

RESEARCH ARTICLE

Learnable Cross-Scale Sparse Attention Guided Feature Fusion for UAV Object Detection

XIN ZUO¹, CHENHUI QI¹, YIFEI CHEN², JIFENG SHEN¹², HENG FAN¹³,
AND WANKOU YANG¹⁴, (Member, IEEE)

¹School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China

²School of Electronic and Informatics Engineering, Jiangsu University, Zhenjiang 212013, China

³Department of Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA

⁴School of Automation, Southeast University, Nanjing, Jiangsu 210096, China

Corresponding author: Jifeng Shen (shenjifeng@ujs.edu.cn)


This work was supported in part by NSF of China under Grant 61903164 and Grant 62276061; and in part by NSF of Jiangsu Province, China, under Grant BK20191427.

ABSTRACT Object detection in Unmanned Aerial Vehicle (UAVs) faces a significant challenge in computer vision. Traditional methods are difficult to model object appearance feature with large scale variations and viewpoint differences, when drones fly at different altitudes and capture images from diverse shooting angles. To address this issue, we propose a Learnable Cross-scale Sparse Attention (LCSA) guided feature fusion method to improve the performance of UAV object detection. Specifically, the LCSA feature fusion module enables each point in a feature map to aggregate discriminative information from a set of points with learnable offsets in neighbor feature maps. It enhances local discriminative features of the object by facilitating semantic information interaction across multiple feature maps. The LCSA can function as a novel neck method that complements the existing neck methods and is also transplantable to different object detection frameworks. Moreover, we also employ a scale-aware loss function to integrate the normalized Wasserstein distance with CIoU in order to improve the incompatibility of IoU for objects with large scale variance. Experimental results on the SeaDroneSeev2 and VisDrone2019-DET datasets show that the proposed method achieves superior performance. At a resolution of 640*640, our method achieves 81.9% AP50 and 47.4% AP on SeaDroneSeev2, surpassing baseline 4.9% and 4.8%, achieves state-of-the-art performance. Furthermore, our method outperforms baseline by 5% AP on VisDrone2019-DET. Code will be available at <https://github.com/qch777/LSACF>.

INDEX TERMS UAV object detection, cross-scale feature fusion, sparse attention.

I. INTRODUCTION

With the rapid evolution of unmanned aerial vehicle (UAV) technology, object detection in UAV [1], [2], [3] has emerged as a focal point in computer vision for aviation applications. In comparison to the generic object detection methods, object detection in UAV presents a multitude of distinct advantages. Firstly, the UAV images present unique advantages via their high flight altitude and broad field of view. Secondly, UAV possesses the capability to survey areas that are inaccessible to humans, thereby extending the scope

The associate editor coordinating the review of this manuscript and approving it for publication was Wenbing Zhao .

of monitoring and bestowing object detection tasks with a broader coverage capacity. These distinct perspectives enable the comprehensive acquisition of information pertaining to ground-level objects, thereby conferring an exceptional capability for holistic object perception.

However, object detection in UAV still confronts significant challenges. Firstly, owing to the substantial variation in visual field and object scale that arises from images captured at different altitudes necessitates a focused emphasis on the localization, extraction, and fusion of pivotal features. As shown in Fig. 1, images captured at lower altitudes show more details of the objects, highlighting characteristics such as shape, texture and color. Conversely, images captured



FIGURE 1. Sample images captured in UAV at different altitudes and different shooting angles on the SeaDroneSeeV2 [4] dataset. The images from left to right are captured from UAV with flying height of 5.4m, 41.1m and 226.9m respectively.



FIGURE 2. Some failure examples with FPN. (The red boxes in left indicate missed detections, while it represents false positives in right.)

at higher altitudes can obscure the appearance information of objects. Subsequently, the inherent multi-scale and wide field-of-view nature of images captured by UAV, which is imperative to efficaciously address the intricacies associated with multi-scale features.

In order to extract discriminative feature from raw images, the generic object detection frameworks usually employ the Backbone+FPN (Feature Pyramid Network) [5] architecture for feature extraction and fusion. Although FPN can fuse features from various scales [6], [7], it is still insufficient in capturing local discriminative features for extremely small or large objects. As shown in Fig. 2(b), false positives can appear when confronted with oversized objects in low-altitude UAV images, while missed detections are often happen due to the insufficient appearance feature of small objects in Fig. 2(a). Although the FPN can model multi-scale feature in hand-hold camera images, the significant feature discrepancy from large scale variance in UAV image still pose a great challenge to the performance of UAV object detection.

Although existing multi-scale feature fusion methods have significantly improved the detection of small objects, they often rely on manually designed cross-scale connections or fusion strategies, which are difficult to train and debug, and tend to be unstable. This is particularly problematic when dealing with large-scale variations in objects caused by changes in the altitude of drone photography. In recent years, attention mechanisms have become a core component for improving accuracy in the Transformer [8], [9], [10] and DETR [11], [12], [13] series. However, when attention spans across global spatial positions, it consumes a large amount of

memory and has extremely high computational complexity. Manual sparsification methods, such as local windows [14], extended windows [15], or axial stripes [9], do not reliably ensure the effectiveness of these strategies.

To address these issues, we propose a Learnable Cross-scale Sparse Attention guided (LCSA) feature fusion module, which enables each point in a feature map to aggregate discriminative information from a set of points with learnable offsets in neighbor feature maps. The positions of these points are dynamically learned during the model training, thereby effectuating semantic information exchange across scales. It is noteworthy that the proposed LCSA module minimizes the use of manually designed cross-scale connections or sparsification methods by employing learnable parameters that dynamically adjust to accommodate large-scale variations in objects. LCSA can function as a novel neck method, significantly outperforms the FPN, PANet and BiFPN methods. Besides, the LCSA is also generalizable, which can be plugged into different object detection frameworks, such as the one-stage, two-stage, anchor-based [16] or anchor-free [17] detectors. Furthermore, the LCSA module can work in conjunction with other SOTA neck methods, showing consistent performance improvement.

Traditional Intersection over Union (IoU) [18] is a key performance metric of object detection, but it is sensitive to the size of objects. For the small objects, even a one pixel shift can result in a considerable change in IoU value between the predicted result and ground truth. As a result, it significantly impacts IoU values, leading to a notable

performance drop for these small and medium size objects. Fortunately, CIoU [19] takes the location, size and shape of the object into consideration, which is robust to scale variation. Motivated by that, we employ a scale-aware CIoU loss function with Normalized Wasserstein Distance (NWD) [20] to deal with the scale variation problem. In contrast to utilizing CIoU or NWD independently, our approach models bounding boxes as two-dimensional Gaussian distributions and simultaneously accounts for the comparison of aspect ratios, areas, and center-point distances. In our experiment, it is found that these two components complement each other, leading to improved accuracy.

We have extensively evaluated the performance of our method on the SeaDroneSeeV2 dataset and also tested the generalization capabilities on the VisDrone2019-DET dataset. The main contributions of this work are outlined as follows:

- A learnable Cross-scale Sparse Attention (LCSA) guided feature fusion module is proposed to model the cross-scale feature interaction in the feature pyramid.
- The LCSA can function as a neck method, which is complementary to the existing neck methods and is also adaptive to different object detection frameworks.
- A scale-aware CIoU loss function with NWD is employed to improve the performance of small objects.
- The proposed method has achieved state-of-the-art performance in terms of mAP on the SeaDronesSeeV2 dataset, and it has also demonstrated significant improvements on the VisDrone2019-DET dataset.

II. RELATED WORK

A. OBJECT DETECTION

In object detection, two primary paradigms have emerged: one-stage detectors based on sliding-window mechanisms and two-stage detectors built on region proposal methods.

The YOLO [21], [22], [23], [24] detector family is a representative of one-stage object detector. YOLOv5 [25] incorporates CSPDarknet53 [26] as its backbone architecture, and leverages data augmentation techniques and adaptive training strategy which enhancing its overall performance. FCOS [27] is an anchor-free object detection method, which employs a fully convolutional network to densely predict the object center points, categories and bounding boxes from the feature map. Faster R-CNN [28] is one of the most classical two-stage object detectors. It introduces a Region Proposal Network (RPN) for the extraction of candidate regions and combines it with a classification/regression network. Cascade R-CNN [29], built upon the foundation of Faster R-CNN, which proposes a cascade architecture to refine bounding boxes progressively. Recently, DETR (Detection Transformer) [11] is proposed for object detection which is based on the Transformer [30] architecture. It employs self-attention mechanisms to fulfill object detection tasks by globally encoding and decoding the entire image, directly yielding both the class labels and bounding boxes of the objects. In order to solve the problem of large-scale

variation in target detection from a drone perspective, DHEM [31] leverages attention mechanisms and multi-scale feature fusion to enhance feature information. OGMN [32] improves detection accuracy by introducing auxiliary tasks. TridentNet [33] adopts a three-branch network structure and incorporates dilated convolutions to handle scale variations. TPH-yolov5 [34] integrates Transformer prediction heads and small-object detection heads. MFEFNet [35] employs global aggregation progressive adaptive feature fusion to effectively extract feature information. In this study, we employed an object detection model for UAVs, designing a network structure specifically to reduce the impact of large object scale variations observed from the UAV perspective.

