An Aerial Object Detection Model Based on Dynamic Feature Cropping and Multi-Level Guiding

1st Lanbin Liang
School of Cyber Security and Computer
Hebei University
Baoding, China
3454424910@qq.com

2nd Wenzhu Yang*
School of Cyber Security and Computer
Hebei University
Machine Version Engineering Research Center
Hebei University
Baoding, China
wenzhuyang@hbu.edu.cn

***Abstract*—To address the challenges in aerial object detection, including complex backgrounds, large number of small targets, and the redundant computations incurred by existing detection models during feature extraction, this paper proposes a novel aerial detection model based on Dynamic Feature Cropping and Multi-Level guiding(DFCMNet). First, a Lightweight Feature Extraction(LFE) module based on dynamic information filtering is designed to enable the model to dynamically distinguish between salient and redundant features during training, thereby performing feature extraction operations with different levels of computational complexity to reduce model redundancy. Second, this paper proposes an improved detection layer, incorporating a shallow-level feature interaction module into the neck's upsampling network to enhance the model's capacity to extract detailed information. By using the C2 and C3 layers, the model is specifically tailored to detect small scale objects in aerial images. Third, an efficient backbone network is constructed by integrating the proposed LFE module into the feature learning layers, yielding a compact feature representation while emphasizing the acquisition of shallow-layer information. Furthermore, this study performs multi-scale modeling in the feature space and incorporates deformable convolutions together with a full-dimension, parameter-free attention module, enabling the model to learn complementary representations of multi-scale object information and contextual information. An optimized localization loss is subsequently adopted to reduce computational cost, accelerate convergence, and maintain high detection accuracy. On the VisDrone-DET2019 and AI-TOD datasets, the proposed model achieves mAP50 improvements of 7.6% and 8.9% over the baseline, respectively, while maintaining a parameter count of only 1.4M, demonstrates that our method achieves a favorable balance between detection accuracy and model size, exhibiting a clear advantage over existing aerial object detection models.**

***Index Terms*—Aerial images; Object detection; Lightweight; Information filtering.**

I. Introduction

With the rapid development and widespread application of unmanned devices and related technologies, object detection in aerial images has become an important research direction in the field of computer vision. It demonstrates tremendous potential in various domains, including smart cities, transportation, environmental protection, agriculture, and the military. However, aerial object detection faces unique challenges: first, due to the long shooting distances, most objects in the images are small with limited pixel information, making feature extraction difficult; second, the scenes often contain a large number of densely distributed objects, resulting in severe occlusion and clustering; moreover, there is a significant variation in object scales, with small objects and larger targets possibly coexisting in the same scene.

To address these challenges, numerous deep learning–based object detection models have been proposed. Representative detectors include Fast R-CNN [1] and Faster R-CNN [2], which are classic two-stage detection frameworks. These models employ a region proposal network to generate candidate regions and subsequently predict object categories and bounding boxes based on the proposed regions. Although two-stage detectors generally achieve high detection accuracy, they suffer from substantial model complexity and slow inference speed, making them difficult to deploy in aerial detection scenarios. Single-stage detectors, represented by the YOLO family [3], have gained widespread attention due to their favorable trade-off between speed and accuracy and have been extensively applied to aerial detection tasks. However, these models exhibit suboptimal performance when applied to aerial object detection tasks. To this end, researchers have attempted to reduce model size by introducing lightweight convolutional modules or efficient feature extraction strategies in the backbone network or detection head structures [4], while improving detection performance through enhanced feature learning strategies [5]. Existing approaches commonly adopt pyramid-based feature fusion modules [6] to strengthen multi-scale feature interactions; however, this inevitably increases the computational burden. Other studies [7] have tailored the feature extraction network to improve performance in aerial detection environments by mitigating the loss of fine-grained details and enhancing the detection of small-scale objects, but such improvements likewise incur substantial computational overhead.

