


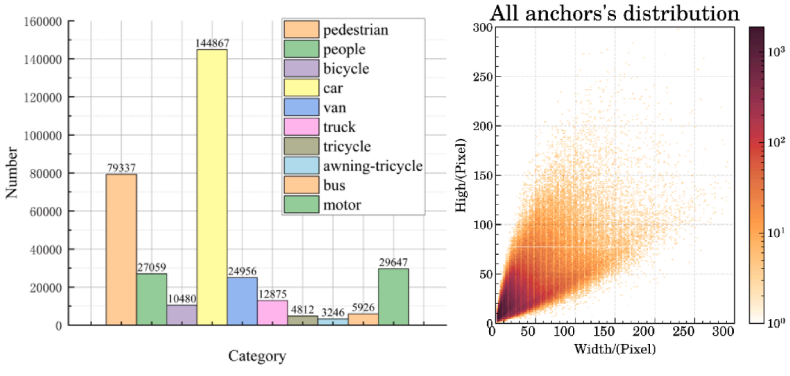
Fig. 5. The structure of the proposed LUD-YOLO.

LUD-YOLO will be verified in VisDrone2019 with extensive and rich samples to facilitate better evaluation of model performance. Fig. 6 shows the specific distribution of the example images, the number of labels, and the size of the label in the dataset. It can be seen that this dataset has the following difficulties and characteristics. First, the detection objects obviously have the salient characteristics of a large number, blurriness, small size, and ease of confusion. It is difficult to distinguish the pedestrian class from the people class in the example scenario. From the perspective of UAV shooting, the size of objects varies over a large range, and mutual occlusion is more obvious, which is a big challenge to the performance of the object detection algorithm. Second, there is a large difference in the number of labels for each category. The number of anchors for the Car class in the dataset reached 144867, while the number of anchors for the Awning-tricycle class only reached 3246. The unbalanced performance of samples between classes will put higher requirements on the robustness of the detection model. Third, consider the size distribution of anchors overall or for each category, except for the large number of medium and large-sized anchors in the Van, Truck, Car and Bus categories, most of the anchors are within the 100*100 pixel value, especially the number of small targets within the 50*50 pixel value is the most concentrated, so the small object detection ability of the model is critical in UAV object detection.

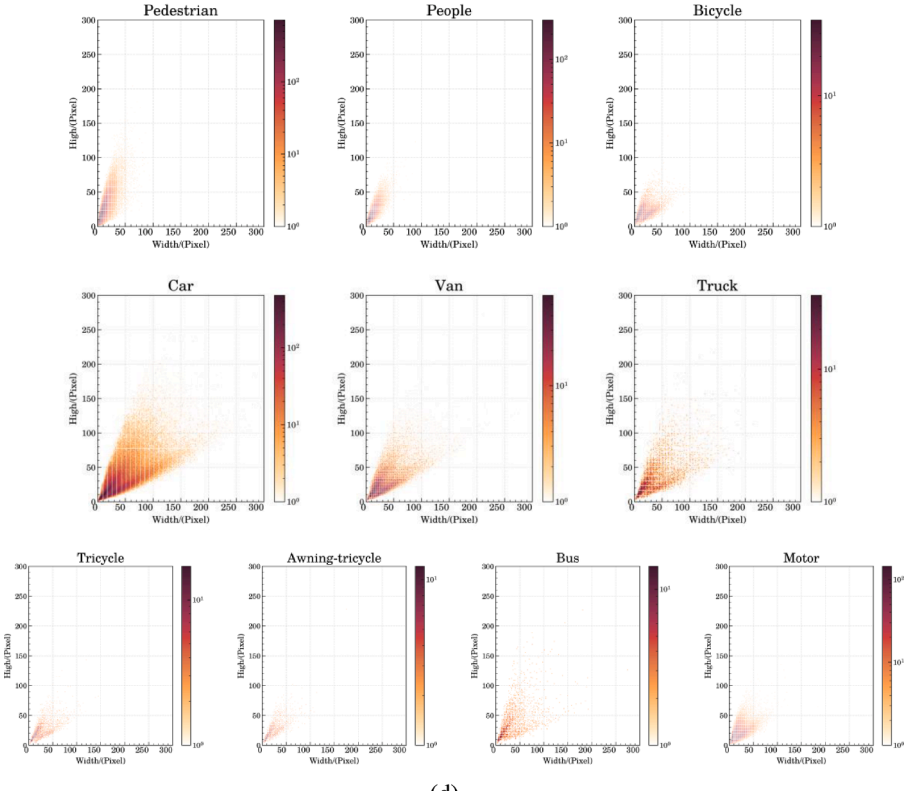
In this paper, similar to the dataset division of the VisDrone2019 Challenge, we divided the entire dataset into training, validation, and testing sets, each containing 6471 samples, 548 samples, and 1610 samples. Since the original design of the model is to be deployed in UAVs with limited space and computing power through embedded devices, this paper only investigates the improvements of YOLOv8-n and YOLOv8-s, and the sample images for training and testing are both set to the size of 640*640. Specifically, on the



(a)



(b) (c)



(d)

(caption on next page)

Fig. 6. Analysis of the labels for the VisDrone2019 dataset. (a) Label visualization example; (b) Distribution of the number of tags;(c) Overall distribution of anchor sizes; (d) Distribution of anchor sizes by category.