B. ATTENTION MECHANISM

In object detection, attention mechanisms are employed to select regions of interest or enhance critical features, thereby enhancing the accuracy of target localization and classification.

Previous attention mechanisms [36], [37], [38], [39], [40], [41], [42], [43] have demonstrated the effectiveness of enhancing neural network feature within a framework. Nevertheless, they operate on single-layer feature maps and are limited in integrating global relationships. Fortunately, Transformers [14], [44], [45], [46] can learn dependencies within input sequences at a global scale, thus enabling comprehensive relationship modeling. However, this approach can lead to a substantial increase in the model's parameters and computational complexity, resulting in higher memory requirements. In this paper, we employed a sparse multi-scale attention mechanism, which selectively focuses on specific regions within the image, reducing computational overhead and enhancing the model's efficiency.

C. MULTI-SCALE FEATURE FUSION

Modern object detectors can be mainly categorized into the CNN-based and DETR-based models. CNN-based object detectors typically employ feature pyramids to fuse multi-scale features. Feature Pyramid Network (FPN) aggregates multi-scale features through a combination of top-down feature propagation and lateral connections [47], [48], [49], [50], [51]. Path Aggregation Network (PANet) [52] enhances FPN by introducing a bottom-up connection, facilitating the smoother passage of lower-level information to the uppermost levels. Building upon PANet, Bidirectional Feature Pyramid Network (BiFPN) [53] processes each bidirectional path and introduces a straightforward yet efficient mechanism for weighted feature fusion. CE-FPN [54] utilizes high-level semantic features and integrates an attention mechanism for selective feature fusion. FaPN [55] designs feature selection and feature alignment modules to improve fusion accuracy.

DETR employs attention mechanisms to facilitate multi-scale feature fusion by enabling the decoder of the Transformer to integrate information from various levels of the encoder and the global context. Deformable DETR builds upon DETR and introduces a deformable

attention mechanism resembling deformable convolutions. This adaptive mechanism accommodates changes in object shape and size, effectively aggregates features from different levels. RT-DETR [12] achieves multi-scale feature fusion through FPN and cross-scale attention mechanism. MFDS-DETR [13] introduces FPN and multi-scale decoder for multi-scale feature extraction and fusion.

In this study, the proposed LCSA feature fusion module enhances the discrimination of local object features through cross-scale semantic information interaction.

D. LOSS FUNCTION

The Intersection over Union (IoU) [18] is commonly used for performance evaluation in object detection.

It measures the overlap between the predicted region and the ground truth box by computing the ratio of their intersection to their union. However, IoU focuses solely on overlap and doesn't consider size differences. To address this issue, the Generalized IoU (GIoU) [56] is introduced, which takes into account the relative positions and sizes of bounding boxes simultaneously. However, GIoU would then be equivalent to IoU when the predicted box completely encloses the ground truth label. To mitigate this, the Complete Intersection over Union (CIoU) [19] is introduced. CIoU builds upon GIoU by incorporating the aspect ratio and center point distance of bounding boxes, reducing the impact on IoU due to the size of the bounding boxes. The Wasserstein distance is a distance metric employed to quantify the similarity between two probability distributions. It takes into account both the global structural aspects and the local correspondence between the distributions. The Normalized Wasserstein Distance introduces a normalization procedure that maps its range onto the interval between 0 and 1. This adaptation addresses potential range discrepancies in the original Wasserstein distance when comparing distributions of varying scales or sizes. In this study, we employed a scale-aware CIoU loss function with normalized Wasserstein distance to improve detection accuracy. By establishing a balance hyper-parameter, we enhance the localization and evaluation methodologies for objects.

III. METHOD

A. OVERVIEW

The UAV object detection framework is shown in Fig. 3, which mainly composed of three main components: backbone, neck, and head. The ResNet and CSPDarknet53 are usually used as backbones, while PANet and BiFPN are good candidates of improved neck methods which also serve as the input to the detection head. The LCSA module is placed between the backbone and neck, which can also function as a novel neck method.

B. LEARNABLE CROSS-SCALE SPARSE ATTENTION GUIDED FEATURE FUSION

Comparing to the convolutional neural networks, Transformers excels in global feature modeling and fusion, enabling

effective capture of contextual information between feature maps. However, this approach significantly increases model complexity and memory consumption. To address this issue, we propose a LCSA guided feature fusion module inspired by deformable attention mechanisms, as shown in Fig. 4.

Previous studies usually apply self-attention on a single feature map or replace traditional convolutions with deformable convolutions [57] to enhance feature. However, the proposed LCSA module operates on multi-scale feature maps, which selects a set of points with learnable offsets. These selected points are capable of learning contextual information from the anchor points in neighbor feature maps during the model training process. This approach dynamically adjusts these parameters, thereby mitigating information loss and improving the discriminative features of local objects.

The details of the LCSA module are shown in Fig. 4. Given a set of feature maps with L scales, each feature map is represented as $X^l \in \mathbb{R}^{N_l \times C}$, where l is the scale index, N_l is the number of feature points and C is the number of channels. Assume $p^l(x, y)$ is a feature vector from anchor point (x, y) in the l -th feature map X^l . Then, the feature vector $p^l(x, y)$ is linearly transformed to obtain a set of offset vectors¹ $\Delta p^l(x, y) \in \mathbb{R}^{2K}$ with function f , which is formulated in Eq. 1.

$$\Delta p^l = f(p^l) = w_1^l * p^l \quad (1)$$

where w_1^l is a linear transformation matrix, K is the number of points sampled in each feature map (e.g. $K = 3$ in Fig. 4), $k = 1, 2, \dots, K$. For each offset point, the corresponding feature $X^l(p^l + \Delta p^l)$ can also be obtained from the input feature map X^l . Since each anchor point can undergo a linear transformation to obtain multiple offset vectors Δp_k^l , forming a set of offset points, each of these offset points can be sampled to acquire the corresponding feature $X^l(p^l + \Delta p_k^l)$, $k \in [1, K]$.

The corresponding anchor points in the $(l-1)$ -th and $(l+1)$ -th feature maps are denoted as $p^{l-1}(x/2, y/2)$ and $p^{l+1}(2x, 2y)$ respectively. Due to the effect of fractional coordinates in $x/2$ and $y/2$, bi-linear interpolation are employed to handle such cases.

Simultaneously, each offset point is associated with an attention weight $A_k^l \in [0, 1]$, normalized by $\sum_{k=1}^K A_k^l = 1$. The attention weight A_k^l is obtained through two linear transformations applied to the offset point, along with a set of learnable parameters W_p , as shown in Eq. 2.

$$A_k^l = W_p \cdot (w_3^l \cdot (w_2^l \cdot p^l)) \quad (2)$$

where w_2^l and w_3^l are linear transformation matrices. The output Y^l of the l -th layer feature map is obtained by computing a weighted sum of the features using the

¹In the following text, $p^l(x, y)$ is abbreviated as p^l for clarity.

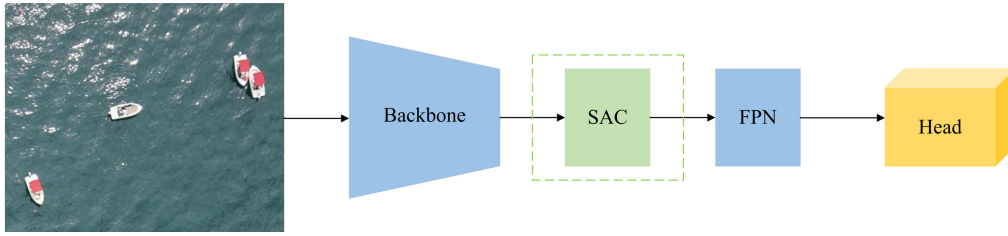


FIGURE 3. Overview of our proposed UAV Object Detection with LCSA module. Our proposed LCSA module works after backbone for sparse feature fusion, neck module aggregates multi-scale feature maps and head module outputs the final detection results.