To address the issues of existing detection models, this paper proposes a lightweight aerial object detection model driving by Dynamic Feature Cropping and Multi-Level guiding(DFCMNet). The main contributions of this work are summarized as follows.

- To address the extensive redundant computations in the

*Corresponding author.

feature extraction process, we propose a dynamic feature partitioning strategy that employs learnable channel-wise variables during training to guide the model in distinguishing informative features from redundant ones, thereby enabling the design of a lightweight feature learning module.

- By enhancing the detection layers of the model, its capability to detect small objects is improved.
- The backbone network of the detection model is improved by incorporating the proposed lightweight module, which refines the feature learning strategy, alleviates model redundancy, and enables more effective extraction of shallow-layer information, thereby enhancing the detection performance for small targets.
- In response to the distinctive characteristics of aerial images, an improved Spatial Pyramid Pooling Fast method is designed to effectively extract multi-scale target and contextual features, thereby enhancing the model's representational flexibility and promoting its ability to capture object information in complex environments.
- The regression loss function is optimized to accelerate convergence, reduce the network's parameter count and computational complexity, while maintaining the accuracy of object detection.

II. Method

To address the imbalance between detection accuracy and model complexity in aerial image object detection, we propose a detection model from an aerial perspective based on Dynamic Feature Cropping and Multi-Level guiding(DFCMNet), whose overall architecture is illustrated in Fig. 1. The design of the proposed model mainly comprises the following five components: (1)**Lightweight Feature Extraction(LFE) module based on dynamic information filtering**, which aims to reduce computational resource consumption while maintaining detection accuracy; (2)**Improved detection layer structure**, designed to enhance the model's capability to recognize small objects; (3)**Enhanced backbone network**, in which the feature learning layers are optimized by incorporating the proposed LFE module; (4)**ISPP module**, which performs multi-scale categorization of objects in the image, models contextual feature information, and guides the model to focus on regions of interest, thereby improving detection performance in complex backgrounds; (5)**Optimized loss function**, which accelerates model convergence, improves detection accuracy, and reduces both parameter count and computational cost. The following sections provide a detailed description of each module and the corresponding improvements.

A. Lightweight Feature Extraction Module Based on Dynamic Information Filtering

In conventional convolutional operations, the convolution kernel applies convolution across every channel of the input feature map to extract spatial information. However, in practice, the channels of a feature map are not entirely independent; rather, they exhibit inter-channel correlations.

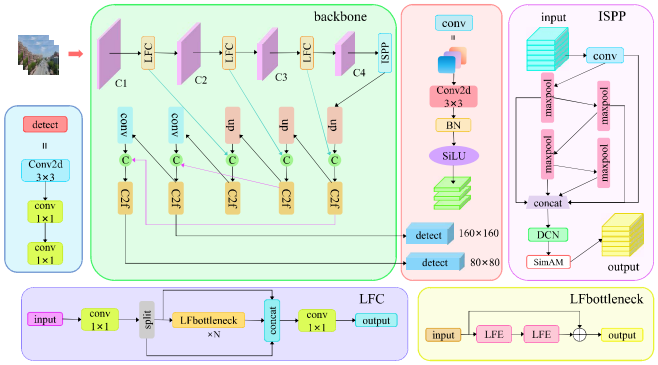

Fig. 1. Architecture of DFCMNet.

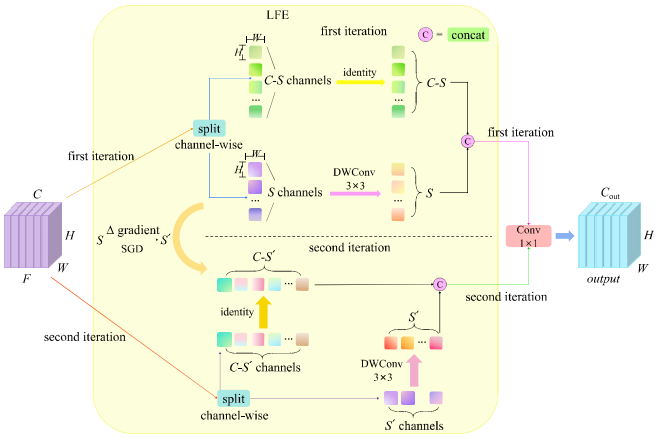

Fig. 2. Architecture of the LFE module.

