Computing and Informatics, Vol. 44, 2025, 445–466, doi: 10.31577/cai_2025_2_445

EFFICIENT DRONE DETECTION METHOD BASED ON YOLOV8S IMPROVEMENT

Bing Su, Jie Zhang, Yifeng Lin

Changzhou University
School of Computer and Artificial Intelligence
No. 2468, YanZeng West Rd, Wujin District
Changzhou City, Jiangsu Province, China
e-mail: s22150812060@smail.cczu.edu.cn

Abstract. Combating illegal drone activities is an important task for national defense and security. How to spot drones quickly and accurately is the key. While there are many ways to detect drones, their reasoning is generally slow and complex. Therefore, in this work, we propose an improved and efficient UAV detection method YOLOv8s-C3AS based on YOLOv8s. There are three main improvements to this approach: First, we propose a new Coordinate Channel Spatial Attention Module (CCSM) and add it to the backbone of the model to enable better feature extraction. Secondly, in order to solve the scale inconsistency problem of YOLOv8s PANet, we propose a new adaptive fusion feature network (PANet-AF), which enables the model to fuse the features of the three scales better, which enables the model to better integrate features of different scales. Third, we use a more reasonable bounding box regression loss function SIoU, which improves the detection accuracy of the model without cost. Finally, we refined and made public the drone dataset and conducted a series of experiments combined with the PASCCOL VOC dataset. Our proposed approach achieves 77.2 % mAP, 98.9 % mAP_{50}, 87.1 % mAP_{75} and 120.5 FPS on the drone dataset. Experiments demonstrate that our proposed method outperforms other methods by achieving high detection accuracies while maintaining faster inference speed and lower model parameters. The drone datasets used for this research has been uploaded to kaggle: https://www.kaggle.com/datasets/zhangtutu123/drone-dataset123/dat.

Keywords: Drone detection, attention module, multi-scale feature fusion, SIoU

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs), also known as drones, are being used more and more widely due to the rapid technology development. Although we are under the impression that the impact of drones is mostly beneficial, e.g. some military applications and the exploration of farmland in remote areas [1]. However, drones have also been used for illegal activities such as drug trafficking, gun smuggling, or posing a threat to security-sensitive sites such as airports and nuclear power plants [2]. Combating illegal drone activities is an important task for national defence and security, and how to detect drones quickly and accurately is the key. Recently proposed detection methods primarily rely on radar [3], or utilize radio frequency and acoustic technologies [4, 5] to achieve detection tasks. However, UAV targets are characterised by the high flight altitude, speed and miniaturisation [6], making the radar and RF detection methods very difficult and costly [7, 8]. Although the sound detection method is simple, the detection position of the target may be disturbed by noise [9]. Therefore, there is an urgent need to investigate an advanced UAV detection method.

In the past decade, deep learning has developed rapidly and has been applied to a large number of fields, such as agriculture, facial recognition, medical imaging, and drone detection tasks [10]. At this stage, the object detection algorithms are mainly divided into two-stage and one-stage algorithms. The most classic two-stage representatives are the Faster R-CNN [11] of the R-CNN series and FPN [12]. The representative of the first-stage algorithm is SSD [13], RetiNet [14], FCOS [15] and the YOLO series. Due to the rise of CNNs, so many researchers use CNNs for drone detection. Mahdavi and Rajabi use a CNNs as backbone of model to detect drones [16]. Compared with the SVM and nearest neighbor algorithms compared in this paper, it achieves better results, but the method in [16] has the problem of large number of parameters and slow reasoning speed, which cannot complete the task of real-time detection well. But YOLO has come up with a good solution to this problem.

Singha and Aydin chose the then state-of-the-art target detection algorithm YOLOv4 to detect UAVs and achieved 74.3 % mAP in a self-built dataset [17]. However, YOLOv4 officially does not provide a lightweight version, so the model itself still has a large number of parameters. Aiming at the problem of large number of parameters in the YOLOv4 model, Cheng et al. present a novel drone detection method, YOLOv4-MCA [18], based on the lightweight MobileViT and Coordinate Attention. This method is improved compared to the YOLOv4 framework method, replacing YOLOv4 backbone with lightweight MobileViT, incorporating coordinate attention into YOLOv4's PANet. This method achieves 92.81 % mAP_{50} and 40FPS on the self-built drone dataset. However, the accuracy of the model compared with the YOLOv4 benchmark under the VOC dataset decreases. In the above methods, there are trade-offs between model accuracy, speed, and parameters.

In order to better weigh these indicators, we propose an efficient real-time lightweight drone detection method YOLOv8s-C3AS, which is improved based on YOLOv8s. This article offers the following contributions:

1. We propose a new Coordinate Channel Spatial Attention Module (CCSM) that allows attention blocks to capture remote dependencies between channels along the channel direction while retaining precise location information along another spatial direction, while also giving the model a spatial sense, which helps the network locate pairs of interest more accurately.
2. In order to solve the scale inconsistency problem of YOLOv8s PANet, we propose a new adaptive fusion feature network PANet-AF, which enables the model to fuse the features of the three scales better.
3. Since the CIoU used by YOLOv8s bbox_loss lacks the angle factor, we replace it with a more suitable SIoU [19].
4. Since the dataset made public by [20] lacked labels, we relabelled it and made it public.

2 RELATED WORK

This section is mainly used to introduce the development and application of related methods, including the development of the YOLO series, the method of attention mechanism, ASFF, and some research work of bbox_loss.

2.1 The Development of the YOLO Series

Since the rise of convolutional neural networks, efforts to achieve higher precision have resulted in models becoming larger and deeper, which led to the fact that the model of the object detection algorithm at that time was very large, difficult to train, and the inference speed of the model was very slow. To address these issues, Redmon et al. proposed You Only Look Once (YOLO) [21], a simple architecture, and its one-stage model, which is faster than other object detection models. However, the trade-off for fast speed is a lack of sufficient accuracy. In order to solve the accuracy problem, Redmon et al. proposed YOLOv2, v3 [22, 23], while maintaining speed they improved accuracy, and thus YOLOv3's accuracy is comparable to SOTA technology at that time. Subsequent versions are all improved on the basic framework of YOLOv3, the latest is YOLOv8, which has excellent performance and provides different sizes of models to achieve a balance of accuracy and speed, so that developers can choose the most suitable model to complete their tasks.