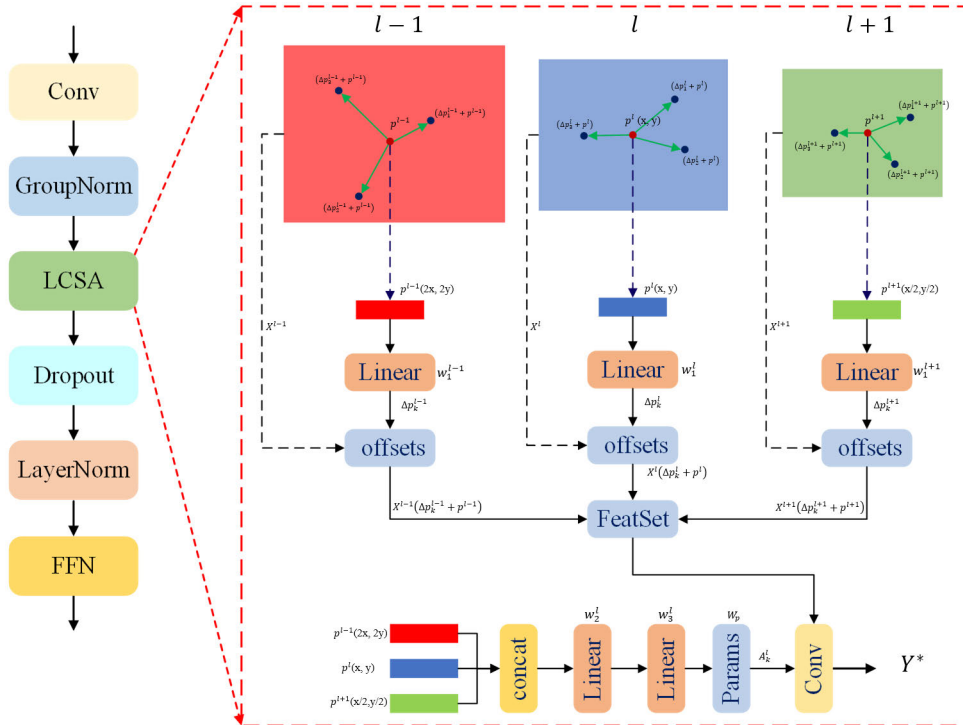


FIGURE 4. Illustration of the proposed LCSA module. (On the left, the proposed LCSA module comprised of Conv, GroupNorm, Learnable Cross-scale Sparse Attention, Dropout, LayerNorm and FFN module. The right side illustrates the details of Learnable Cross-scale Sparse Attention.

corresponding attention weights, which is shown in Eq. 3.

$$Y^l = \sum_{k=1}^K A_k^l \cdot X^l(\Delta p_k^l + p^l) \quad (3)$$

Due to the inherent advantages of the multi-head mechanism in improving both training and inference efficiency, as well as its effectiveness in extracting rich information, we can extend our approach to incorporate a multi-head mechanism, as shown in Eq. 4.

$$Y_h^l = \sum_{k=1}^K A_{h,k}^l \cdot X^l(\Delta p_{h,k}^l + p^l) \quad (4)$$

where $\Delta p_{h,k}^l$ represents the k -th offset in the h -th head of the l -th layer feature map, and similarly, $A_{h,k}^l$ denotes the attention weight for the k -th offset point in the h -th head of

the l -th layer feature map, where $A_{h,k}^l \in [0, 1]$. The attention weights are normalized through $\sum_{l=1}^L \sum_{k=1}^K A_{h,k}^l = 1$. Finally, the outputs Y_h^l from all scales and heads are fused to obtain the ultimate output feature Y^* , as shown in Eq. 5.

$$Y^* = \sum_{l=1}^L \sum_{h=1}^H Y_h^l \cdot W_{O_{h,l}} \quad (5)$$

where $W_{O_{h,l}}$ are learnable parameters, H is the number of heads.

As shown in Fig. 5, The input of LCSA module are the multi-scale feature maps which are extracted from the feature maps output by the P2, P3, P4, and P5 stages of the model's Backbone. Through a 1×1 convolution, the channel dimensions of the feature maps are standardized to 256. It is worth noting that the P3 stage originally outputs feature maps with 256 channels, so no 1×1 convolution is

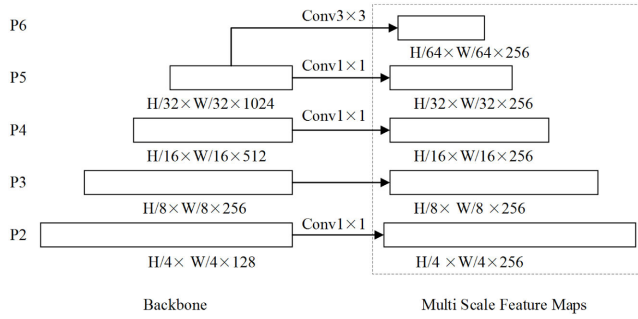


FIGURE 5. Constructing multi-scale feature maps for LCSA module.

required for the P3 layer. The feature maps of the P6 stage are obtained from the P5 stage through a 3×3 convolution. After achieving channel alignment, each point in any layer's feature map can be corresponded to a point in the feature maps of other layers through positional encoding. Consequently, each layer's feature map can utilize the LCSA module to fuse semantic information from multiple layers of feature maps.

C. LOSS FUNCTION

Traditional IOU primarily relies on the computation of the intersection and union of the ground truth and predicted bounding boxes, neglecting any consideration of the bounding box sizes. CIoU, building upon IOU, takes into account the aspect ratio and distance between the center points of the bounding boxes, reducing the impact of bounding box size on IoU. However, when dealing with large-scale variations, CIoU may not effectively highlight prediction errors, especially in the case of small objects, where it might struggle to accurately reflect positional errors. In contrast, Normalized Wasserstein Distance can map bounding boxes of different scales to a consistent range, thereby alleviating the influence of object scale variations. Therefore, we employ a scale-aware CIoU loss function incorporating Normalized Wasserstein Distance to more effectively capture the position and size information of bounding boxes.

The second-order Wasserstein distance between two-dimensional Gaussian distributions $\mu_1 = \mathcal{N}(\mathbf{m}_1, \mathbf{b}_1)$ and $\mu_2 = \mathcal{N}(\mathbf{m}_2, \mathbf{b}_2)$ can be formally defined as Eq. 6.

$$W_2^2(\mu_1, \mu_2) = \|\mathbf{m}_1 - \mathbf{m}_2\|_F^2 + \text{Tr} \left(\Sigma_1 + \Sigma_2 - 2 \left(\Sigma_2^{\frac{1}{2}} \Sigma_1^{\frac{1}{2}} \right)^{\frac{1}{2}} \right) \quad (6)$$

where m_1 and m_2 are the center points of μ_1 and μ_2 , \mathbf{b}_1 and \mathbf{b}_2 are the covariance matrices of μ_1 and μ_2 , $\|\cdot\|_F$ is the Frobenius norm. For Gaussian distributions \mathcal{N}_a and \mathcal{N}_b are modeled by bounding box $A = (cx_a, cy_a, w_a, h_a)$ and bounding box $B = (cx_b, cy_b, w_b, h_b)$, so the above formula

can be simplified as Eq. 7.

$$W_2^2(\mathcal{N}_a, \mathcal{N}_b) = \left\| \left(\begin{bmatrix} cx_a, cy_a, \frac{w_a}{2}, \frac{h_a}{2} \end{bmatrix}^T, \begin{bmatrix} cx_b, cy_b, \frac{w_b}{2}, \frac{h_b}{2} \end{bmatrix}^T \right) \right\|_2^2 \quad (7)$$

Nevertheless, the distance $W_2^2(\mathcal{N}_a, \mathcal{N}_b)$ is merely a metric and cannot be directly employed as a similarity measure. Consequently, it is necessary to normalize it in order to produce a measure that can be effectively utilized, as seen in Eq. 8.

$$L_{\text{NWD}} = \text{NWD}(N_a, N_b) = \exp \left(-\frac{\sqrt{W_2^2(N_a, N_b)}}{c} \right) \quad (8)$$

Let $\mathbf{b}, \mathbf{b}^{\text{gt}}$ denote the center points of the predicted frame and the real frame, respectively. The variable ρ indicates the Euclidean distance between these two center points, while c represents the diagonal distance between the predicted frame and the minimum closure area of the real frame. The loss calculation formula of CIoU is defined as Eq. 9.

$$L_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{\text{gt}})}{c^2} + \alpha v \quad (9)$$

IoU is expressed as the ratio of the intersection and union of the predicted bounding box and the true bounding box. α is the weight parameter, which takes into account the aspect ratio details of the predicted box and the real box, and its expression is Eq. 10. The variable v is employed to quantify the degree of consistency in aspect ratio, and its mathematical representation is Eq. 11.

$$\alpha = \frac{v}{(1 - \text{IoU}) + v} \quad (10)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{W^{\text{gt}}}{h^{\text{gt}}} - \arctan \frac{W}{h} \right)^2 \quad (11)$$

$$L_{\text{CIoU+NWD}} = (1 - \beta) * L_{\text{NWD}} + \beta * L_{\text{CIoU}} \quad (12)$$

Finally, the reconstructed loss function is calculated as Eq. 12. Where β is a hyperparameter that controls the threshold, which combines two loss functions to control the weight ratio at the same time.

IV. EXPERIMENTS

A. DATASETS AND EVALUATION METRICS

1) SeaDronesSee OBJECT DETECTIONV2 DATASET

SeaDronesSee object detectionv2 is a large-scaled sea-based visual object detection benchmark. There are a total of 8,930 training photos, 1,547 validation images, and 3,750 test images. In order to enhance the precision of annotation, we use the sanitized annotations for training and validation.

2) VisDrone2019-DET DATASET

VisDrone2019-DET is a large-scale UAV object detection benchmark dataset. It encompasses a diverse range of scenarios and categories, and its primary purpose is to assess

the effectiveness of object detection algorithms when applied to UAV imagery. There are a total of 6,471 training images, 548 validation images and 1,610 test images.