basis of YOLOv8, two groups of parameters [0.33, 0.25, 1024] and [0.33, 0.50, 1024] will be used as the scaling factors for the network depth, width and maximum channel in the model. Meanwhile, we use four conventional data enhancement methods, including scaling, flipping, etc., to suppress the imbalance problem caused by the small number of anchor types, and enhance the robustness of the algorithm. To fairly compare and accelerate model training, the ablation experiment training process of LUD-YOLO was performed at the Supercomputing Center of the State Key Laboratory of Public Big Data, Guizhou University. The graphics processing unit was an NVIDIA TESLA A100 with 40 GB video memory, and the central processing unit used Intel(R) Xeon(R) Silver 4314 with a main frequency of 2.40 GHz, a running memory of 32 GB, and an operating system of CentOS 7.9. All models were trained using the environments of CUDA 11.6, Python 3.9.7 and PyTorch 1.9.1. The hardware used for training and testing are listed in Table 2. In addition, the test was performed on the local Windows 10 operating system, using the Intel(R) Core (TM) i9-12900 K@3.19 GHz CPU and the GeForce RTX 3090 GPU, with running memory and video memory of 24 GB and 32 GB, respectively. Other environments are the same as the training environments. Table 3 lists the important parameters used during the training process.

4.2. Ablation experiment

This paper introduces the idea of AFPN and Biform on the basis of the original YOLOv8, and performs pruning operations on the generated network to minimize the number and complexity of network parameters while ensuring accuracy. We use YOLOv8-n and YOLOv8-s as the benchmark models to evaluate the impact of each proposed units on the model in turn. The ablation experiments used precision rate (P), re-call rate (R), mean value of average precision (mAP), frames per second (FPS), number of parameters, and model size that are widely used in the field of object detection as evaluation indicators [20–22], which can be displayed the performance changes of each model in multiple dimensions. The ablation experimental results of proposed LUD-YOLO are presented in Table 4, Table 5 and Fig. 7, the optimal results in the tables are presented in bold.

Table 4 and Table 5 respectively present the experimental results of the two lightweight benchmark models after adding the AFPN module, Biform module, AFPN and Biform module are added together with the pruning method. It can be seen that:

First, as the proposed modules are integrated into the two types of benchmark models one by one, the accuracy indicators P, R, and mAP show an overall upward trend in the verification set and test set. However, in some cases, the P value of the benchmark model + AFPN was the best. That is, AFPN can ensure the quality of multi-dimensional fusion from small feature maps to large feature maps, greatly improving feature engineering, which ensures that the accuracy and recall rate during object detection are always maintained at excellent levels. The proposal of the C2f-BRA module fully applies the sparsity idea of the Biform algorithm, so that the complexity of the entire model can be reduced while ensuring the detection effect. The two modules promote each other, which enhances both the accuracy and the speed of the model's detection capabilities.

Second, after pruning to remove redundant channels of block convolution, such as conventional convolution, C2f-BRA and other module convolutions, the accuracy indicator of the model decreased slightly, and the final mAP indicator only decreased by 0.1 %-0.3 %. However, the number of model parameters decreased by 7.35 %-7.76 %.

This directly makes the parameters and size of the model better than the benchmark model, and the mAP is 7.77 %-11.75 % better than the benchmark model respectively. The FPS indicator performance is optimal in all scenarios. This suggests that the strategies outlined in this paper for the original model are advantageous when taking into account the accuracy and speed requirements of edge device detection scenarios.

In addition, Fig. 7 visually presents the performance of each indicators of the model, as well as the comprehensive performance of the entire model. The A, B, C, D, and E correspond to the five models in the table.

It can be clearly noticed from the Fig. 7 on the left that the improved model combining AFPN and Biform has the most outstanding accuracy performance. However, in terms of FPS, Parameters and Model size, it is only better than the baseline model with the AFPN module added, and the model's detection speed and model universality are average. The two lightweight models proposed in this paper have achieved the most balanced performance results. Especially in the radar chart on the right, the model E shows the best

Table 2
Configuration of training and testing experiment environments.

Environment	Parameter
CPU	Intel(R) Xeon(R) Silver 4314 @ 2.40 GHz (training)Intel(R) Core (TM) i9-12900 K@3.19 GHz (testing)
GPU	NVIDIA TESLA A100 (training) GeForce RTX 3090 (testing)
VRAM	40 GB (training) 24 GB (testing)
RAM	32 GB
Operating System	CentOS 7.9
Language	Python 3.9.7
Frame	Pytorch 1.9.1
CUDA Version	11.6

Table 3
Training parameters setting.

Epochs	Batch Size	Initial Learning Rate	Final Learning Rate	Optimizer	Momentum	NMS IoU	Weight-Decay
150	8	1.00E-02	1.00E-04	SGD	0.937	0.7	5*1e-4

Table 4
Results of the LUDY-N ablation experiment.