2.2 Attention Mechanisms

With the rise of Transformer in recent years, more and more researchers have begun to pay attention to the attention mechanisms. Attention mechanisms that have been

shown to be effective include SE (Squeeze and Excitation), CBAM (Convolutional Block Attention Module), CA (Coordinate Attention), etc. [24, 25, 26]. Cao and Yuan proposed a real-time detection of mango based on improved YOLOv4 [27]. By adding CBAM module to YOLOv4 to improve the detection accuracy, the improved model is 3.93 % higher than YOLOv4, which can identify mangos more accurately. Cheng et al. present a novel drone detection method, YOLOv4-MCA [18], based on the lightweight MobileViT and coordinate attention. It utilizes coordinate attention to improve YOLOv4's PANet. This method achieves a high detection accuracy for multi-scale drone targets.

2.3 Multi-Scale Feature Fuse

In previous works, the multi-scale problems in object detection were usually solved by the pyramidal feature representation. However, the inconsistency across different feature scales is a primary limitation for the single-shot detectors based on the feature pyramid. So Liu et al. propose a novel and data driven strategy for pyramidal feature fusion, referred to as adaptively spatial feature fusion (ASFF) [28]. It fuses spatial information of different scales to suppress inconsistency between different feature scales.

2.4 Bbox_loss

In addition to improving the structure of the model, the improvement of bbox_loss is also one of the important parts of object detection. From the initial L1, L2 Loss, to IoU, GIoU, DIoU, CIoU Loss, and the recently proposed SIoU Loss, etc. [19, 29, 30, 31], bbox_loss design is becoming more and more reasonable and efficient. Lv et al. [32] replaced the original DIoU loss with α-DIoU loss to improve the accuracy of boundary box regression in UAV detection. Therefore, it is essential to select a suitable bbox_loss.

Combined with the improvement of the above related work, we propose an efficient and lightweight real-time UAV detection method YOLOv8s-C3AS based on YOLOv8s.

3 MODEL DESIGN

This section mainly introduces the overall model structure of YOLOv8s-C3AS and the principles of related improvement methods.

3.1 YOLOv8s-C3AS

In this section, we will introduce the overall structure of the YOLOv8s-C3AS model, which are Backbone, Neck, and Head. The overall structure of the model is shown in Figure 1. A brief description of how to improve each section follows:

Bakcbone: This algorithm enhances the original backbone of YOLOv8s, and we add a new coordinate channel spatial attention module (CCSM) after the first three stages in the backbone. The specific location of this insertion is shown in Figure 1. After integrating the CCSM module, the model not only gains the ability to capture the relationships between channels and coordinate information but also develops spatial awareness. This enhancement helps suppress irrelevant feature information, improving the model's feature extraction capability and increasing its focus on small targets.

Neck: We propose a new feature fusion network, PANet-AF, which allows the original three independent branches to fuse and enhance information of different scales through an attention mechanism. It is the addition of three adaptive fusion modules (AFM) to the original PANet of YOLOv8s.

Head: For the structure of the prediction header we follow YOLOv8s and keep it unchanged. However, in terms of the loss function, we replace the box_loss, which is used in YOLOv8s, with the more appropriate SIoU loss, which has proved to be effective after experiments.

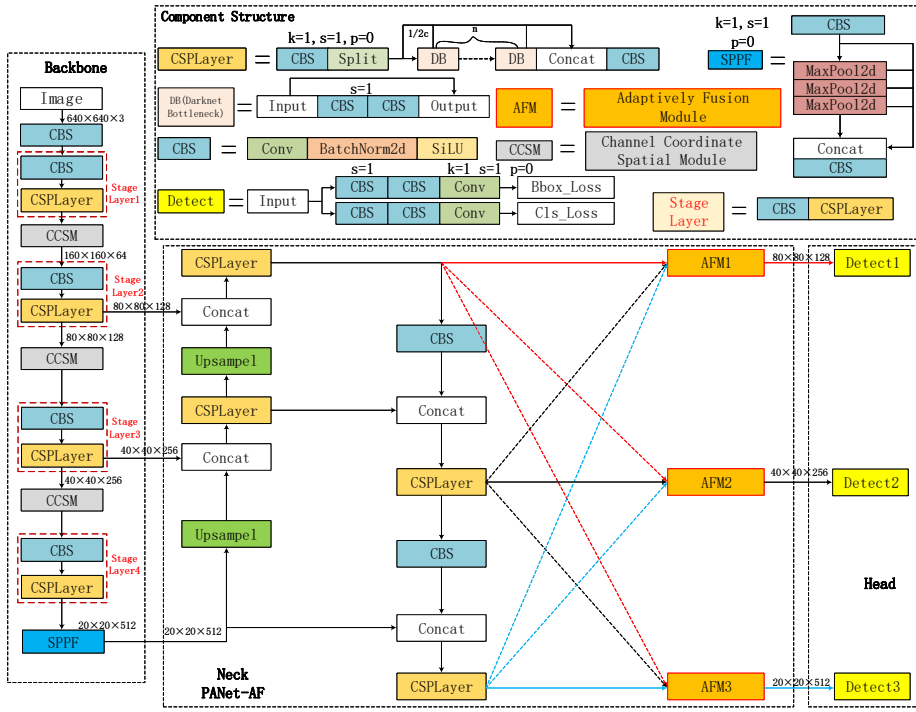


Figure 1. YOLOv8s-C3AS model structure diagram

Other modules, CBS block is composed of a convolution block, a normalization layer, and an activation function (SiLU). The convolution block in the figure uses the default parameters (kernel = 3, stride = 2, padding = 1) if the parameters are not specifically labeled. The Stage Layer module and the CCSM module are the main structures of the backbone. CSPLayer is the key component of Stage Layer block, which is not only used in backbone, but also in neck. CSPLayer is composed of several DB blocks. The SPPF block consists mainly of 3 maximal pooling layers with different pooling cores. Note that SPPF is included in the Stage Layer4 Block, which is also different from the previous Stage Layer blocks. Figure 1 shows the overall structure of YOLOv8s-C3AS and the structure of each block.

3.2 Channel Coordinate Spatial Module

3.2.1 Other Common Shortcomings of the Attention Module

Hu et al. [24] proposed a Squeeze-and-Excitation (SE) block to obtain the corresponding weights for each channel. But it only has channel attention, ignoring the importance of spatial attention. Woo et al. [25] proposed the Convolutional Block Attention Module (CBAM), which combines channel attention and spatial attention. As a plug-and-play module, it can be embedded into convolutional neural networks to improve network performance. Although SE and CBAM enable the network to achieve good performance, Hou et al. [26] found that the compression characteristics of SE and CBAM lost too much information and ignored the coordinate information. Therefore, they proposed Lightweight Coordinate Attention (CA) to solve this problem.