For instance, certain channels may exhibit high similarity and jointly emphasize salient object-related features, while others may predominantly encode task-irrelevant background information. Performing standard convolutions uniformly across all channels inevitably incurs substantial computational overhead and introduces significant redundancy. To address this inefficiency, convolutional operations should be adapted according to the relative importance of different channels. Specifically, high-priority channels—those carrying critical semantic cues—should undergo full-scale convolution, whereas low-priority channels could be processed with lightweight convolutional operators. This prioritized, adaptive strategy offers an effective means to reduce redundant computation, enhance inference efficiency, and maintain model accuracy.

Existing lightweight feature learning modules rely on a predefined, fixed partitioning scheme, in which features are divided into key and redundant components according to a predetermined ratio. In contrast, we propose a novel feature allocation strategy that introduces a learnable variable representing the number of channels assigned to key features. During training, this variable is adaptively adjusted to progressively converge toward the ground-truth distribution. Unlike fixed settings, our data-driven approach is guided by the training data, enabling more accurate feature partitioning.

Consequently, the proposed strategy is more principled and practical.

The architecture of the Lightweight Feature Extraction(LFE) module based on dynamic information filtering is illustrated in Fig. 2. A learnable parameter S is employed to partition the input feature F along the channel dimension. We initialize S as $C/2$ and optimize S using Stochastic Gradient Descent (SGD).

$$F \xrightarrow[\text{dim}=1]{\text{split}} \begin{cases} F_1,\ F_1 \in \mathbb{R}^{S\times H\times W}; \\ F_2,\ F_2 \in \mathbb{R}^{(C-S)\times H\times W}. \end{cases} \qquad F \in \mathbb{R}^{C\times H\times W} \tag{1}$$

The key features are denoted as F_1, while the redundant features are denoted as F_2 (where S represents the number of channels marked as important). Therefore, we designate the representative features F_1 to participate in the convolution operations, enabling the extraction of spatial contextual information from the image.

$$F_1^* = \text{DWConv}_{3\times3}(F_1) \tag{2}$$

The secondary features, F_2, do not participate in any operations. The features obtained after the convolution are denoted as $F_1^* \in \mathbb{R}^{S\times H\times W}$. Subsequently, F_1^* and F_2 are concatenated along the channel dimension (dimension = 1) to form a new feature representation, denoted as F^*.

$$F^* = \text{concat}_{(\text{dim}=1)}(F_1^*,\ F_2) \tag{3}$$

The convolution operation is a 3×3 depthwise convolution with a stride of 1.

$$output = \text{SiLU}(\text{BN}(\text{Conv2d}_{1\times1}(F^*))) \tag{4}$$

Finally, a 1×1 convolution is applied to F^* to perform a channel-wise transformation, yielding the output features.

B. Enhanced Detection Layer Focusing on Shallow Features

The architecture of the detection model consists of a backbone network, a neck with upsampling and downsampling pathways, and a detection head. The input image is sequentially processed by the backbone, the neck, and the detection head to complete the inference process. The original detection network comprises five hierarchical stages, denoted as C1 to C5. In this work, we remove certain redundant structures, including the C5 layer in both the backbone and the detection head, as these layers are primarily designed to extract features of large-scale objects rather than small targets. Building upon the backbone, the neck network performs progressive upsampling to obtain higher-level semantic features and complements them with lower-level features to enable information exchange and fusion. Although the standard neck network effectively performs feature fusion, it is insufficient in capturing informative cues for small objects. The conventional upsampling neck is composed of the C3 and C2 layers. On this basis, we introduce a new C1 layer with a spatial resolution of 320×320 to acquire richer and more comprehensive representations of small-scale objects. This layer encodes abundant fine-grained details and high-level semantic information relevant to small targets. Consequently, the proposed C1 layer enables the model to more effectively learn small-object characteristics in aerial scenes, thereby significantly enhancing small-object detection performance.