Data	AFPN	Biform	Pruning	P	R	mAP	FPS	Parameters/million	Model size/MB
Val	×	×	×	0.412	0.325	0.315	205	3.008	6.082
	√	×	×	0.458	0.344	0.348	171	3.491	7.090
	×	√	×	0.427	0.322	0.319	218	2.552	5.183
	√	√	×	0.471	0.349	0.353	180	3.035	6.191
	√	√	√	0.470	0.347	0.352	218	2.812	5.560
Test	×	×	×	0.365	0.286	0.257	218	3.008	6.082
	√	×	×	0.396	0.302	0.281	197	3.491	7.090
	×	√	×	0.382	0.285	0.256	257	2.552	5.183
	√	√	×	0.401	0.305	0.281	199	3.035	6.191
	√	√	√	0.401	0.303	0.279	263	2.812	5.560

Table 5
Results of the LUDY-S ablation experiment.

Data	AFPN	Biform	Pruning	P	R	mAP	FPS	Parameters/million	Model size/MB
Val	×	×	×	0.504	0.373	0.381	186	11.129	22.203
	√	×	×	**0.535**	0.411	0.422	173	13.048	25.789
	×	√	×	0.507	0.373	0.384	179	9.301	18.389
	√	√	×	0.531	**0.413**	**0.424**	164	11.209	22.204
	√	√	√	0.525	0.408	0.417	**194**	10.339	20.492
Test	×	×	×	0.430	0.331	0.309	209	11.129	22.203
	√	×	×	**0.471**	0.352	0.339	193	13.048	25.789
	×	√	×	0.447	0.326	0.311	209	9.301	18.389
	√	√	×	0.468	**0.347**	**0.336**	197	11.209	22.204
	√	√	√	0.464	0.343	0.333	**213**	10.339	20.492

performance.

The detection results of the model on some test sets are shown in Fig. 8. The experiments selected multiple types of detection scenarios such as complex backgrounds occlusions dark light, dense, small targets, overhead shooting angles and other detection scenes as detection samples, which places high requirements on the robustness of the detection model. It can be seen that the proposed model has completed the detection task well, and can accurately identify and determine the target position in various scenarios. However, the following problems also exist. On the one hand, the proposed model's accuracy in distinguishing Pedestrain and People in dense scenes needs to be improved. Especially in dense crowd scenes, people have different postures, and the detection effects of the above two categories are prone to confusion; On the other hand, there are still missed detections for small-sized Motor and Tricycle classes. Under the premise that the imaging effect is fixed and the model is lightweight, continuously improving the detection effect of small targets will be of great benefit to UAV object detection.

4.3. Comparisons with other object detection networks

This paper focuses on aerial image detection methods that are more in line with actual engineering scenarios, have low requirements on hardware conditions and have good universality. Therefore, the selection range of the comparison model is only the one-stage object detection algorithm. This type of model is easy to deploy and has been widely used in various fields of production and life. Based on mechanistic differences, this paper selected the more advanced TOOD in the current field [23]. The VFNet [24] method, various YOLO models, and the classic improved model based on YOLO were used as comparison objects. Specifically, the latter contains YOLOv3-Tiny [24,25], YOLOv4-S [26], YOLOv5-Lite-G [25], YOLOv5-S [27], YOLOX-Tiny [25,28], YOLOX-S [28], PP-PicoDet-L [25,29] and YOLOv7-tiny [30]. The abovementioned YOLO series models cover a wide range, and a large amount of literature has demonstrated their reliability and validity in different scenarios. Table 6 summarizes the results of the comparison experiment. Except for [23] and [24], which are official data from Visdrone2021, all other algorithms used Pytorch as the training framework and a Random Gradient Descent (SGD) optimizer during the training process. The initial learning rate is set to 0.01, and the final learning rate is set to 12 % of the initial learning rate. In addition, all input images were resized to 640 x 640, and all networks in the experiment were trained from scratch without using official pre training weights. Due to the limitations of the experimental environment, experimental hardware selection, the comparison dimensions of multiple models (such as FPS and other indicators)