However, CA also has shortcomings, it only focuses on the information of features in the horizontal and vertical directions, missing the information of the overall spatial position, such as the bias of the target of interest in the overall space, we simply call this process spatial sense. The coordinate information can only let the model know where the target is located (only one (x, y) coordinate vertical information), but it cannot let the model know where the target appears in the overall target deviation, whether it is centered or it is left or right, or up and down. Than means it cannot let the model have a sense of space. Therefore, we propose a CCSM module that captures the relationships and coordination information between channels while also providing the model with a sense of space.

3.2.2 CCSM Module Details

CCSM combines the advantages of CBAM and CA, as shown in Figure 2 c), which is mainly divided into two parts.

The top half. In the first half, the remote dependencies between channels are captured along the channel direction, while the precise location information is retained along the other spatial direction, which helps the network locate the object of interest more accurately.

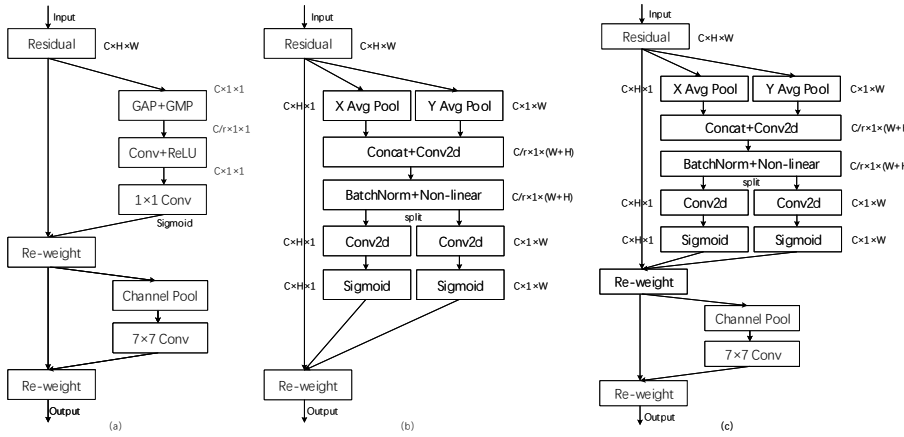


Figure 2. Structure comparison of a) CBAM, b) CA and c) CCSM

The calculation of input feature map can be divided into two parts: coordinate information embedding and coordinate information generation.

Coordinate Information Embedding. The embedded part is abstracted into two one-dimensional features in the horizontal and vertical directions. The specific operation is, using the pooling check of size $(H, 1)$ or $(1, W)$ to pool the input feature X, then the output of channel c with height H can be expressed as:

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

Similarly, the output of c^{th} channel with width w can be written as:

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w). \tag{2}$$

Coordinate Information Generation. The steps for generating coordinate information are as follows: First, coordinate information is embedded in Equation (1) and the aggregate feature maps generated by Equation (2) are connected, and then they are sent to the shared 1×1 convolution transform function f to obtain the concatenated features, and then through a BatchNorm and nonlinear activation function, it can be expressed as:

$$F_{h,w} = \delta(\langle f^{1\times1}([z^h, z^w]) \rangle), \tag{3}$$

where δ is the nonlinear activation function, $\langle\ \rangle$ is the BatchNorm layer, and $[\times]$ denotes the concatenation operation along the spatial dimension.

Then, we first divide $F_{h,w}$ into two tensors, F_h and F_w, along the spatial dimension. And then use two other 1×1 convolution transformations f, to transform F_h and F_w, respectively, into tensors with the same channel number as the input x.

$$\begin{aligned} g^h &= \sigma(f_h^{1\times1}(F_h)), \\ g^w &= \sigma(f_w^{1\times1}(F_w)). \end{aligned} \tag{4}$$

Finally, the output of our coordinate attention block Y can be written as:

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j). \tag{5}$$

The bottom half. This part is mainly used to generate spatial attention and give the model a sense of space. It is only necessary to send the input feature through the channel pooling layer (P_c), and then through a 7×7 size convolution layer, which can be written as:

$$y_s = f^{7\times7}(P_c(x)). \tag{6}$$

Finally, the process of CCSM module can be summarized as follows:

$$\begin{aligned} F_1 &= y_c(x) \otimes x, \\ F_2 &= y_s(F_1) \otimes F_1, \end{aligned} \tag{7}$$

where $\otimes$ represents element-by-element multiplication with feature mappings; x represents the input feature; y_c represents the upper half of CCSM to capture the relationship and coordinate information between feature channels; y_s represents the computational spatial attention feature to give the model a sense of space, and F_2 is the final feature obtained by the CCSM module.

3.3 PANet-AF

We refer to the idea of [28] to improve the neck of YOLOv8s. The improved neck is called PANet-AF. However, in [28] the required parameter cost is so high that we have simplified it, which we will call the Adaptively Fusion Module (AFM) in this article. It enables the model to learn how to spatially filter conflicting information to suppress inconsistencies, thereby improving scale invariance.

AFM. Its key idea is to adaptively learn the spatial weight of fusion for feature maps at each scale. AFM consists of two steps: feature map scaling and adaptive fusion, as shown in Figure 3.

1) **Feature Resizing.** Set the resolution of the L layer ($L \in 1, 2, 3$) to x^L. For the other layers N ($N \neq L$) the feature x^N will be adjusted to the same shape as x^L. Since the three scales of features in YOLOv8 have different resolutions and channels, corresponding adjustment strategies are also required, where for the up-sampling and down-sampling strategies we follow [28].