In some studies, researchers [8] have improved the detection head architecture of models. Specifically, they adjusted the original five-layer structure to a four-layer design, consisting of C1, C2, C3, and C4 layers. Additionally, they introduced a C2 layer in the neck upsampling network and the detection head, and added a C3 layer in the neck downsampling structure. These modifications aimed to reduce the loss of small-object information and enhance the overall performance of the model in aerial object detection tasks. Although this approach achieved promising detection results, further optimization of the detection head architecture is necessary to continue improving performance. Considering that objects in aerial images are generally very small, we removed the C4 layer detection head and its directly associated neck downsampling C4 layer, which are more suitable for medium and large-scale object detection. Moreover, due to the substantial computational complexity and high memory consumption of the C1 layer detection head, it was also excluded. Ultimately, only the C2 and C3 layers were retained as the detection head structure. This design balances the scale characteristics of objects in aerial images with the overall model size and computational complexity.

C. Backbone Network Reconstruction

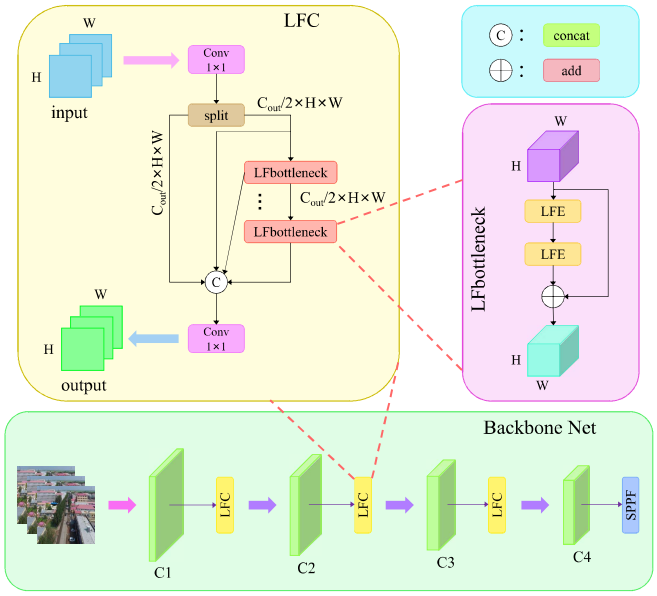

Fig. 3. Architectures of improved backbone network.

We enhance the model efficiency by designing a lightweight feature-extraction backbone network. The standard C2f module employs Bottleneck blocks to optimize gradient flow and enable high-level feature representation. Each Bottleneck consists of two cascaded convolutional layers and incorporates

residual connections to alleviate the vanishing gradient problem during training. In the Bottleneck, standard convolutions are adopted. In this work, we replace these conventional convolutions with the proposed LFE module based on dynamic information filtering, and denote the resulting improved Bottleneck as LFbottleneck. Accordingly, the entire multi-gradient-flow feature learning module is referred to as LFC. The execution procedure is as follows. The input features are first passed through a 1×1 convolution block to align the channel dimensions, after which the features are split into two branches along the channel dimension. One branch is sequentially processed by one, two, and multiple LFbottleneck blocks, producing hierarchical features ranging from low-level to high-level representations. Finally, the primary features that bypass the LFbottleneck blocks are concatenated with the features at each hierarchical level, and a 1×1 convolution layer is applied to generate the output features. The overall workflow is illustrated in Fig. 3. Owing to the adoption of lightweight feature-extraction operations, the proposed LFC achieves reduced parameter count and computational cost, while maintaining efficient feature extraction and learning capabilities. Furthermore, the multi-level feature representations and enriched gradient flow facilitate comprehensive information modeling from the input images.