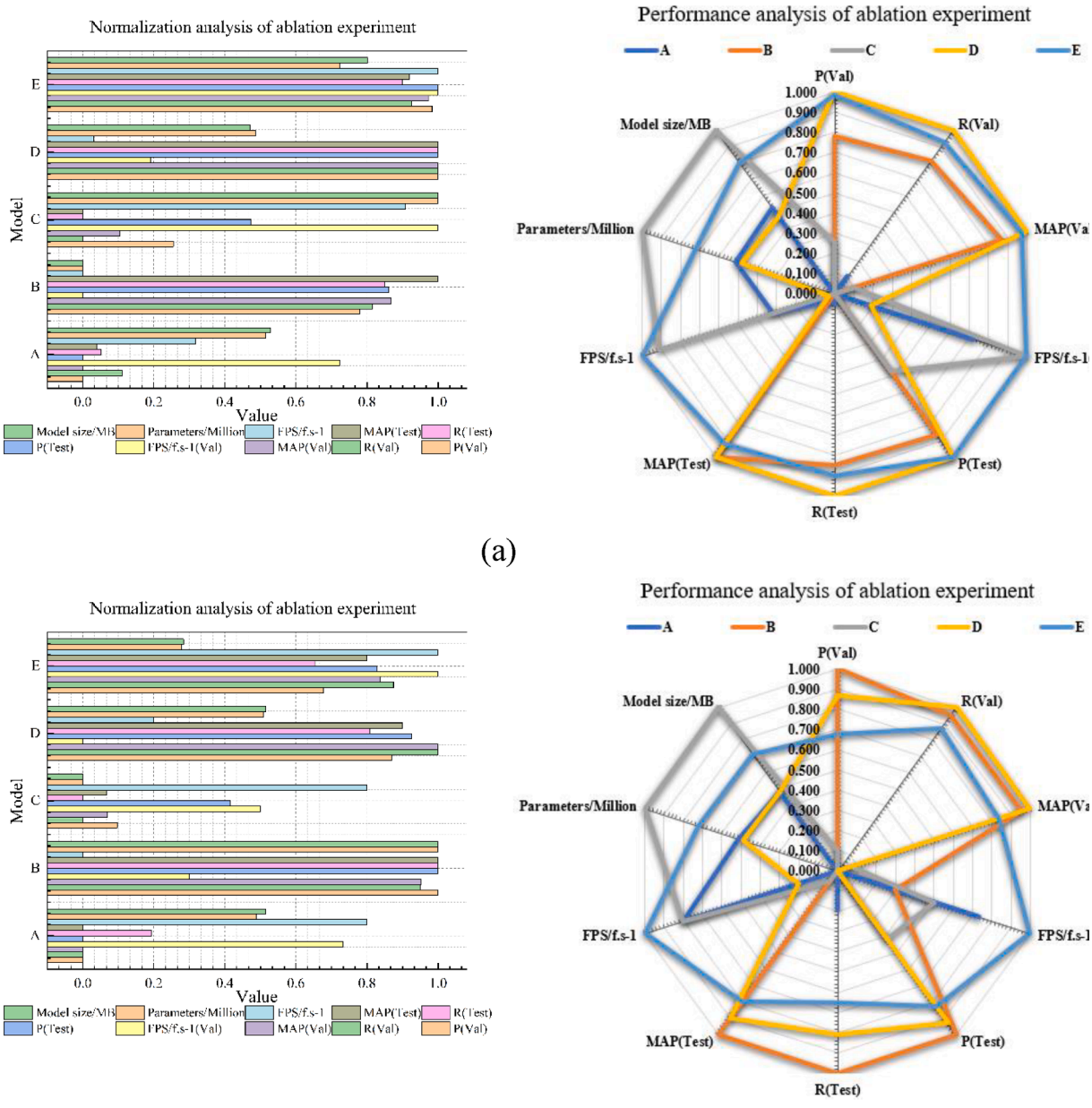


Fig. 7. The effect of normalization of overall indicators. (a) The ablation experiment of LUDY-N; (b) The ablation experiment of LUDY-S.

could not be aligned. Therefore, in this paper, only the mAP (Val) data of Params and various models were compared. The latter was accurate to 10 categories in the dataset. It should be noted that to ensure fairness, the model in this paper did not use the pretrained weights for training. Meanwhile, generally speaking, the above indicators can reflect the differences in accuracy and complexity among the models.

Table 6 lists the experimental results of the proposed LUD Yolo-N, LUD Yolo-S, and other 10 compared algorithms. Compared to the one-stage object detection model for TOOD and VFNet, the LUDY-S achieved the best results. When the Params of the model was only 1/3 of the one-stage object detection model, the overall mAP was 41.7. Even in the Awning Tricycle with the smallest amount of data, mAP was higher than these two algorithms by 2.8 and 1.4, respectively. Meanwhile, compared with the same type of lightweight models YOLOv4-S, YOLOv5-S and YOLOX-S, the proposed LUDY-S model params were only higher than those of 1.21, 1.22 and 1.4 respectively, but for most targets in VisDrone-2021, LUDY-S achieved high detection accuracy, which shows that the proposed strategies can greatly improve the comprehensive performance of the original model.

Compared with the typical lightweight models YOLOv3-Tiny, YOLOX-Tiny and YOLOv7-Tiny, the LUDY-N Params proposed in this

Fig. 8. The detection results of the proposed LUD-YOLO in some scenes.

Table 6
Comparison of Experimental Results.

Network	Params(million)	mAP50(%)										
		All	People	Pedestrian	Tricycle	Van	Truck	Awning Tricycle	Car	Bus	Bicycle	Motor
TOOD	31.81	41.0	31.9	41.5	31.8	46.5	39.6	14.1	81.4	53.5	19.2	50.5
VFNet	33.50	41.3	25.4	41.8	35.1	47.4	41.7	15.5	80.4	57.0	20.0	48.8
YOLOv3-Tiny	8.68	15.9	18.4	19.4	8.2	12.7	9.7	4.0	49.9	14.6	3.2	18.9
YOLOv4-S	9.13	31.8	32.9	42.4	17.3	33.2	24.8	9.9	74.2	36.2	8.4	38.4
YOLOv5-Lite-G	5.39	27.3	26.6	34.6	13.8	28.4	24.0	6.7	69.3	28.3	7.7	33.4
YOLOv5-S	9.12	38.5	31.8	41.9	27.0	44.7	34.8	16.3	79.3	55.0	11.5	43.0
YOLOX-Tiny	5.04	31.3	21.9	35.8	18.1	34.7	28.1	10.2	73.3	46.3	9.6	34.9
YOLOX-S	8.94	32.5	13.6	41.8	18.0	40.5	39.0	12.4	76.5	51.2	7.2	25.2
PP-PicoDet-L	3.30	34.2	35.3	40.2	21.1	35.4	29.3	12.1	75.6	44.3	12.8	36.3
YOLOv7-Tiny	6.03	31.3	33.9	37.6	16.4	34.9	21.9	7.8	75.2	40.1	4.72	40.4
LUDY-N	**2.81**	35.2	29.3	36.9	22.2	41.8	31.4	13.6	77.4	49.8	9.97	39.4
LUDY-S	10.34	**41.7**	34.3	44.8	29.8	48.4	39.4	16.9	80.9	62.2	14.5	46.2