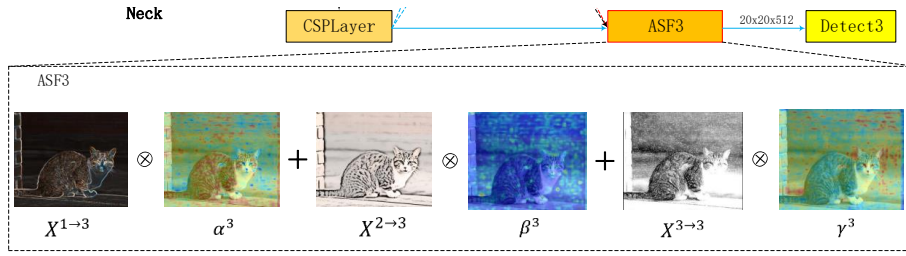


Figure 3. Adaptive spatial feature fusion mechanism process, where the cat image is from [28]

2) Adaptive Fusion. Let $x_{ij}^{N\to L}$ be the feature vector of the N^{th} layer adjusted to the position corresponding to (i, j) of the L^{th} layer feature. The features of the L layer are fused as follows:

$$y_{ij}^{L} = \alpha_{ij}^{L} x_{ij}^{1\to L} + \beta_{ij}^{L} x_{ij}^{2\to L} + \gamma_{ij}^{L} x_{ij}^{3\to L}, \tag{8}$$

where y_{ij}^{L} denotes the fusion output of the corresponding (i, j) feature vector in the L^{th} layer. $\alpha_{ij}^{L}, \beta_{ij}^{L}, \gamma_{ij}^{L}$ are importance weights for the feature maps at three different levels after feature resizing to level L, which are adaptively learned by the network. Provided that $\alpha_{ij}^{L} + \beta_{ij}^{L} + \gamma_{ij}^{L} = 1$ and $\alpha_{ij}^{L}, \beta_{ij}^{L}, \gamma_{ij}^{L} \in [0, 1]$, and we can define:

$$\alpha_{ij}^{L} = \frac{e^{\lambda_{\alpha_{ij}^{L}}^{L}}}{e^{\lambda_{\alpha_{ij}^{L}}^{L}} + e^{\lambda_{\beta_{ij}^{L}}^{L}} + e^{\lambda_{\gamma_{ij}^{L}}^{L}}}, \tag{9}$$

where $\lambda_{\alpha_{ij}^{L}}^{L}, \lambda_{\beta_{ij}^{L}}^{L}, \lambda_{\gamma_{ij}^{L}}^{L}$ are computed using a 1×1 convolution for the three feature maps $x^{1\to L}, x^{2\to L}, x^{3\to L}$, and a softmax function is used to define the $\alpha_{ij}^{L}, \beta_{ij}^{L}, \gamma_{ij}^{L}$. The feature resizing and adaptive fusion described above will operate at each of the three different scales.

AFM-sim. For the simplified version of AFM-sim, the overall process is the same as the previous AFM, except that the Feature Resizing part is different. Instead of using convolution for downsampling and channel scaling, the Focus layer in YOLOv5 [33] is used to increase the channels of the feature map and reduce the resolution. At the same time, an averaging operation is used for channel compression. Note: The AFM modules we used in this experiment are all AFM-sim.

3.4 SIoU Loss

YOLOv8s bbox_loss uses the CIoU Loss, which takes into account only the aspect ratio of the prediction box and gtbox, the IoU, and the distance between the box

centers. An important influencing factor angle cost is ignored. We therefore use the SIoU Loss recently proposed by Gevorgyan [19] to replace the CIoU Loss. SIoU Loss function consists of 4 cost functions: Angle cost, Distance cost, Shape cost, IoU cost. Follow-up experiments also proved that our idea was right.

1) Angle cost. The basic idea is to return the prediction box to the X or Y axis first (prioritizing smaller angles) and then continue along the X or Y axis. Based on this idea, the model will first try to minimize α if $\alpha \leq \pi/4$ during training, otherwise minimize $\beta = \pi/2 - \alpha$. Angle cost is defined as follows:

$$\begin{aligned} A &= 1 - 2 * \sin^2(\arcsin(x) - \pi/4), \\ x &= \frac{c_h}{\sigma} = \sin(\alpha), \\ A &= 1 - 2 * \sin^2(\alpha - \pi/4) = \cos(2\alpha - \pi/2) = \sin(2\alpha), \end{aligned} \tag{10}$$

where

$$\begin{aligned} \sigma &= \sqrt{(b_{c_x}^{gt} - b_{c_x})^2 + (b_{c_y}^{gt} - b_{c_y})}, \\ c_h &= \max(b_{c_y}^{gt}, b_{c_y}) - \min(b_{c_y}^{gt}, b_{c_y}). \end{aligned} \tag{11}$$

Some of the specific variables are shown in Figure 4.

2) Distance cost. Combine the angle cost defined above into the new distance cost:

$$\begin{aligned} D &= \sum_{t=xy} (1 - e^{-\gamma \rho_t}), \\ \rho_x &= \left(\frac{b_{c_x}^{gt} - b_{c_x}}{c_w} \right)^2, \quad \rho_y = \left(\frac{b_{c_y}^{gt} - b_{c_y}}{c_h} \right)^2, \quad \gamma = 2 - A. \end{aligned} \tag{12}$$

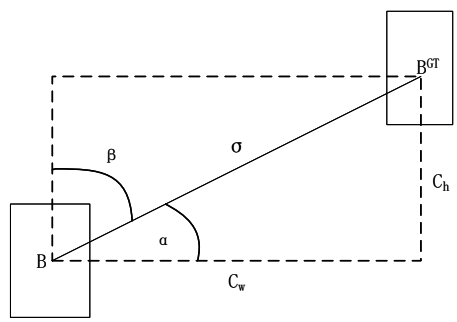


Figure 4. Schematic diagram of SIoU Loss variables

When $\alpha \to 0$, then the angle loss $A = 0, \gamma = 2$, the effect of angle loss becomes smaller and the contribution of distance loss in the whole SIoU Loss becomes

larger, so that the distance loss is mainly optimized. When $\alpha \to \pi/4, \gamma = 1$, the angle has the greatest effect on the distance loss, and the contribution of the distance loss becomes smaller, so the angle will be optimized first. Thus as the angle increases, the priority of the angle becomes progressively greater than that of the distance.

3) Shape cost. Shape loss definition:

$$\begin{aligned} S &= \sum_{t=w,h} (1 - e^{-\omega_t})^{\theta}, \\ \omega_w &= \frac{|w - w^{gt}|}{\max(w, w^{gt})}, \quad \omega_h = \frac{|h - h^{gt}|}{\max(h, h^{gt})}. \end{aligned} \tag{13}$$

θ value controls how much the model focuses on shape cost. We follow the setting of [19] and also set θ to 4 in this paper.