3) EVALUATION METRICS

In object detection, average precision (AP) is widely used for model evaluation. AP is the area under the precision-recall curve calculated at different thresholds. The calculation method is as Eq. 13 Precision is defined as the ratio of the number of true positives (TP) to the number of samples predicted as positive:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (13)$$

where FP represents the number of false positives. Recall is defined as the ratio of TP to the number of samples that are actually positive, The formula is Eq. 14

$$\text{Recall} = \frac{TP}{TP + FN} \quad (14)$$

For different confidence thresholds, calculate the precision and recall values. By interpolating these points, calculate the area under the precision-recall curve. The formula is Eq. 15

$$AP = \int_0^1 p(r) dr \quad (15)$$

where $p(r)$ is the precision at a given recall r .

AP50 is determined by calculating the precision when the IOU is equal to or greater than 0.5. AP75 is computed by considering instances where the IOU is equal to or greater than 0.75. Following the coco evaluation metric, the metric AP_S quantifies the mean precision of small-sized objects, defined as bounding boxes with an area of 32 × 32 pixels or less. Similarly, AP_M calculates the average precision for medium-sized objects, which are bounding boxes with an area between 32 × 32 pixels and 96 × 96 pixels. Lastly, AP_L evaluates the average precision for large-scale objects, which are bounding boxes larger than 96 × 96 pixels.

B. IMPLEMENTATION DETAILS

Our method is implemented with Pytorch 1.12 on an Ubuntu 20.04 server equipped with a single GPU (NVIDIA GTX-3080TI) and a single CPU (Intel i9-12900HX). The SGD optimizer is employed in this study with an initial learning rate of 0.01. YOLOv5 and YOLOv8 models are trained for a total of 100 epochs, while Faster R-CNN and FCOS models are trained for 50 epochs. The momentum parameter is set at 0.937. Additionally, a preheating phase of 3 epochs is implemented, with an initial momentum of 0.8. The initial bias is set to 0.1, and the process of upsampling is accomplished by the utilization of bilinear interpolation. Both the training and testing images have a size of 640 × 640 pixels. In this study, we present a LCSA feature fusion module, defaults to using 8 attention heads and 4 sample points. The Pytorch framework is employed for its implementation, and a CUDA version is also provided to enhance the

TABLE 1. Comparison on the SeaDronesSeeV2 dataset.

Method	Backbone	AP	AP50	AP75
Cascade R-CNN [29]	R50	39.6	69.2	33.0
Faster R-CNN [28]	R50	31.3	61.7	28.9
FCOS [27]	R50	42.5	71.6	47.2
Deformable DETR [61]	R50	39.2	72.7	36.4
YOLOv5 [25]	CSPDarkNet	42.6	77.0	40.9
TPH-YOLOv5 [34]	CSPDarkNet	46.4	81.1	47.4
YOLOv8	CSPDarkNet	41.7	68.0	43.4
RT-DETR [12]	R50	40.7	77.5	43.2
MFDS-DETR [13]	R50	40.6	79	43.3
Ours	CSPDarkNet	47.4	81.9	46.7

TABLE 2. Comparison on the VisDrone2019-DET dataset.

Method	Backbone	AP	AP50	AP75
FCOS [27]	R50	6.5	13.5	6.3
Faster R-CNN [28]	R50	19.1	29.1	22.0
Deformable DETR [61]	R50	19.6	35.1	19.1
Cascade R-CNN [29]	R50	23.2	39.9	27.4
TPH-YOLOv5 [34]	CSPDarkNet	23.6	41.2	23.8
YOLOv5 [25]	CSPDarkNet	24.4	41.1	24.6
RetinaNet [5]	R50	24.3	44.3	18.7
Libra R-CNN [62]	-	24.3	41.2	24.9
HawkNet [63]	-	25.6	44.3	25.8
VFNet [64]	-	25.9	42.1	27.0
DetectoRS [65]	R50	26.8	43.2	28.0
DMNet [66]	-	28.2	47.6	28.9
ClusDet [67]	R50	26.7	50.6	24.7
RRNet [68]	-	29.1	55.8	27.2
Ours	R50	29.4	48.6	30.6

training speed. Given the huge data volume and plentiful samples available in the VisDrone2019-DET training set and the SeaDronesSeeV2 training set, no data augmentation approach was employed in this experiment. Due to multiple iterations and optimizations, YOLOv5 has been applied and validated in numerous real-world projects [58], [59], [60]. It achieves a good balance between speed, accuracy, and model size, facilitating rapid deployment across different platforms. This study adopts YOLOv5 as the baseline model, given its current status as the most efficient and convenient one-stage detector in the YOLO series. The network's depth and width are both 1.00, and the number of parameters is comparable to that of Faster R-CNN, FCOS, and YOLOv8. The Faster R-CNN algorithm utilizes the implementation code provided by the official torchvision library in Pytorch, whereas the FCOS algorithm makes use of the mmdetection library.

C. COMPARISON WITH STATE-OF-THE-ART METHODS

Tab. 1 and Tab. 2 presents a comparison between our method and the state-of-the-art object detection methods on the SeaDronesSeeV2 and VisDrone2019-DET dataset. We have compared with current state-of-the-art detectors, including one-stage, two-stage, anchor-free and anchor-based methods.

a: SeaDronesSeeV2 DATASET

It is evident that our method outperforms all existing methods, achieving state-of-the-art performance. Specifically, our

approach achieves an AP of 47.4%, significantly surpassing other detectors. Notably, our proposed method not only improves AP by 4.9% over the baseline model, but also achieves remarkable improvements in AP50 and AP75 metrics, with increases of 4.9% and 5.8% respectively.

b: VisDrone2019-DET DATASET

It is evident that our approach surpasses the current state-of-the-art methods in terms of both AP and AP75. Despite a 7.2% difference in AP50 compared to RRNet, our method achieves a remarkable improvement of 5% in AP, 7.5% in AP50, and 6% in AP75 when compared to the baseline. This demonstrating the strong generalization capability of our proposed LCSA module.

We conducted an analysis of the factors contributing to the lower performance of our method compared to ClusDet and RRnet on AP50. Our approach relies on capturing effective information by fusing features at learnable points corresponding to different scales on feature maps. However, in the VisDrone2019 dataset, the presence of numerous mutually occluded objects makes it challenging to obtain valuable information, thereby limiting the performance on AP50.

In contrast, ClusDet focuses on handling potentially occluded clustered regions using specialized methods. RRNet employs adaptive re-sampling techniques during data augmentation, introducing more occluded targets in the process, leading to superior performance on AP50. It is noteworthy that under more stringent evaluation metrics such as AP75 and AP, our method outperforms both ClusDet and RRnet.

D. ABLATION STUDIES

1) THE EFFECT OF EACH MODULE

We analyzed the effectiveness of each proposed module on the SeaDronesSeeV2 validation dataset, as presented in Tab. 3.

a: IMPACT OF LCSA

The addition of the LCSA module significantly improved all metrics, with the most notable increase observed in AP75, reaching 7.1%. However, FLOPs increase from 107G to 170.0G, and the parameter count increased from 46.5M to 50.0M. Despite the increased computational load and complexity, we were still able to achieve a real-time speed of 76.3 FPS.

b: IMPACT OF RECONSTRUCTING THE LOSS FUNCTION

After the reconstruction of the loss function, the FLOPs and parameter number remain relatively stable. All metrics exhibited a significant improvement, with AP75, AP_S, and AP_M showing the most noticeable enhancements, increasing by 3.7%, 1.4%, and 1.6%, respectively.

c: IMPACT OF OUR METHOD

Our method incorporates the LCSA module and the reconstruction of the loss function after reconstruction. Compared to the model with LCSA added, although there is a slight

decrease in AP75 and AP_S, AP, AP50, AP_M and AP_L have improved by 0.1%, 0.6%, 0.4%, and 0.5%, respectively.

2) THE EFFECT OF LCSA MODULE UNDER DIFFERENT DETECTION FRAMEWORKS

To validate the generality and effectiveness of the proposed LCSA module, we seamlessly integrated it into representative Two-stage, one-stage and Anchor-Free algorithms, dividing them into four experimental groups. The experimental results are shown in Tab. 4.

Experiments on the SeaDroneSeeV2 dataset have shown that the use of our proposed LCSA module on Faster R-CNN, YOLOv5, FCOS, and YOLOv8 models yields improved performance over baseline models. Notably, there is a significant enhancement in AP50, with improvements of 1.6%, 4.3%, 3.1%, and 1.0% for these respective models. Furthermore, there are varying degrees of improvement in AP and AP75 metrics. Simultaneously, it is worth noting that both parameter count and computational complexity increase noticeably, resulting in a decrease in real-time model speed. As a result, we conclude that the LCSA module is applicable to various of object detection models utilizing the Backbone+FPN structure, all evaluation metrics consistently demonstrate its effectiveness. However, it comes with significant increases in both model parameters and computational demands.

3) THE EFFECT OF REPLACING FPN WITH LCSA MODULE

In this section, we conducted experiments by removing the model's neck and stacking 1, 3, 6, and 9 layers of LCSA modules, as shown in Tab. 5. When LCSA modules were stacked to 6 layers, the optimal performance was achieved. Specifically, AP reached 46.5%, AP50 reached 80.4%, and AP75 reached 46.7%. Compared to the baseline model without a neck, our proposed method exhibited improvements of 6.5%, 6.7%, and 5.8% in AP, AP50, and AP75, respectively. Furthermore, these results surpassed the performance of the baseline model. It is worth noting that despite a significant increase in FLOPs, the number of parameters remained comparable to introducing LCSA into the baseline model. Although the speed decreased to 51 FPS, it is still sufficient to meet practical application requirements.