article is much smaller than the above models, only 2.81 MB, but the overall average accuracy mAP is 35.2, which is better than the comparison models YOLOv3-Tiny (+19.3) and YOLOX-Tiny (+3.9) respectively. Although the overall mAP is not dominant compared to YOLOv7-Tiny, the proposed LUDY-N all performed better for both the Car with the highest number of anchors and the Awning cycle with the lowest number, indicating that the model has higher robustness.

Similarly, compared with other advanced lightweight models YOLOv5-Lite-G and PP-PicoDet-L, the proposed LUDY-N achieves better results in Params and mAP, which is better than YOLOv5- Lite-G (−2.58, +7.9) and PP-PicoDet-L (−0.49, +1), respectively. In summary, the proposed LUD-Yolo has high detection accuracy, strong generalization and robustness in solving object detection tasks of UAV aerial images. At the same time, the LUD-Yolo model is easier to deploy in UAV embedded devices for future practical applications.

To more intuitively show the evaluation results of the above 12 models for different scenarios, this paper uses a histogram of index normalization to describe the comparative experiment, but we subtracted the normalized result of the parameter quantity from one. Therefore, all results are better as they are closer to one, as shown in Fig. 9. Except for the one-stage object detection algorithms and PP-PicoDet-L, the mAP of the proposed LUDY-N is far better than the other seven lightweight detection models. When considering the size of the lightweight models, the Params of the proposed LUDY-N are much smaller than other comparison models, and it is the lightest target detection model. Therefore, the proposed LUDY-N has the best comprehensive performance.

In addition, Fig. 9 shows that compared with the one-stage algorithms, the proposed LUDY-S also has better performance. In terms of Params and detection accuracy, the LUDY-S achieved the best results, further indicating that the proposed work can minimize the complexity of the model while ensuring accuracy. Compared with the same type of YOLO series lightweight models, the proposed LUDY-S also shows satisfactory results in various object detection. Therefore, the proposed improvement strategies have competitive overall performance when applied to the original YOLOv8-S and YOLOv8-N for small object detection of UAVs.

4.4. Comparative experiments on other datasets

To further verify the robustness and practicality of the proposed method, the proposed LUD-YOLO will be applied to Unmanned Aerial Vehicle Benchmark Object Detection and Tracking (UAVDT) [31] for experiments. The UAVDT dataset is a large-scale and challenging benchmark dataset specially designed for UAV object detection and tracking tasks. The dataset contains 10 h of raw video data with approximately 80,000 video frames. These video data are mainly from drone cameras, covering a variety of practical application scenarios, such as traffic monitoring, security monitoring and so on. The UAVDT includes 14 attributes such as multiple scenes, weather conditions, flight altitude, and vehicle class, and the dataset is highly diverse and challenging due to factors such as drone movement, light changes, and occlusion. In order to demonstrate the generalization capability of LUD-YOLO, this paper will continue to explore the performance of LUD-YOLO with various one-stage models in the UAVDT-M dataset, Different from the previous ones, the comparison objects are mainly the improved models and benchmark models of YOLO series: YOLOv8-s, YOLOv8-n, YOLOv7tiny-silu [32], YOLOv7tiny-mobilevit-odconv [33], YOLOv5s-Repvgg [34], YOLOv5n-Bifpn [35], YOLOXs [36], MobileNetv3-SSD [37]. In the process of model training, the UAVDT-M data set was randomly divided into 6:4 as the training set and validation set, respectively, and the training parameters and evaluation indexes were consistent with those mentioned above. The results are shown in Table 7.

As with the previous experimental results, the proposed LUDY-N achieved the optimal mAP of 0.809. Except for the benchmark model YOLOv8-s, LUDY-N achieved the best results for all comparison algorithms. Even compared to multiple large models, LUDY-N has achieved better accuracy. But compared to the same type of LUDY-S, the mAP of LUDY-S is 0.031 higher than that of the benchmark model YOLOv8-s. Therefore, it is proved that the object detection accuracy of the proposed LUD-YOLO is still high even under different types of datasets, indicating that the proposed model has strong generalization.

In addition, by comparing the sizes of all detection models, it can be seen that, while ensuring higher detection accuracy, the proposed LUDY-N and LUDY-S are both smaller than the benchmark models YOLOv8n and YOLOv8-s. Although LUDY-N is 0.11 MB (+2.11 %) larger than the smallest YOLOv5n-Bifpn model, mAP is 0.041 (+5.34 %) higher than it, and LUDY-S is smaller than the same type YOLOXs model with better detection accuracy.

Similarly, a bar chart with normalized indicators was used to describe the comparison experiment, as shown in Fig. 10. As can be seen from the figure, the comprehensive performance of the models proposed in this paper has achieved a relatively clear lead. The P, R and mAP indexes of LUDY-S are superior to the other 8 types of lightweight detection models, and also have advantages in FPS indexes. LUDY-N has the best performance in FPS index, and P, R, and mAP index are only inferior to the benchmark model YOLOv8-s, which is very suitable for deployment in all kinds of edge devices.