4) Total formula of SIoU loss function. The formula of SIoU loss can be obtained by combining the three loss functions mentioned above, namely Angle loss, Distance loss and Shape loss:

$$\begin{aligned} L_{box} &= 1 - \mathrm{IoU} + \frac{D + S}{2}, \\ \mathrm{IoU} &= \frac{|B \cap B^{GT}|}{|B \cup B^{GT}|}. \end{aligned} \tag{14}$$

4 DATASET AND EXPERIMENT DETAILS

4.1 Dataset Preparation

The drone dataset used in this paper is a publicly available dataset constructed by Aksoy et al. [20]. Since the data set provided in literature [20] lacked labels, we used labeling software to re-label and expose it. The drone dataset consists of 3 sizes of targets, small, medium and large, totaling 3752 images and 4784 targets. We divide the dataset into training set and test set in the ratio of 8:2, i.e., 3002 images in the training set and 750 images in the test set. The distribution of the drone dataset is specifically shown in Table 1.

In addition to using the UAV dataset, this paper also uses the PASCAL VOC $07 + 12$ dataset for generalization performance experiments. The VOC dataset [34] has a total of 20 categories containing images and labels for classification, detection, segmentation, and human body layout. VOC $07 + 12$ is a union of the training and test sets of PASCAL 2007 and 2012, which were then tested on the test set of PASCAL 2007. A total of 16 551 training images and 4 952 test images are included.

Class	Number of Images	Number of Targets	Large	Medium	Small
train	3 002	3 829	1 118	1 686	1 025
test	750	955	262	435	258
total	3 752	4 784	1 380	2 121	1 283

There may be multiple targets in a single image.
We removed several images that are difficult to identify by human eyes, and added several untargeted images to enhance the generalization ability of the model. As for the test set, it is the same as the test set.

Table 1. Drone-dataset's targets information

4.2 Experimental Environment and Training Details

The experimental equipment uses a Tesla V100 (32 G) for training and a 3070ti (8 G) to test the inference speed of the model.

YOLOv8s-C3AS are still augmented with the Mosica dataset from YOLOv5 [33], while the Mosica dataset augmentation is turned off for the last 10 epochs of the training phase as proposed by YOLOX [35]. Table 2 shows the hyperparameter information used in model training. Figure 5 shows the trend of the model training loss. Where loss_cls represents the classification loss, loss_bbox represents the bounding box regression loss, and loss_dfl is the Distribution Focal Loss proposed in [36]. The total loss is obtained by applying a certain percentage of weighting to the other three losses.

For the VOC $07+12$ dataset we used the AP_{50} metrics to evaluate. Instead, the drone dataset is evaluated using MSCOCO-style evaluation metrics Average Precision, including $AP, AP_{50}, AP_{75}, AP_s, AP_m$, and AP_l, where AP_s, AP_m, AP_l represent the average precision of small, medium, and large scale objects.

Type	Parameter	Note
Image size	640×640	Image input size
Epoch	200	Total training times
Batch size	16 or 8	Freeze size or Normal size
Learning rate	0.01 and 0.0001	Initial and Minimum rate
Optimizer	SGD	Optimizer type
Momentum	0.937	Momentum of optimizer
Weight decay	0.0005	The decay of weights
Learning rate schedule	Linear	Learning rate adjustment strategy

Table 2. The hyperparameter setting of model training

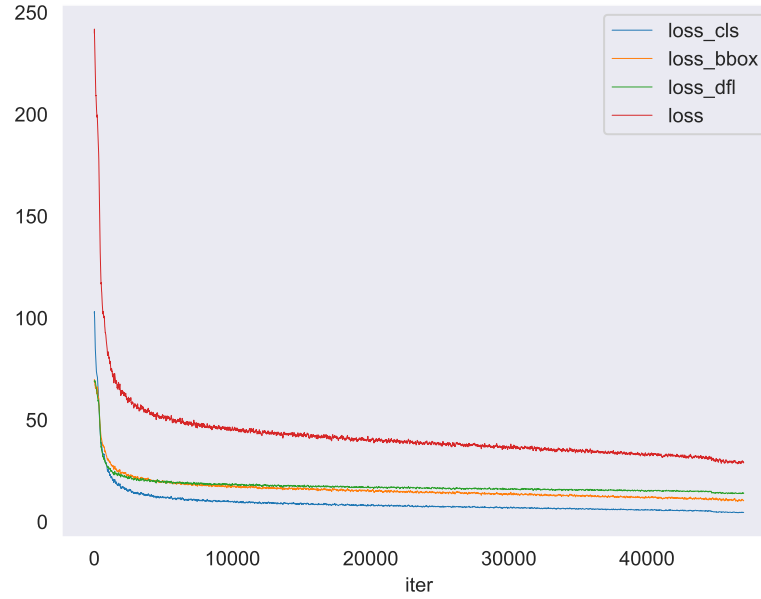


Figure 5. The loss value of YOLOv8s-C3AS on drone dataset

5 EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we will use a series of methods to validate our proposed approach. First, ablation experiments are conducted on the YOLOv8s-C3AS algorithm in the drone dataset and the utility of each improved strategy is explored. Secondly, the effectiveness comparison experiment is carried out on the drone dataset, which verifies that our algorithm is more effective for UAV detection than other algorithms. Finally, we conduct a comparison experiment with other commonly used algorithms on the PASCAL VOC $07+12$ dataset, and prove that YOLOv8s-C3AS model is not only suitable for specific data sets but also has a strong generalization ability.

5.1 Ablation Experiments

5.1.1 Ablation Experiment of CCSM Module

To verify the effectiveness of our proposed CCSM module, we conducted ablation experiments on the basis of YOLOv8s. We compare commonly used Attention modules such as SE (Squeeze and Excitation), CBAM (Convolutional Block Attention Module), CA (Coordinate Attention), and our proposed CCSM (Coordinate

Channel Spatial Attention Module). We add these four modules respectively to the backbone of YOLOv8s, and the specific insertion location is shown in CCSM in Figure 1.

Model	FLOPS(G)	Param(M)	FPS	mAP	mAP_{50}	mAP_{75}
YOLOv8s	28.5	11.147	130.6	76.2	98.5	84.7
+ SE	28.7	11.189	128.6	76.3	98.5	85.1
+ CBAM	28.9	11.213	123.3	76.5	98.6	85.7
+ CA	28.6	11.171	125.3	76.7	98.5	86.5
+ CCSM (our)	28.7	11.165	128.3	**76.8**	**98.7**	**86.9**

Table 3. Experimental results of different attention modules on drone dataset

As it can be seen from Table 3, the CCSM module we proposed only adds a little FLOPS and the number of parameters, the speed is almost the same as YOLOv8s, but the accuracy is better than other attention modules.