As shown in Fig. 6, we visualized the feature maps of different feature layers within the LCSA module. Comparing with the baseline, it is evident that the feature maps in the P3 layer predominantly focus on small objects, while the feature maps in the P4 and P5 layers cater to medium to large objects, albeit with limited effect. Since the LCSA module learns from multiple feature layers, it has already aggregated most of the fundamental features. Therefore, the P3 layer alone can effectively recognize almost all objects, while P4 and P5 are used to enhance object detection, thereby improving accuracy.

With an increase in the number of stacked LCSA modules, there is a gradual increase in noise points, especially when using 9 layers, which leads to a performance decrease due to excessive noise. Overall, the LCSA module accomplishes

TABLE 3. Ablation experiments on the SeaDronesSeeV2 dataset.

	LCSA	CIoU+NWD	AP	AP50	AP75	AP _S	AP _M	AP _L	Param(M)	GFLOPs	Speed (FPS)
YOLOv5			42.6	77.0	40.9	32.5	44.0	59.3	46.5	109.0	121.9
	✓		47.3	81.3	48.0	39.8	47.7	62.4	50.0	170.0	76.3
Ours		✓	43.8	78.6	44.6	33.9	45.6	59.8	109.0	109.1	121.8
	✓	✓	47.4	81.9	46.7	39.5	48.2	62.9	50.0	170.0	76.3

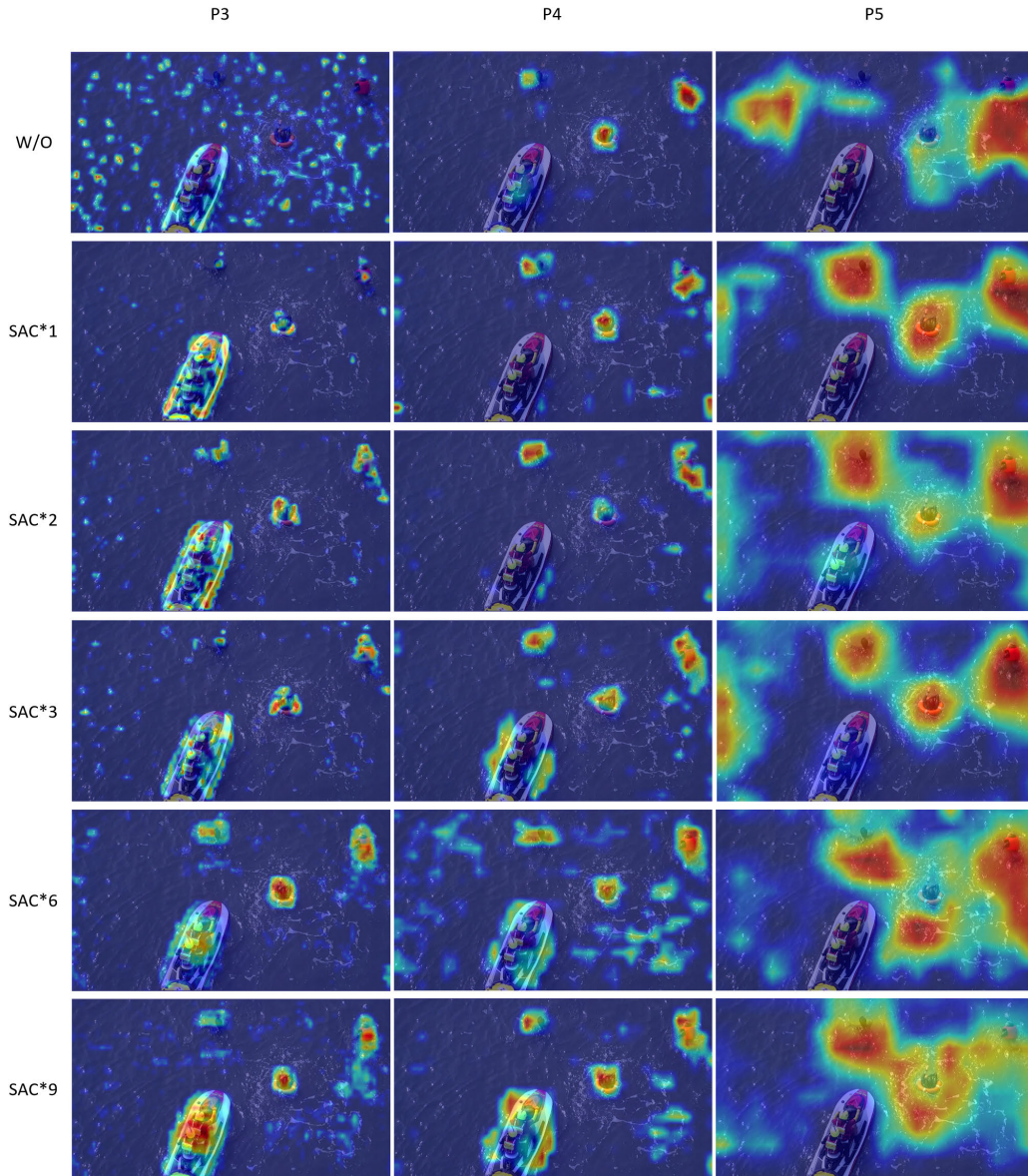


FIGURE 6. Visualization results of LCSA module with different stacking number under different feature layers. The first row represents the baseline results, while the others are heatmaps for different stacking number of the LCSA module. Each column represents a different feature layer.

semantic interactions between cross-scale features, enhancing the local discriminative features of objects.

4) THE EFFECT OF SERIAL USE OF LCSA MODULE AND FPN
 In this section, we integrated the LCSA module in conjunction with FPN and conducted stacking experiments,

as depicted in Tab. 6. Following the concatenation of LCSA and FPN, there were notable advancements in the AP, AP50, and AP75 metrics, with respective values of 48.3%, 80.8%, and 49.8%. This represents an improvement of 5.7% in AP, 3.8% in AP50, and 8.9% in AP75 over the baseline model. However, this improvement came at the cost of a reduction in

TABLE 4. Comparison of experimental results of adding LCSA module to different SOTA models on the SeaDronesSeeV2 dataset. TS, OS and AF denotes Two-Stage, One-Stage and Anchor-Free respectively.

Method	Type	Backbone	AP	AP50	AP75	Param(M)	GFLOPs	Speed(FPS)
Faster-RCNN	TS	Resnet50	31.3	61.7	28.9	41.3	32.2	26
+LCSA	TS	Resnet50	32.4	63.3	29.0	48.7	40.9	18
YOLOv5	OS	CSPDarkNet	42.6	77.0	40.9	46.5	109.0	121.9
+LCSA	OS	CSPDarkNet	47.3	81.3	48.0	50.0	170.0	76.3
FCOS	AF	Resnet50	42.5	71.6	47.2	32.1	47.7	19
+LCSA	AF	Resnet50	44.0	74.7	48.5	39.3	55.4	11
YOLOv8	AF	CSPDarkNet	41.7	68.0	43.4	43.6	164.8	98
+LCSA	AF	CSPDarkNet	42.5	69.0	44.4	46.2	225.0	52.1

TABLE 5. Effect of replacing FPN with LCSA module on experimental results.* represents the number of stacks.

Methods	AP	AP50	AP75	Param(M)	GFLOPs	Speed(FPS)
YOLOv5 (w/o Neck)	40.7	73.7	40.9	24.0	67.5	192
+LCSA*1	41.8	76.1	38.8	30.6	114.6	99.0
+LCSA*3	42.0	78.5	39.0	38.2	166.4	74.1
+LCSA*6	46.5	80.4	46.7	49.1	219.2	51.0
+LCSA*9	45.8	79.9	46.1	60.4	271.6	39.8

TABLE 6. Effect of LCSA module stacking times on experimental results. * represents the number of stacks.

Methods	AP	AP50	AP75	Param(M)	GFLOPs	Speed(FPS)
YOLOv5(w/ Neck)	42.6	77.0	40.9	46.5	109.0	121.9
+LCSA*1	47.3	81.3	48.0	50.0	170.0	76.3
+LCSA*2	48.0	81.0	48.7	53.8	187.4	68.1
+LCSA*3	48.5	81.1	48.6	57.4	204.9	60.2

speed from 121.9 FPS to 75.7 FPS, an increase in parameter from 46.5M to 50.0M and an increase in FLOPs from 109.0 G to 170.0 G.

As the stacking depth of LCSA modules increased, in comparison to a single LCSA layer, there was a discernible boost in AP by 0.7% and 1.2%, accompanied by a marginal decrease in AP50 and a notable enhancement in AP75 by 0.7% and 0.6%. Simultaneously, parameter count and computational complexity substantially increased, resulting in a decrease in real-time processing speed. Analyzing the visualizations of feature maps in the second to fourth rows of Fig. 6, the most probable reason for the decrease in AP50 appears to be the influence of background noise. Conversely, the repeated stacking of LCSA modules enhances the model's ability to precisely locate objects, consequently elevating AP and AP75. In summary, as the stacking depth increases, it leads to an overall enhancement in the model's performance, while the escalation in parameter count and computational complexity results in a reduction in processing speed.