It can be seen that our proposed model also achieved outstanding performance in UAVDT dataset, both in detection accuracy and detection speed are better than the comparison model of the same magnitude, which proves the effectiveness of relevant improvement strategies, and the LUD-YOLO has good generalization ability and strong universality.

In summary, the results of the ablation experiments and comparison experiments above show that each proposed strategy has a positive enhancement effect on YOLOv8. Both LUDY-N and LUDY-S can be deployed on UAVS to achieve various object detection tasks. When the computing power of edge computing equipment is limited, a smaller LUDY-N model can be selected. When it is necessary to complete the UAV target detection task with higher precision, LUDY-S model with more balanced performance can be selected.

5. Conclusions and future work

This paper proposes an efficient, lightweight, and practical object detection framework based on YOLOv8 to address the

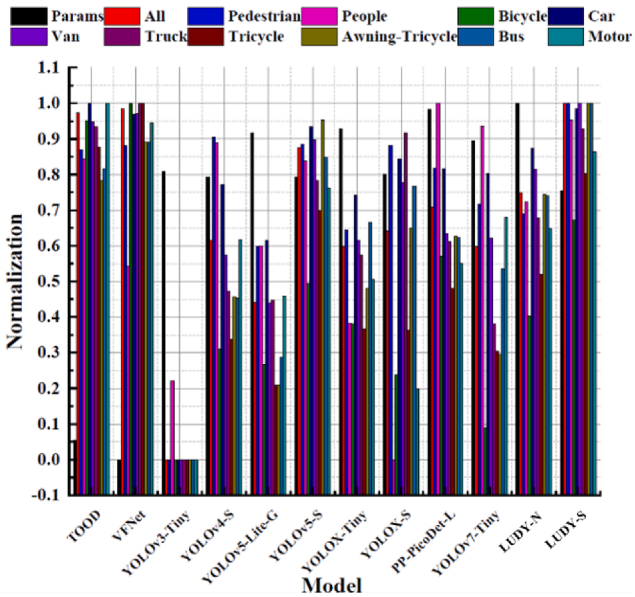


Fig. 9. Normalization analysis of multiple indicators.

Table 7
Comparison of Experimental Results.

Model	P	R	mAP(50:95)	FPS	Parameters/million	Model size/MB
YOLOv8-s	0.978	0.975	0.831	238.00	11.127	21.968
YOLOv8-n	0.971	0.963	0.773	278.00	3.011	6.079
YOLOv7tiny-silu	0.961	0.960	0.709	263.00	6.012	11.974
YOLOv7tiny-mobilevit-odconv	0.945	0.886	0.639	208.00	8.572	17.072
YOLOv5-MobileNetv3	0.964	0.963	0.728	257.00	3.553	7.431
YOLOv5n-Bifpn	0.969	0.964	0.768	268.00	**2.513**	**5.203**
YOLOXs	0.960	0.952	0.710	55.00	8.928	35.172
MobileNetv3-SSD	0.933	0.875	0.613	156.00	3.926	15.680
LUDY-N	0.971	0.974	0.809	**277.00**	3.262	5.313
LUDY-S	**0.981**	**0.982**	**0.862**	251.00	9.716	19.210

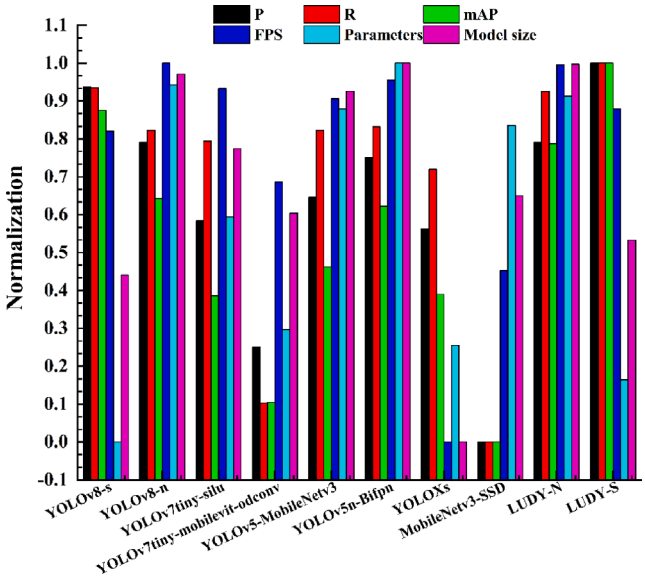


Fig. 10. Normalization analysis of multiple indicators.

shortcomings of small object detection in UAVs. To ensure that UAVs can have high detection accuracy under the premise of real-time detection, three effective improvement strategies are proposed. First, this paper proposes a new image feature fusion method, which uses adaptive spatial fusion operation to filter the features in the multi-stage fusion process, so as to achieve the feature fusion with higher quality and smaller semantic gap. Second, a novel feature extraction module, C2f BRA, is constructed to introduce sparse representation of features into Backbone, and take advantage of the self-attention mechanism to capture long-distance contextual semantic connections to reduce model parameters and improve inference speed. Third, perform pruning operations on the currently proposed LUD-YOLO to minimize the complexity of the network while ensuring accuracy. Finally, the experimental results show that, compared with the original YOLOv8, LUDY-N and LUDY-S achieve excellent detection performance on the two types of datasets VisDrone2019 and UAVDT, the indexes mAP and FPS are greatly improved, and the number of parameters and model size are greatly reduced compared with the benchmark model, providing deployment possibilities for the proposed model to realize object detection on UAVs.