Conventional detection backbones typically incorporate downsampling operations based on 3×3 convolutions followed by deeper feature-extraction modules to capture image representations and perform feature learning. In standard architectures, the C1 stage consists of only a single convolutional operation, without detailed feature extraction, and directly proceeds to a second downsampling step to enter the C2 stage. Given that aerial images contain a large number of small objects, accurate detection in such scenarios requires abundant fine-grained details to be preserved in the feature maps. Since lower-level layers retain richer spatial and structural information, we insert the LFC feature learning module after the downsampling operation at the C1 stage to enhance the extraction and learning of detailed features, thereby improving the model's ability to recognize small targets.

D. ISPP Module

To enhance the model's multi-scale feature extraction capability, this work proposes an improved Spatial Pyramid Pooling Fast module, referred to as the ISPP module.

The receptive field of a standard convolution is fixed in shape and size, whereas objects in images are often irregular, complex, and diverse, which limits the ability of convolutions to effectively capture target features. DCNv3 [9] addresses this limitation by using learnable offset parameters to dynamically adjust sampling positions, enabling more accurate feature capture. The input feature F is first passed through a 1×1 convolution for channel alignment, producing the feature map denoted as A.

$$A = \text{SiLU}(\text{BN}(\text{Conv2d}_{1\times1}(F))) \tag{5}$$

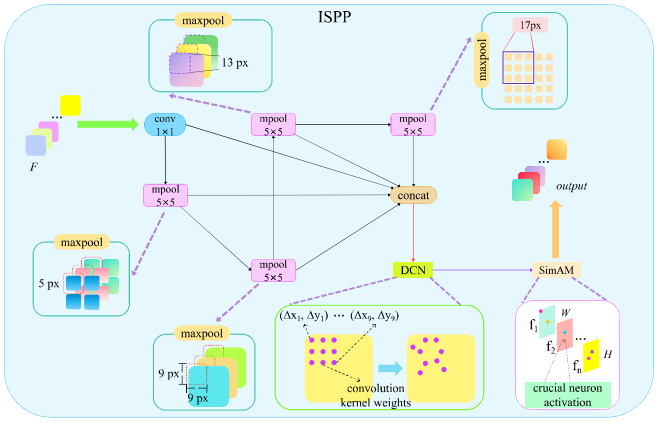

Fig. 4. Architecture of the ISPP module.

The feature map A is then subjected to 5×5 pooling sequentially one, two, three, and four times, producing feature maps a, b, c, and d, respectively. Performing 5×5 pooling twice is equivalent to a 9×9 pooling, three times corresponds to 13×13 pooling, and four times corresponds to 17×17 pooling.

$$a = \text{Maxpool}_{5\times5}(A),\ b = \text{Maxpool}_{5\times5}(a) \tag{6}$$

$$c = \text{Maxpool}_{5\times5}(b),\ d = \text{Maxpool}_{5\times5}(c) \tag{7}$$

A total of four levels of pooling operations are performed to model the large-scale variations in aerial image features, enabling the network to better classify objects of different scales and thereby improve detection performance. Subsequently, the five feature maps A, a, b, c, and d are concatenated to form the final feature representation, denoted as F^*.

$$F^* = \text{concat}((A, a, b, c, d), \text{dim} = 1) \tag{8}$$

Subsequently, the feature map F^* is processed through a deformable convolution operation and the SimAM attention module, yielding the final output features.

$$output = \text{SimAM}(\text{DCN}_{3\times3}(F^*)) \tag{9}$$

Deformable convolutions provide precise feature representations on top of multi-scale contextual feature modeling, enabling the model to focus on regions of interest within the feature maps and enhancing the network's image understanding capability. SimAM [10] is a parameter-free mechanism that constructs an energy function based on the relationships among neurons in neuroscience to assess the importance of each neuron in the network. It thereby guides the model to focus on salient positions across different feature dimensions, ultimately improving detection accuracy.