5) COMPARISON OF LCSA MODULE WITH DIFFERENT FPN STRUCTURES

In the current research, various FPN (Feature Pyramid Network) structures yield different experimental results. Therefore, in this section, we conducted comparative analysis experiments using different FPN methods in conjunction with the LCSA (Semantic Adaptive Convolution) module we proposed, as shown in Tab. 7. Specifically, among the traditional FPN methods, BiFPN performs the best.

TABLE 7. Comparison of different neck methods* represents the number of stacks.

Methods	AP	AP50	AP75	Param(M)	GFLOPs	Speed(FPS)
YOLOv5(w/o Neck)	40.7	73.7	40.9	24.0	67.5	192
+FPN	41.3	77.5	38.7	41.1	99.8	133.3
+PANet	42.6	77	40.9	46.5	109.0	121.9
+BiFPN	43.1	78.2	41.6	46.8	109.8	109.9
+LCSA*6	46.5	80.4	46.7	49.1	219.2	51.0

Compared to the baseline model without a neck, it achieves a 2.4% improvement in AP (Average Precision), a 4.5% improvement in AP50, and a 0.7% improvement in AP75. The use of LCSA modules stacked six times produces the best results, with a 3.4% improvement in AP, a 2.2% improvement in AP50, and a 5.1% improvement in AP75 compared to BiFPN. However, it comes at the cost of a 2.3M increase in parameters and a significant rise in computational complexity by 109.4 GFLOPs.

In summary, the LCSA module enhances model performance by promoting semantic interactions between cross-scale features, allowing the model to focus more effectively on critical information extraction. Furthermore, stacking LCSA modules can surpass the performance of traditional neck methods, but it comes with the trade-off of increased parameters and computational complexity.

6) ABLATIONS FOR THE NUMBER OF SAMPLING POINTS IN LCSA

In this section, we conducted experimental analyses on the LCSA module with varying numbers of sampling points, and the experimental results are presented in Tab. 8. When the number of sampling points is set to 1, the LSA module degenerates into deformable convolution, resulting in a slight performance improvement. Specifically, AP, AP50, and AP75 increased by 0.5%, 0.6%, and 0.3%, respectively. When we increase the number of sampling points to 2 or more, a significant performance improvement is observed. The best performance is achieved when the number of sampling points is set to 6, with an AP of 48.8%. For 4 sampling points, AP50 performance reaches 81.3%, and with 2 sampling points, AP75 reaches 50.8%. It is worth noting that this performance enhancement is achieved with almost no increase in model parameters and only a slight increase in computational complexity. Further increases in the number of sampling points result in some performance fluctuations. Therefore, we conducted feature map analysis for different numbers of sampling points, which correspond to different channels, as shown in Fig. 7. When the number of sampling points is 1, there is significant background noise, leading to poorer model performance. As the number of sampling points increases to 2, background noise decreases, and AP75 reaches its peak. As the number of sampling points continues to increase, there is a gradual appearance of slight background noise, resulting in some experimental fluctuations. When the number of sampling points reaches 10, background noise interference becomes prominent, leading to a decrease in experimental results.

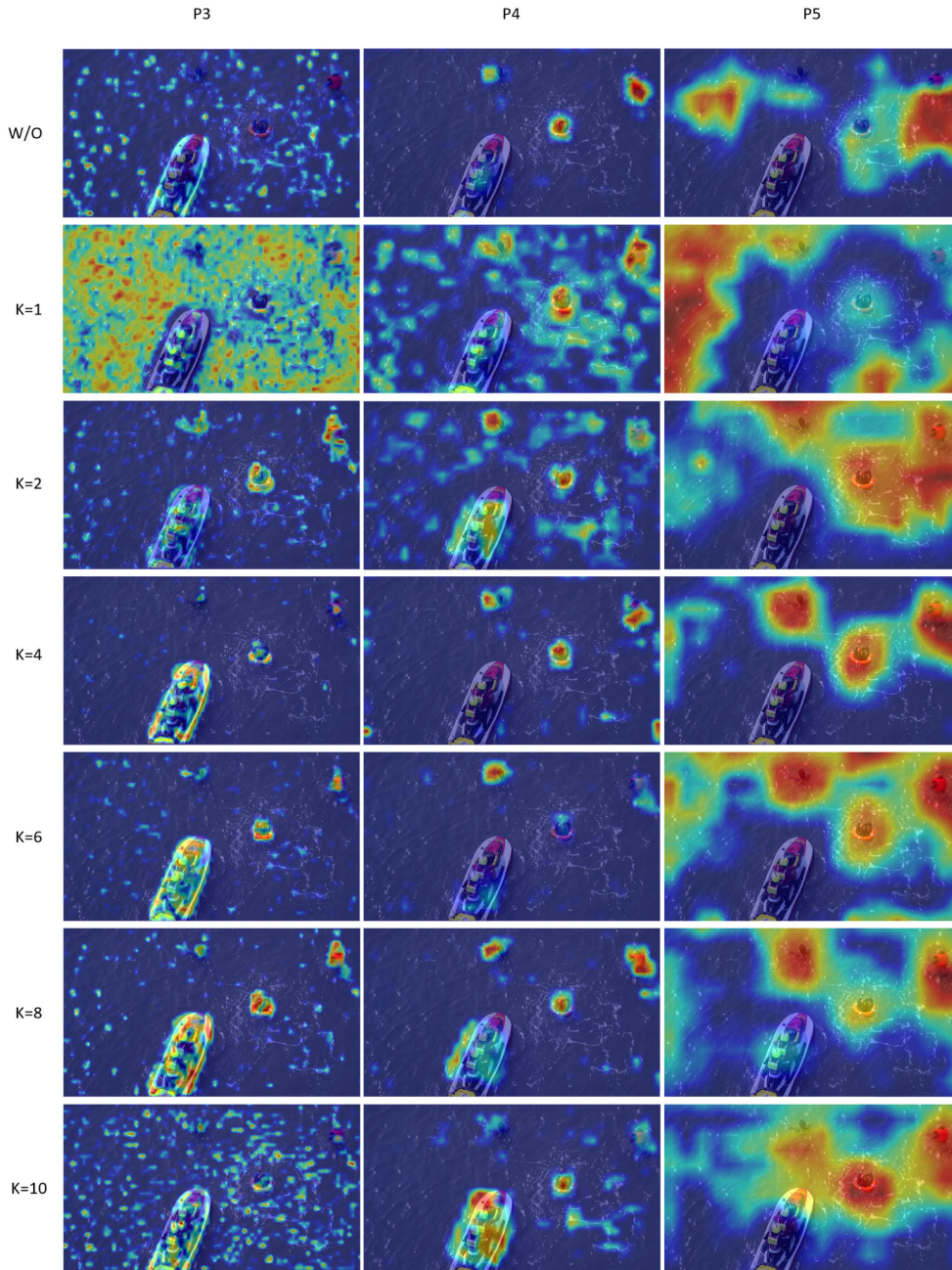


FIGURE 7. Visualization heatmap of LCSA module with different numbers of sampling points. The top row represents the baseline results, while the others depict feature maps after one LCSA module. Each column represents different feature layer.

TABLE 8. Ablations for LSA in LCSA. K is the number of sampling points.* represents the number of stacks.

Method	K	AP	AP50	AP75	Param(M)	GFLOPs	Speed(FPS)
YOLOv5(w/Neck)	-	42.6	77.0	40.9	46.5	109.0	121.9
	1	43.1	77.6	41.2	50.0	163.6	81.9
	2	48.7	81.0	50.8	50.0	165.7	80.0
+LCSA*1	4	47.3	81.3	48.0	50.0	170.0	76.3
	6	48.8	81.2	50.7	50.2	174.2	73.5
	8	48.1	81.3	48.3	50.2	178.4	70.9
	10	48.3	80.4	49.9	50.3	182.7	68.0

In summary, as the number of sampling points increases, the increase in model parameters is not significant, but there

is a noticeable increase in FLOPs, and the speed also slightly decreases. The model achieves good results when the number of sampling points is between 2 and 8, but there is some degree of fluctuation in the experimental outcomes.