5.1.2 YOLOv8s-C3AS Model Ablation Experiment

In order to verify that each of our proposed improvements is effective, we conducted additional ablation experiments. We experimented with each of the improved parts of YOLOv8s-C3AS one by one on the drone dataset.

Model	Method			mAP	mAP_{50}	mAP_{75}	FPS	Parameter (M)
	CCSM	PANet-AF	SIoU					
YOLOv8s	×	×	×	76.2	98.5	84.7	130.6	11.136
YOLOv8s-C3	✓	×	×	76.8	98.7	86.9	128.3	11.147
YOLOv8s-A	×	✓	×	76.9	98.8	85.7	122.2	12.240
YOLOv8s-C3A	✓	✓	×	77.0	98.8	86.9	120.5	12.251
YOLOv8s-C3AS	✓	✓	✓	**77.2**	**98.9**	**87.1**	120.5	12.251

Table 4. The ablation experiment based on drone dataset

From the results in Table 4, it can be seen that YOLOv8s has an accuracy of 76.2 % for mAP, 98.5 % for mAP_{50}, 84.7 % for mAP_{75}, an inference speed of 130.6 f/s, and a model parameter count of 11.136 M. The improvement based on CCSM is very good, resulting in a 0.6 % improvement in the model's mAP with a negligible decrease in the FPS and negligible increase in the number of parameters. The AP_{50} improves by 0.2 % and the AP_{75} improves by 2.2 %.

C3 indicates that 3 CCSM modules are inserted (see Figure 1 for details), here we only chose to insert CCSM after the first 3 Stage Layers because Stage Layer 4 is immediately followed by the SPPF module. Because the bottom half of CCSM (the part that computes spatial attention features) will conflict with the role of MaxPool2d layer in the SPPF module, which both operate on spatial information. MaxPool2d layer will filter the spatial information extracted by the CCSM module,

which results in the reduction of the role of CCSM. The experiments in Table 5 also prove our conjecture that C4, which means adding CCSM module after each Stage Layer, can only reach 76.5 mAP in the end, which is far less effective than that of C3.

The neck improvement based on PANet-AF was also significant. It improves the mAP of the benchmark YOLOv8s by 0.8 %, mAP_{50} by 0.3 %, and mAP_{75} by 1 %, but it also carries the largest burden, with a drop in FPS of 8 f/s and an increase in the number of parameters by 1 M. The improvement in bbox_loss is the SIoU, which improves the model accuracy without adding any additional burden. Finally, compared to YOLOv8s, our algorithm YOLOv8s-C3AS improves mAP by 1 %, mAP_{50} by 0.4 %, and mAP_{75} by 2.3 %.

Despite the 1 M increase in model parameters, there is only a small impact on the model inference speed, which is reduced by only 10 f/s, which is negligible compared to the increase in accuracy. In conclusion, our model achieves an effective balance between accuracy, FPS and model complexity, and it is suitable for UAV target detection tasks.

Model	Position	mAP	mAP_{50}	mAP_{75}
YOLOv8s	(×, ×, ×, ×)	76.2	98.5	84.7
YOLOv8s-C3	(✓, ✓, ✓, ×)	76.8	98.8	86.9
YOLOv8s-C4	(✓, ✓, ✓, ✓)	76.5	98.5	86.3

Tick to add CCSM after the corresponding stage layer.

Table 5. The insertion position of the CCSM module

5.2 Comparison with Other Methods

5.2.1 Comparative Experimental Results on Drone Dataset

To verify the effectiveness of our method in drone detection tasks, we conducted comparative experiments on drone datasets with other advanced algorithms. In the UAV dataset, we use COCO's evaluation index. As can be seen in Table 6, our method has the highest accuracy among all algorithms, with 77.2 % for $mAP(0.5:0.95)$, 98.9 % for mAP_{50}, and 87.1 % for mAP_{75}. In addition, our model has the best AP performance on all three scales. Note that since YOLOv7 [37] does not provide an s-version of the model, we use YOLOv7_tiny, which has the closest parameters, as a comparison.

As it can be seen from the effect comparison graph in Figure 6, our model YOLOv8s-C3AS (lower six images) has better detection results compared to YOLOv8 (upper six images). The second, fourth and sixth images are correctly detected images while the first image is a missed detection image and the third and fifth images are incorrectly detected images. It is clear that our model YOLOv8-C3AS can avoid missed and wrong detection very well.

Model	mAP (%)	mAP_{50} (%)	mAP_{75} (%)	mAP_s	mAP_m	mAP_l
Faster R-CNN	67.6	96.8	78.7	47.2	71.0	78.1
SSD	58.7	93.2	64.8	33.4	60.0	76.9
RetinaNet	62.5	94.7	73.6	41.3	67.3	73.2
YOLOv4-MCA [18]	N/A	92.8	N/A	24.6	38.3	58.5
TDRD-YOLO [38]	N/A	96.8	N/A	N/A	N/A	N/A
SAG-YOLOv5s [32]	N/A	97.6	N/A	N/A	N/A	N/A
YOLOv5s	68.9	97.5	77.1	46.1	71.3	82.3
YOLOvxs	68.3	97.8	77.1	46.6	71.5	80.8
YOLOv6s	70.2	97.9	79.3	48.3	72.0	83.8
YOLOv7_tiny	67.0	96.2	76.4	45.2	68.8	81.5
YOLOv8s	76.2	98.5	84.7	49.1	77.7	94.9
YOLOv8s-C3AS (our)	**77.2**	**98.9**	**87.1**	**50.4**	**79.0**	**95.1**

N/A indicates not available.

Table 6. The performance comparison of algorithms on drone dataset

Model	Input Size	mAP_{50} (%)	FPS (f/s)	Parameter (M)
Faster R-CNN	$1\,000 \times 600$	79.5	29.9	41.22
SSD	300×300	73.8	85.1	26.29
RetinaNet	$1\,000 \times 600$	75.7	30.6	36.5
YOLOv4-MCA [18]	416×416	80.7	40	13.47
YOLOv5s	640×640	76.4	97.6	7.07
YOLOvxs	640×640	84.2	114.8	8.94
YOLOv6s	640×640	85.0	44.1	17.20
YOLOv7_tiny	640×640	79.5	79.1	**6.07**
YOLOv8s	640×640	84.8	**130.6**	11.14
YOLOv8s-C3AS (our)	640×640	**85.3**	120.5	12.25

FPS was tested on the RTX 3070Ti.