This paper proposes a lightweight object detection model suitable for UAVs, but only for high-quality labeled data scenarios. In future Work, we will explore how to improve the detection accuracy of the model with a small amount of labeled data or even no labeled data from the perspective of small-sample learning and transfer learning methods, so as to reduce the dependence on a large number of labeled data. When the model is deployed to the UAVs to carry out related tasks, it can achieve faster, more accurate and more scene object detection.

CRediT authorship contribution statement

Qingsong Fan: Writing – original draft, Validation, Methodology, Conceptualization. **Yiting Li:** Writing – original draft, Visualization, Methodology, Investigation, Data curation. **Muhammet Deveci:** Writing – review & editing, Visualization, Supervision. **Kaiyang Zhong:** Writing – review & editing, Supervision, Investigation. **Seifedine Kadry:** Writing – review & editing, Supervision, Investigation.

Declaration of competing interest

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

Data availability

The data that has been used is confidential.

Acknowledgments

“This work was supported by the National Natural Science Foundation of China [grant 52165063]; the Guizhou Provincial Basic Research Program(Basic Science of Guizhou-[2024] Youth 185); the Joint Open Fund of Guizhou Provincial Department of Education (No. [2022] 436); the Science and Technology Foundation of Guizhou Province (Qiankehe pingtai rencai-GCC [2022] No.006-1); the Guizhou Provincial Key Technology R&D Program (Qiankehe support normal [2023] No.348 and No.309, Qiankehe support normal [2022] No.165 and No.008). In addition, thanks for the computing support of the State Key Laboratory of Public Big Data, Guizhou University.”

References

[1] X. Hua, X. Wang, T. Rui, F. Shao, D. Wang, Light-weight UAV object tracking network based on strategy gradient and attention mechanism, Knowledge-Based Syst. 224 (2021) 107071.
[2] J. Rao, C. Xiang, J. Xi, J. Chen, J. Lei, W. Giernacki, M. Liu, Path planning for dual UAVs cooperative suspension transport based on artificial potential field-A* algorithm, Knowledge-Based Syst. 277 (2023) 110797.
[3] S. Bhadra, V. Sagan, S. Sarkar, M. Braud, T.C. Mockler, A.L. Eveland, PROSAIL-Net: a transfer learning-based dual stream neural network to estimate leaf chlorophyll and leaf angle of crops from UAV hyperspectral images, ISPRS J. Photogramm. Remote Sens. 210 (2024) 1–24.
[4] C. Duo, Y. Li, W. Gong, B. Li, G. Qi, J. Zhang, UAV-aided distribution line inspection using double-layer offloading mechanism, IET Gener. Transm & Distrib. (2024).
[5] J. Liu, H. Liu, R. Liu, J. Xue, Y. Li, F. Wang, Application of aeromagnetic survey to mineral exploration of Jinping, Yunnan, China by using multirotor UAV, Trans. Nonferrous Met. Soc. China 33 (2023) 1550–1558.
[6] M. Wan, G. Gu, W. Qian, K. Ren, X. Maldague, Q. Chen, Unmanned aerial vehicle video-based target tracking algorithm using sparse representation, IEEE Internet Things J. 6 (2019) 9689–9706.
[7] İ. Paçal, İ. Kunduracıoğlu, Data-efficient vision transformer models for robust classification of sugarcane, J. Soft Comput. Decis. Anal. 2 (2024) 258–271.
[8] X. Song, X. Fang, X. Meng, X. Fang, M. Lv, Y. Zhuo, Real-time semantic segmentation network with an enhanced backbone based on Atrous spatial pyramid pooling module, Eng. Appl. Artif. Intel. 133 (2024) 107988.
[9] Y. Tang, Y. Chen, S.A.S.M. Sharifuzzaman, T. Li, An automatic fine-grained violence detection system for animation based on modified faster R-CNN, Expert Syst. Appl. 237 (2024) 121691.
[10] H.- Liu I, Y.-W. Tseng, K.-C. Chang, P.-J. Wang, H.-H. Shuai, W.-H. Cheng, A DENOISING FPN with transformer R-CNN for tiny object detection, IEEE Trans. Geosci. Remote Sens. 62 (2024).
[11] X. Han, Z. Qu, S.-Y. Wang, S.-F. Xia, S.-Y. Wang, End-to-end object detection by sparse R-CNN with hybrid matching in complex traffic scenes, IEEE Trans. Intell. Veh. 9 (2024) 512–525.