7) THE EFFECT OF THE HYPER-PARAMETER β

In this section, we compared the impact of different β values on the experimental results and other IoU improvement methods, as shown in Tab. 9. When the β value is set to 0.5, the best performance is obtained. Compared to the baseline,

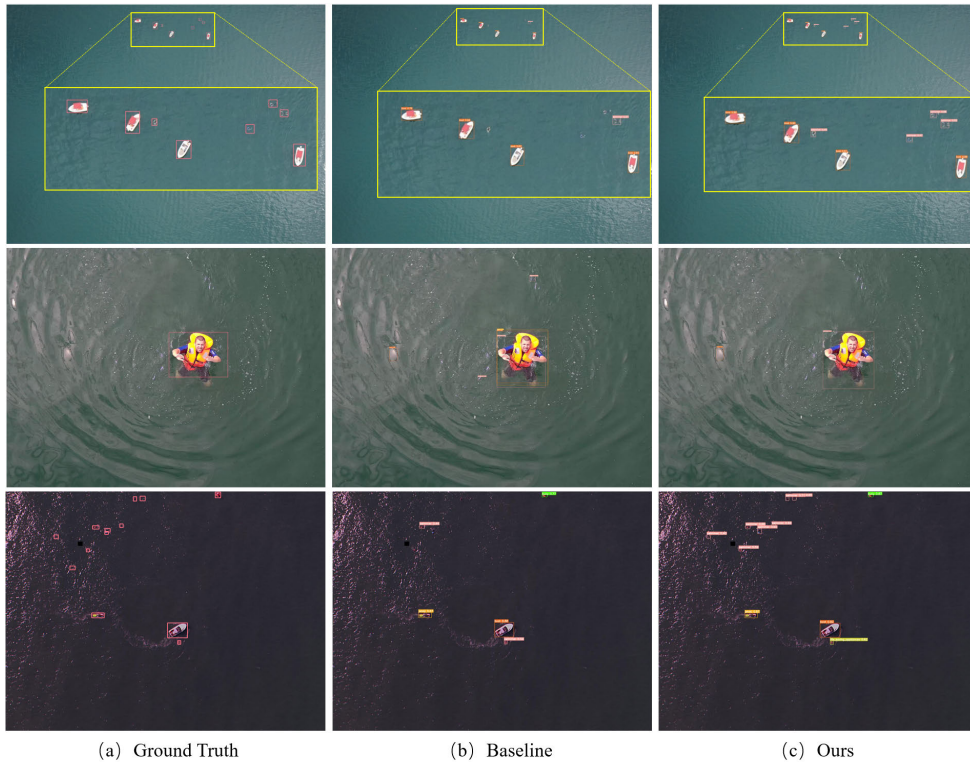


FIGURE 8. Sample detections on SeaDronesSeeV2 dataset, zoom in for more details. From left to right column: ground truth, baseline and our proposed method.

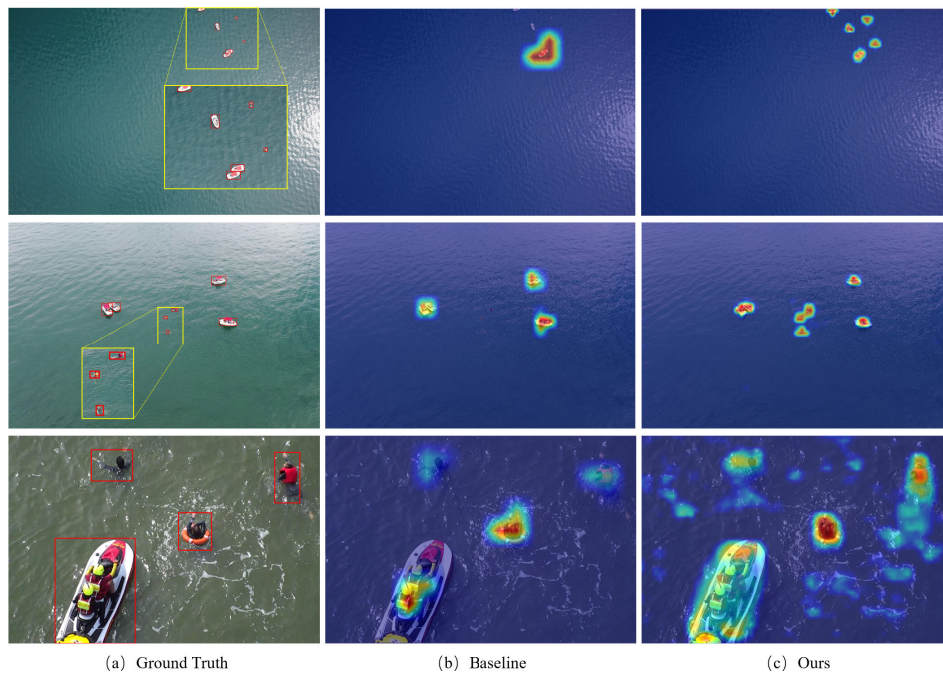


FIGURE 9. The sample heatmaps on the SeaDronesSeeV2 dataset. From left to right column: ground truth, baseline and our proposed method.

AP, AP50, and AP75 improved by 1.2%, 1.6%, and 3.7% respectively. Additionally, AP_S , AP_M , and AP_L also showed improvements of 1.4%, 1.6%, and 0.5%, respectively. When

β value is set to 0.1 and 0.9, the detection accuracy slightly decreases due to a mismatch between the object distribution and size range in the dataset and the β value. Furthermore,



FIGURE 10. Some representative detections on VisDrone2019 dataset, zoom in for more details.

TABLE 9. The influence of the threshold β value on the SeaDronesSeeV2 dataset.

	AP	AP50	AP75	AP _S	AP _M	AP _L
Baseline(CIoU)	42.6	77.0	40.9	32.5	44.0	59.3
$\beta = 0.1$	41.9	76.9	39.2	30.5	43.3	58.2
$\beta = 0.5$	43.8	78.6	44.6	33.9	45.6	59.8
$\beta = 0.9$	39.5	74.5	36.7	28.3	41.9	56.0
GIoU	42.3	76.9	39.5	33.2	43.9	58.5
DIoU	43.3	78.3	40.9	32.1	44.5	59.6

we compared the experimental results with DIOU and GIoU, and our method still outperformed these two methods when β was set to 0.5. Therefore, the above experimental results demonstrate the effectiveness of adopting the scale-aware CIoU loss function with normalized Wasserstein distance.

8) SAMPLE DETECTION RESULTS

We have selected representative images from the SeaDronesSeeV2 validation dataset, as shown in Fig. 8. From left to right, the images represent the Ground Truth, the Baseline, and our method. We deliberately chose scenarios that include small objects, large objects, and clustered objects. The comparison clearly demonstrates that our method excels in dealing with multi-scale objects, overcoming the limitations of the Baseline method, which excels at accurately identifying small objects and ensuring error-free detection of large objects.

We further visualize the heatmaps of the baseline model and the model incorporating the LCSA module, as depicted in Fig. 9. The baseline model fails to distinguish the boundaries of small targets and objects that are closely grouped together. In contrast, our method provides clearer and more accurate delineation of the objects and their boundaries. It significantly improves the issue of missed detections of small objects and achieves precise localization for medium and large objects. To showcase the test results, we have selected representative images from the VisDrone2019 dataset, as presented in Fig. 10. Our method consistently demonstrates high detection performance for both large and small objects, as well as in scenarios with densely packed objects.

TABLE 10. Discussion on different orders of using both LCSA module and PANet.

Methods	AP	AP50	AP75	Param(M)	GFLOPs	Speed(FPS)
YOLOv5(w/o Neck)	40.7	73.7	40.9	24.0	67.5	192.0
LCSA+PANet	47.3	81.3	48.0	50.0	170.0	76.3
PANet+LCSA	47.0	81.1	47.9	50.0	170.0	76.2

9) ADDITIONAL EXPERIMENTS

To investigate the impact of using LCSA module and FPN structure in different orders, we compared the results of using LCSA+FPN with those of using FPN+LCSA methods. As shown in Tab. 10, the experimental results demonstrate that the initial implementation of LCSA for feature enhancement resulted in a slight performance improvement, achieving a gain of 0.3%, 0.2% and 0.1% on AP, AP50 and AP75. Compared to using FPN first, LCSA promotes the interaction of semantic information across different scales to enhance local discriminative features of objects.

V. CONCLUSION

In this paper, we propose a novel learnable LCSA feature fusion module for UAV object detection. Serving as an innovative neck approach, LCSA not only outperforms existing neck methods, but also complements to them. Besides, the LCSA module can be plugged into multiple state-of-the-art object detectors, which improve performance consistently, showcasing its broad applicability. It demonstrates remarkable effectiveness by facilitating semantic interaction between cross-scale features and enhancing local discriminative features of objects. Experiments on the VisDrone2019-DET and SeaDronesSeeV2 datasets demonstrate that our method can obtain state-of-the-art performance.

Although our LCSA module has good performance, but it is not enough to fully meet the needs of real-time monitoring in drone scenarios. In the future, we will explore more efficient lightweight cross-scale feature fusion methods, and reduce the computational overhead of the LCSA module through sparse or partial sampling methods without significantly affecting performance, making it advantageous in practical scenarios.