Table 7. The performance comparison of algorithms on PASCAL VOC 07 + 12

5.2.2 Comparative Experimental Results on PASCAL VOC Dataset

To verify that our model is generalized and not only applicable to a specific dataset, we also generalize all algorithms on the dataset of PASCAL VOC 07 + 12. The indexes used in the generality experiment are mAP_{50} (IoU = 0.5), model inference speed FPS, and model parameters.

From the experimental results in Table 7, our method mAP_{50} has the highest accuracy among all algorithms. Although YOLOv6s can achieve 85.0 % accuracy, which is already better than the common YOLOv8s, it sacrifices too much speed to meet the speed requirement of UAV detection. In contrast, our method outperforms YOLOv6s in all aspects. Our model achieves an effective trade-off between the speed and accuracy as well as parameters. It is also a good proof that our model YOLOv8s-

Figure 6. YOLOv8s (top) and YOLOv8s-C3AS (bottom) comparison results

C3AS is not only suitable for UAV datasets, but also for general datasets with strong generalization ability.

6 CONCLUSIONS

Aiming to address the issues of low precision, slow speed and high complexity of the UAV detection model, an efficient and real-time UAV detection algorithm YOLOv8s-C3AS was proposed based on YOLOv8s.

First, the YOLOv8s backbone network was improved. Except for the last level of the backbone network, the coordinate channel space attention module (CCSM) was added after each stage of the backbone network, so that the model could not only capture the feature channel and coordinate information, but it also gives the

model a spatial sense, so as to improve the feature extraction ability of the model. Secondly, we add the adaptive fusion model (ASM) after the PANet of YOLOv8s, and we call the improved neck PANet-AF. It can use the original three independent scale branches of PANet to self-adapt the information fusion and enhancement of different scales. Finally, we use a more appropriate SIoU Loss to replace the original bbox_loss of YOLOv8s, which makes up for CIoU lack of consideration of angle factors in boundary box regression, so as to further improve the detection accuracy of the model.

The experimental results indicate that the method performs well. It achieves 98.9 % mAP_{50} accuracy on drone datasets and 85.3 % mAP_{50} accuracy on VOC datasets, with a high inference speed of 120 FPS and a parameter count of only 12.25 M. While maintaining fast model inference speed and low model parameters, it has higher detection accuracy than other mainstream UAV detection algorithms. There is still room for improvement in the model's speed and accuracy, which will be addressed in future work.

REFERENCES

[1] Mandal, P.—Roy, L. P.—Das, S. K.: Internet of UAV Mounted RFID for Various Applications Using LoRa Technology: A Comprehensive Survey. In: Dahal, K., Giri, D., Neogy, S., Dutta, S., Kumar, S. (Eds.): Internet of Things and Its Applications. Select Proceedings of ICIA 2020. Springer, Singapore, Lecture Notes in Electrical Engineering, Vol. 825, 2022, pp. 369–380, doi: 10.1007/978-981-16-7637–6_33.

[2] Basak, S.—Rajendran, S.—Pollin, S.—Scheers, B.: Combined RF-Based Drone Detection and Classification. IEEE Transactions on Cognitive Communications and Networking, Vol. 8, 2022, No. 1, pp. 111–120, doi: 10.1109/TCCN.2021.3099114.

[3] Mandal, P.—Roy, L. P.—Das, S. K.: Intruder Drone Detection Using Unmanned Aerial Vehicle Borne Radar (UAVBR) via Reconfigurable Intelligent Reflective Surface (IRS). 2022 IEEE 19th India Council International Conference (INDICON), 2022, pp. 1–5, doi: 10.1109/INDICON56171.2022.10039834.

[4] Kang, H. Y.—Lee, K.: A General Acoustic Drone Detection Using Noise Reduction Preprocessing. Journal of the Korea Institute of Information Security & Cryptology, Vol. 32, 2022, No. 5, pp. 881–890, doi: 10.13089/JKIISC.2022.32.5.881 (in Korean).

[5] Fang, J.—Li, Y.—Ji, P. N.—Wang, T.: Drone Detection and Localization Using Enhanced Fiber-Optic Acoustic Sensor and Distributed Acoustic Sensing Technology. Journal of Lightwave Technology, Vol. 41, 2023, No. 3, pp. 822–831, doi: 10.1109/JLT.2022.3208451.

[6] Shao, S.—Zhu, W.—Li, Y.: Radar Detection of Low-Slow-Small UAVs in Complex Environments. 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Vol. 10, 2022, pp. 1153–1157, doi: 10.1109/ITAIC54216.2022.9836542.

[7] Mendis, G. J.—Randeny, T.—Wei, J.—Madanayake, A.: Deep Learning Based Doppler Radar for Micro UAS Detection and Classification. MILCOM 2016 – 2016 IEEE Military Communications Conference, 2016, pp. 924–929, doi: 10.1109/MILCOM.2016.7795448.

[8] Bisio, I.—Garibotto, C.—Lavagetto, F.—Sciarrone, A.—Zappatore, S.: Unauthorized Amateur UAV Detection Based on WiFi Statistical Fingerprint Analysis. IEEE Communications Magazine, Vol. 56, 2018, No. 4, pp. 106–111, doi: 10.1109/MCOM.2018.1700340.

[9] Liu, S.—Qi, L.—Qin, H.—Shi, J.—Jia, J.: Path Aggregation Network for Instance Segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[10] Terven, J. R.—Córdova Esparza, D. M.: A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. CoRR, 2023, doi: 10.48550/arXiv.2304.00501.

[11] Ren, S.—He, K.—Girshick, R.—Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, 2017, No. 6, pp. 1137–1149, doi: 10.1109/TPAMI.2016.2577031.

[12] Lin, T. Y.—Dollár, P.—Girshick, R.—He, K.—Hariharan, B.—Belongie, S.: Feature Pyramid Networks for Object Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[13] Liu, W.—Anguelov, D.—Erhan, D.—Szegedy, C.—Reed, S.—Fu, C. Y.—Berg, A. C.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9905, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[14] Apostolopoulos, S.—Ciller, C.—De Zanet, S.—Wolf, S.—Sznitman, R.: RetiNet: Automatic AMD Identification in OCT Volumetric Data. CoRR, 2016, doi: 10.48550/arXiv.1610.03628.