E. Loss Function Optimization

To further improve the model's convergence speed and the localization accuracy of bounding box regression, this work adopts an optimized object localization loss function, MPDIoU [11]. The MPDIoU loss is an enhancement of the traditional IoU and its variants, incorporating a projection

distance-based penalty term to more precisely measure the geometric discrepancy between the predicted and ground-truth bounding boxes.

This optimization not only accelerates the model's training convergence but also significantly improves detection accuracy, particularly for small objects commonly found in aerial images. Moreover, compared with CIoU [12] and DFL losses, MPDIoU does not introduce additional parameters and computational overhead, as it does not require generating multi-element arrays in the detection head for computing the DFL loss, and it avoids the arctangent operations required by the CIoU loss as well as the exponential and logarithmic operations involved in the DFL loss.

III. Experimental Results and Analysis

A. Datasets

a) VisDrone-DET2019: VisDrone-DET2019 [13] is a large-scale open-source UAV aerial imagery dataset released by the Data Mining Research Center of Tianjin University. It comprises 6,471 images in the training set, 548 images in the validation set, and 1,610 images in the test set. The data are collected from diverse UAV-mounted cameras and span a wide range of scenarios, including different cities, environments, and object categories. The VisDrone-DET2019 dataset is characterized by substantial variations in object scale, a high density of small targets, and complex backgrounds. It includes ten categories: pedestrian, person, bicycle, car, van, truck, tricycle, awning tricycle, bus, and motorcycle.

b) AI-TOD: The AI-TOD [14] dataset is specifically designed for object detection in high-altitude aerial imagery. It comprises 11,214 images in the training set, 14,018 images in the test set, and 2,804 images in the validation set, covering eight object categories: airplane, bridge, storage tank, ship, swimming pool, vehicle, person, and wind turbine. Compared with existing aerial object detection benchmarks, the AI-TOD dataset features an average object size of approximately 12.8 pixels, which is substantially smaller than that in other datasets, and exhibits pronounced background clutter and occlusion.

B. Experimental Setup

To evaluate the effectiveness of the proposed UAV-based small-object detection method, a comprehensive experimental assessment was conducted. All experiments were performed on a hardware platform equipped with an Intel(R) Xeon(R) E5 CPU and an NVIDIA RTX 3090 24GB GPU. The models in all experiments were implemented and trained using PyTorch in Python. The main hyperparameter settings were as follows: input image resolution of 640×640, initial and final learning rates (Ir0 and Irf) set to 0.01, and training for 100 epochs. The batch size was set to 8 when using the VisDrone2019 dataset and 4 for the AI-TOD dataset.

C. Comparative Analysis of the Enhanced Detection Layer

The improved detection layer proposed in this study is compared with the detection layers of several other models.

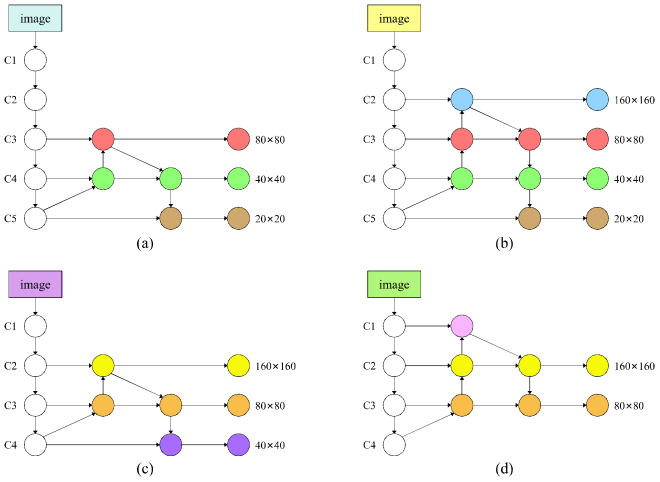

Fig. 5. Different detection layer architectures.