REFERENCES

- [1] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs)," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4992–4997.
- [2] C. Li and H. Duan, "Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function," *Aerosp. Sci. Technol.*, vol. 39, pp. 352–360, Dec. 2014.
- [3] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.
- [4] L. A. Varga, B. Kiefer, M. Messmer, and A. Zell, "SeaDronesSee: A maritime benchmark for detecting humans in open water," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 3686–3696.
- [5] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.
- [6] Y. Huang, J. Chen, and D. Huang, "UFPMP-Det: Toward accurate and efficient object detection on drone imagery," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 1, pp. 1026–1033.
- [7] L. Jiang, B. Yuan, J. Du, B. Chen, H. Xie, J. Tian, and Z. Yuan, "MFF-SODNet: Multiscale feature fusion small object detection network for UAV aerial images," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–14, 2024.
- [8] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "CSWin transformer: A general vision transformer backbone with cross-shaped windows," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12114–12124.
- [9] W. Wang, W. Chen, Q. Qiu, L. Chen, B. Wu, B. Lin, X. He, and W. Liu, "CrossFormer++: A versatile vision transformer hinging on cross-scale attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3123–3136, May 2024.
- [10] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "PVT v2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, Sep. 2022.
- [11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer*, 2020, pp. 213–229.
- [12] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen, "Detrs beat YOLOs on real-time object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Dec. 2024, pp. 16965–16974.
- [13] Y. Chen, C. Zhang, B. Chen, Y. Huang, Y. Sun, C. Wang, X. Fu, Y. Dai, F. Qin, Y. Peng, and Y. Gao, "Accurate leukocyte detection based on deformable-DETR and multi-level feature fusion for aiding diagnosis of blood diseases," *Comput. Biol. Med.*, vol. 170, Mar. 2024, Art. no. 107917.
- [14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.
- [15] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "MaxViT: Multi-axis vision transformer," in *Proc. 17th Eur. Conf. Comput. Vis. Cham, Switzerland: Springer*, 2022, pp. 459–479.
- [16] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 516–520.
- [17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [18] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 734–750.
- [19] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8574–8586, Aug. 2022.
- [20] J. Wang, C. Xu, W. Yang, and L. Yu, "A normalized Gaussian Wasserstein distance for tiny object detection," 2021, *arXiv:2110.13389*.
- [21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
- [22] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
- [23] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [24] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [25] G. Jocher et al., "Ultralytics/YOLOv5: V7.0-YOLOv5 sota realtime instance segmentation," *Zenodo*, Nov. 2022.
- [26] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.
- [27] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9626–9635.
- [28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–20.
- [29] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
- [30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–12.
- [31] J. Qu, Z. Tang, L. Zhang, Y. Zhang, and Z. Zhang, "Remote sensing small object detection network based on attention mechanism and multi-scale feature fusion," *Remote Sens.*, vol. 15, no. 11, p. 2728, May 2023.
- [32] X. Li, W. Diao, Y. Mao, P. Gao, X. Mao, X. Li, and X. Sun, "OGMN: Occlusion-guided multi-task network for object detection in UAV images," *ISPRS J. Photogramm. Remote Sens.*, vol. 199, pp. 242–257, May 2023.
- [33] D. Paz, H. Zhang, and H. I. Christensen, "Tridentnet: A conditional generative model for dynamic trajectory generation," in *Proc. Int. Conf. Intell. Auton. Syst.*, 2021, pp. 403–416.
- [34] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2778–2788.
- [35] L. Zhou, S. Zhao, Z. Wan, Y. Liu, Y. Wang, and X. Zuo, "MFEFNet: A multi-scale feature information extraction and fusion network for multi-scale object detection in UAV aerial images," *Drones*, vol. 8, no. 5, p. 186, May 2024.
- [36] V. Mnih, N. Heess, and A. Graves, "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–15.
- [37] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
- [38] M. Wu, D. Huang, Y. Guo, and Y. Wang, "Distraction-aware feature learning for human attribute recognition via coarse-to-fine attention mechanism," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12394–12401.
- [39] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.
- [40] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.
- [41] H. Lee, H.-E. Kim, and H. Nam, "SRM: A style-based recalibration module for convolutional neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1854–1862.
- [42] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
- [43] C. R. Chen, Q. Fan, and R. Panda, "CrossViT: Cross-attention multi-scale vision transformer for image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 347–356.
- [44] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
- [45] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," 2021, *arXiv:2107.00641*.
- [46] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, "Axial-DeepLab: Stand-alone axial-attention for panoptic segmentation," in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer*, 2020, pp. 108–126.

- [47] S. Du, W. Pan, N. Li, S. Dai, B. Xu, H. Liu, C. Xu, and X. Li, "TSD-YOLO: Small traffic sign detection based on improved YOLOv8," *IET Image Process.*, vol. 1, pp. 1–17, Jun. 2024.
- [48] P. O. Pinheiro, T. Y. Lin, R. Collobert, and P. Dollr, "Learning to refine object segments," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 75–91.
- [49] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 519–534.
- [50] S. Honari, J. Yosinski, P. Vincent, and C. Pal, "Recombinator networks: Learning coarse-to-fine feature aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5743–5752.
- [51] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *14th Eur. Conf.*, 2016, pp. 483–499.
- [52] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "PANet: Few-shot image semantic segmentation with prototype alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9196–9205.
- [53] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.
- [54] Y. Luo, X. Cao, J. Zhang, J. Guo, H. Shen, T. Wang, and Q. Feng, "CE-FPN: Enhancing channel information for object detection," *Multimedia Tools Appl.*, vol. 81, no. 21, pp. 30685–30704, Sep. 2022.
- [55] S. Huang, Z. Lu, R. Cheng, and C. He, "FaPN: Feature-aligned pyramid network for dense image prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 844–853.
- [56] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.
- [57] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.
- [58] M. Yasir, L. Shanwei, X. Mingming, S. Hui, M. S. Hossain, A. T. I. Colak, D. Wang, W. Jianhua, and K. B. Dang, "Multi-scale ship target detection using SAR images based on improved Yolov5," *Frontiers Mar. Sci.*, vol. 9, Jan. 2023, Art. no. 1086140.
- [59] B. Xu, X. Cui, W. Ji, H. Yuan, and J. Wang, "Apple grading method design and implementation for automatic grader based on improved YOLOv5," *Agriculture*, vol. 13, no. 1, p. 124, Jan. 2023.
- [60] X. Jia, Y. Tong, H. Qiao, M. Li, J. Tong, and B. Liang, "Fast and accurate object detector for autonomous driving based on improved YOLOv5," *Sci. Rep.*, vol. 13, no. 1, p. 9711, Jun. 2023.
- [61] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.
- [62] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 821–830.
- [63] H. Lin, J. Zhou, Y. Gan, C.-M. Vong, and Q. Liu, "Novel up-scale feature aggregation for object detection in aerial images," *Neurocomputing*, vol. 411, pp. 364–374, Oct. 2020.
- [64] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, "VarifocalNet: An IoU-aware dense object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8510–8519.
- [65] C. Xu, J. Wang, W. Yang, H. Yu, L. Yu, and G.-S. Xia, "Detecting tiny objects in aerial images: A normalized Wasserstein distance and a new benchmark," *ISPRS J. Photogramm. Remote Sens.*, vol. 190, pp. 79–93, Aug. 2022.
- [66] C. Li, T. Yang, S. Zhu, C. Chen, and S. Guan, "Density map guided object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 737–746.
- [67] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8310–8319.
- [68] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, and J. Dong, "RRNet: A hybrid detector for object detection in drone-captured images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 100–108.

XIN ZUO received the B.S. degree from the School of Computer Science, East China Shipbuilding Institute, in 2003, the M.S. degree from Jiangsu University of Science and Technology, Zhenjiang, China, in 2007, and the Ph.D. degree from the School of Computer Science and Engineering, Southeast University, Nanjing, in 2014. Her research interests include image retrieval and image registration.

CHENHUI QI received the B.S. degree in computer science and technology from Jiangsu University of Science and Technology, Zhenjiang, China, in 2022. He is currently pursuing the M.S. degree. His current research interests include computer vision, image processing, machine learning, and deep learning.

YIFEI CHEN received the B.S. degree from the School of Electronic Information Engineering, Jiangsu University of Science and Technology, China, in 2019. He is currently pursuing the degree with the School of Electronic Information Engineering, Jiangsu University. His research interests include object detection and multispectral object detection.

JIFENG SHEN received the M.S. degree in computer science from Jiangsu University of Science and Technology, Zhenjiang, China, in 2006, and the Ph.D. degree from Southeast University, Nanjing, China, in 2013. He is currently a Faculty Member with the School of Electronic and Information Engineering, Jiangsu University, Zhenjiang. His research interests include pattern recognition and computer vision.

HENG FAN received the B.S. degree from Huazhong Agricultural University, Wuhan, China, in 2013, and the Ph.D. degree from Stony Brook University, Stony Brook, NY, USA, in 2021. Currently, he is an Assistant Professor with the Department of Computer Science and Engineering, University of North Texas, Denton, TX, USA. His research interests include computer vision, machine learning, pattern recognition, and medical image analysis. He has served as the Area Chair for the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2022 and 2023.

WANKOU YANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Computer Science and Technology, Nanjing University of Science and Technology (NUST), China, in 2002, 2004, and 2009, respectively. From July 2009 to August 2011, he was a Postdoctoral Fellow with the School of Automation, Southeast University, China. From August 2010 to August 2011, he was a Postdoctoral Fellow with the Face Aging Group, UNC Wilmington, NC, USA. Since September 2011, he has been an Assistant Professor with the School of Automation, Southeast University. His research interests include pattern recognition, computer vision, and machine learning.

• • •