[15] Tian, Z.—Shen, C.—Chen, H.—He, T.: FCOS: A Simple and Strong Anchor-Free Object Detector. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 44, 2022, No. 4, pp. 1922–1933, doi: 10.1109/TPAMI.2020.3032166.

[16] Mahdavi, F.—Rajabi, R.: Drone Detection Using Convolutional Neural Networks. 2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), 2020, pp. 1–5, doi: 10.1109/ICSPIS51611.2020.9349620.

[17] Singha, S.—Aydin, B.: Automated Drone Detection Using YOLOv4. Drones, Vol. 5, 2021, No. 3, Art. No. 95, doi: 10.3390/drones5030095.

[18] Cheng, Q.—Li, X.—Zhu, B.—Shi, Y.—Xie, B.: Drone Detection Method Based on MobileViT and CA-PANet. Electronics, Vol. 12, 2023, No. 1, Art. No. 223, doi: 10.3390/electronics12010223.

[19] Gevorgyan, Z.: SIoU Loss: More Powerful Learning for Bounding Box Regression. CoRR, 2022, doi: 10.48550/arXiv.2205.12740.

[20] Aksoy, M. Ç.—Orak, A. S.—Özkan, H. M.—Selimoğlu, B.: Drone Dataset: Amateur Unmanned Air Vehicle Detection. 2019, doi: 10.17632/zcsj2g2m4c.4.

[21] Redmon, J.—Divvala, S.—Girshick, R.—Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[22] Redmon, J.—Farhadi, A.: YOLO9000: Better, Faster, Stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.

[23] Redmon, J.—Farhadi, A.: YOLOv3: An Incremental Improvement. CoRR, 2018, doi: 10.48550/arXiv.1804.02767.

[24] Hu, J.—Shen, L.—Sun, G.: Squeeze-and-Excitation Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[25] Woo, S.—Park, J.—Lee, J. Y.—Kweon, I. S.: CBAM: Convolutional Block Attention Module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11211, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[26] Hou, Q.—Zhou, D.—Feng, J.: Coordinate Attention for Efficient Mobile Network Design. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13708–13717, doi: 10.1109/CVPR46437.2021.01350.

[27] Cao, Z.—Yuan, R.: Real-Time Detection of Mango Based on Improved YOLOv4. Electronics, Vol. 11, 2022, No. 23, Art. No. 3853, doi: 10.3390/electronics11233853.

[28] Liu, S.—Huang, D.—Wang, Y.: Learning Spatial Fusion for Single-Shot Object Detection. CoRR, 2019, doi: 10.48550/arXiv.1911.09516.

[29] Yu, J.—Jiang, Y.—Wang, Z.—Cao, Z.—Huang, T.: UnitBox: An Advanced Object Detection Network. Proceedings of the 24th ACM International Conference on Multimedia (MM '16), 2016, pp. 516–520, doi: 10.1145/2964284.2967274.

[30] Rezatofighi, H.—Tsoi, N.—Gwak, J.—Sadeghian, A.—Reid, I.—Savarese, S.: Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 658–666, doi: 10.1109/CVPR.2019.00075.

[31] Zheng, Z.—Wang, P.—Liu, W.—Li, J.—Ye, R.—Ren, D.: Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2019, No. 7, pp. 12993–13000, doi: 10.1609/aaai.v34i07.6999.

[32] Lv, Y.—Ai, Z.—Chen, M.—Gong, X.—Wang, Y.—Lu, Z.: High-Resolution Drone Detection Based on Background Difference and SAG-YOLOv5s. Sensors, Vol. 22, 2022, No. 15, Art. No. 5825, doi: 10.3390/s22155825.

[33] Jocher, G.: YOLOv5 by Ultralytics. 2020, https://github.com/ultralytics/YOLOv5.

[34] Everingham, M.—Van Gool, L.—Williams, C. K. I.: The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, Vol. 88, 2010, No. 2, pp. 303–338, doi: 10.1007/s11263-009-0275-4.

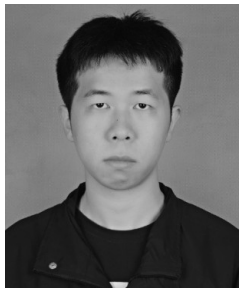
[35] Ge, Z.—Liu, S.—Wang, F.—Li, Z.—Sun, J.: YOLOX: Exceeding YOLO Series in 2021. CoRR, 2021, doi: 10.48550/arXiv.2107.08430.

[36] Li, X.—Wang, W.—Wu, L.—Chen, S.—Hu, X.—Li, J.—Tang, J.—Yang, J.: Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., Lin, H. (Eds.): Advances in Neural Information Processing Systems 33 (NeurIPS 2020). Curran Associates, Inc., 2020, pp. 21002–21012, https://proceedings.neurips.cc/paper_files/paper/2020/file/f0bda020d2470f2e74990a07a607ebd9-Paper.pdf.

[37] Wang, C. Y.—Bochkovskiy, A.—Liao, H. Y. M.: YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 7464–7475, doi: 10.1109/CVPR52729.2023.00721.

[38] Peng, Y.—Tu, X.—Yang, Q.—Li, R.: Lightweight UAV Detection Algorithm Based on Improved YOLOv5. Journal of Hunan University (Natural Science Edition), 2024, http://kns.cnki.net/kcms/detail/43.1061.n.20230901.1403.002.html (in Chinese).

Bing Su received his B.Sc. and Ph.D. degrees from the Nanjing University of Aeronautics and Astronautics (NUAA), China. He is currently Associate Professor with the Department of Computer Science, School of Information and Mathematics, Changzhou University. His current research interests include network security, wireless sensor networks, Internet of Things, routing protocols, and cloud computing.



Jie Zhang received his B.Sc. degree in communication engineering from the Zhejiang Shuren University in 2022 and is now pursuing his M.Sc. degree in computer science and technology at the Changzhou University. His main research interests include machine learning, deep learning and object detection.

Yifeng Lin received his B.Sc. and M.Sc. degrees in software engineering from the Jilin University, Jilin, China in 2005 and 2007, respectively, and his Ph.D. degree in computer application technology from the Jilin University, Jilin, China, in 2012. His current research interest is deep learning.