[12] Y. Chen, H. Xu, X. Zhang, P. Gao, Z. Xu, X. Huang, An object detection method for bayberry trees based on an improved YOLO algorithm, Int. J. Digit. EARTH 16 (2023) 781–805.
[13] Y. Xie, J. Jiang, H. Bao, P. Zhai, Y. Zhao, X. Zhou, G. Jiang, Recognition of big mammal species in airborne thermal imaging based on YOLO V5 algorithm, Integr. Zool. 18 (2023) 333–352.
[14] U. Sirisha, S.P. Praveen, P.N. Srinivasu, P. Barsocchi, A.K. Bhoi, Statistical analysis of design aspects of various YOLO-based deep learning models for object detection, Int. J. Comput. Intell. Syst. 16 (2023).
[15] Q. Gu, H. Huang, Z. Han, Q. Fan, Y. Li, GLFE-YOLOX: Global and local feature enhanced YOLOX for remote sensing images, IEEE Trans. Instrum. Meas. (2024).
[16] F.Y. Zhou, Y.S. Chao, C.Z. Wang, X.C. Zhang, H.Y. Li, X.F. Song, A small sample nonstandard gear surface defect detection method, Measurement 221 (2023).
[17] F.M. Talaat, H. ZainEldin, An improved fire detection approach based on YOLO-v8 for smart cities, NEURAL Comput. Appl. (2023).
[18] W. Yin, G. Dong, Y. Zhao, R. Li, Coresets based asynchronous network slimming, Appl. Intell. 53 (2023) 12387–12398.
[19] D. Du, P. Zhu, L. Wen, X. Bian, H. Lin, Q. Hu, T. Peng, J. Zheng, X. Wang, Y. Zhang, VisDrone-DET2019: The vision meets drone object detection in image challenge results. Proc. IEEE/CVF Int. Conf. Comput. vis Work., 2019.
[20] Z.Z. Sun, X.G. Leng, Y. Lei, B.L. Xiong, K.F. Ji, G.Y. Kuang, BiFA-YOLO: a novel YOLO-based method for arbitrary-oriented ship detection in high-resolution SAR images, Remote Sens. 13 (2021).
[21] W. Cai, Z. Wei, Remote sensing image classification based on a cross-attention mechanism and graph convolution, IEEE Geosci. Remote Sens. Lett. 19 (2020) 1–5.
[22] Y. Li, Q. Fan, H. Huang, Z. Han, Q. Gu, A modified YOLOv8 detection network for UAV aerial image recognition, Drones 7 (2023) 304.
[23] C. Feng, Y. Zhong, Y. Gao, M.R. Scott, W. Huang, Tood: Task-aligned one-stage object detection, in: 2021 IEEE/CVF Int. Conf. Comput. Vis., IEEE Computer Society, 2021: pp. 3490–3499.
[24] H. Zhang, Y. Wang, F. Dayoub, N. Sünderhauf, VarifocalNet: An IoU-aware Dense Object Detector, in: 2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021: pp. 8510–8519.
[25] J. Cao, W. Bao, H. Shang, M. Yuan, Q. Cheng, GCL-YOLO: a ghostconv-based lightweight YOLO network for UAV small object detection, Remote Sens. 15 (2023).
[26] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: Optimal speed and accuracy of object detection, ArXiv Prepr. ArXiv2004.10934 (2020).
[27] B. Yan, P. Fan, X.Y. Lei, Z.J. Liu, F.Z. Yang, A real-time apple targets detection method for picking robot based on improved YOLOv5, Remote Sens. 13 (2021).
[28] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, ArXiv Prepr. ArXiv2107.08430 (2021).
[29] G. Yu, Q. Chang, W. Lv, C. Xu, C. Cui, W. Ji, Q. Dang, K. Deng, G. Wang, Y. Du, PP-PicoDet: a better real-time object detector on mobile devices, ArXiv Prepr. ArXiv2111.00902 (2021).
[30] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2023: pp. 7464–7475.
[31] G. Song, H. Du, X. Zhang, F. Bao, Y. Zhang, Small object detection in unmanned aerial vehicle images using multi-scale hybrid attention, Eng. Appl. Artif. Intel. 128 (2024) 107455.
[32] Y. Dai, P. Zhao, Y. Wang, Maturity discrimination of tobacco leaves for tobacco harvesting robots based on a Multi-Scale branch attention neural network, Comput. Electron. Agric. 224 (2024) 109133.
[33] Y. Wu, Y. Tang, T. Yang, An improved nighttime people and vehicle detection algorithm based on YOLO v7, in: 2023 3rd Int. Conf. Neural Networks, Inf. Commun. Eng., IEEE, 2023: pp. 266–270.
[34] Y. Xiang, J. Zhao, W. Wu, C. Wen, Y. Cao, Automatic object detection of construction workers and machinery based on improved YOLOv5, Int. Conf. Green Build. Civ. Eng. Smart City, Springer, 2022, pp. 741–749.
[35] S. Wang, Q. Dong, X. Chen, Z. Chu, R. Li, J. Hu, X. Gu, Measurement of asphalt pavement crack length using YOLO V5-BiFPN, J. Infrastruct. Syst. 30 (2024) 4024005.
[36] X. Xia, X. Chai, Z. Li, N. Zhang, T. Sun, MTYOLOX: Multi-transformers-enabled YOLO for tree-level apple inflorescences detection and density mapping, Comput. Electron. Agric. 209 (2023) 107803.
[37] N. Anggraini, S.H. Ramadhani, L.K. Wardhani, N. Hakiem, I.M. Shofi, M.T. Rosyadi, Development of face mask detection using SSDLite mobilenetv3 small on raspberry Pi 4, 5th Int. Conf. Comput. Informatics Eng, IEEE 2022 (2022) 209–214.