TABLE I
Comparative Analysis of Experimental Results for Various Detection Layer Architectures on the VisDrone2019 Dataset

Layer	mAP50	mAP50:95	Params(M)	FLOPs(G)
a	36.5	21.5	11.1	**28.7**
b	40.4	24.0	3.3	30.1
c	40.2	23.6	10.6	37.0
d (ours)	**41.0**	**24.4**	**2.3**	30.9

The corresponding detection layer architectures are illustrated in Fig. 5, and the comparative experimental results are reported in Tables I and II. Specifically, subfigures a, b, c, and d (ours) in Fig. 5 correspond one-to-one with the entries a, b, c, and d (ours) in Tables I and II. Our improved detection layers proposed in this study achieves the highest accuracy on both the VisDrone2019 and AI-TOD datasets. Specifically, the mAP50 reaches 41.0% on VisDrone2019 and 38.8% on AI-TOD. Compared with detection layers in group a, the mAP50 increases by 4.5% and 4.6% on the two datasets, respectively. Compared with group c, our designed detection head yields a 0.8% improvement in both mAP50 and mAP50:95 on VisDrone2019, with only 21% of the parameters of group c and a reduction of 6.1G in computational complexity.

D. Comparative Analysis of the LFE Module

From the comparative results in Table III and Table IV, compared with PConv, the proposed module decreases the parameter count by 0.2M and the computational complexity

TABLE II
Comparative Analysis of Experimental Results for Various Detection Layer Architectures on the AI-TOD Dataset

Layer	mAP50	mAP50:95	Params(M)	FLOPs(G)
a	34.2	14.5	11.1	**28.7**
b	38.4	16.7	3.3	30.1
c	38.3	16.5	10.6	37.0
d (ours)	**38.8**	**17.1**	**2.3**	30.9

TABLE III
EXPERIMENTAL RESULTS OF VARIOUS LIGHTWEIGHT MODULES ON THE VISDRONE2019 DATASET

Module	mAP50	mAP50:95	Params(M)	FLOPs(G)
GhostConv [15]	34.3	20.1	**9.4**	**24.3**
GSConv [16]	34.9	20.6	**9.4**	**24.3**
SPConv [17]	35.5	21.1	10.0	26.0
PConv [18]	36.2	21.4	9.7	24.8
LFE(ours)	**37.2**	**22.3**	9.5	24.5

TABLE IV
EXPERIMENTAL RESULTS OF VARIOUS LIGHTWEIGHT MODULES ON THE AI-TOD DATASET

Module	mAP50	mAP50:95	Params(M)	FLOPs(G)
GhostConv	33.0	13.7	**9.4**	**24.3**
GSConv	33.4	13.8	**9.4**	**24.3**
SPConv	33.5	13.9	10.0	26.0
PConv	34.1	14.2	9.7	24.8
LFE(ours)	**35.3**	**15.4**	9.5	24.5

by 0.3G, and improves mAP50 by 1.0% and 1.2% on the VisDrone2019 and AI-TOD datasets, respectively. Furthermore, when compared with GhostConv and GSConv, the proposed LFE module yields mAP50 gains of 2.9 and 2.3 percentage points on the VisDrone2019 dataset, and improvements of 2.3% and 1.9% on the AI-TOD dataset, respectively. In terms of model complexity, the proposed module introduces only an additional 0.1M parameters and 0.2G floating-point operations, thereby maintaining a low computational overhead.

E. Ablation Studies of Each Module

We designed a series of ablation experiments for each module to evaluate the impact of individual architectural components on performance improvements. The experimental results are reported in Tables V and VI, where 1, 2, 3, and 4 correspond to the enhanced detection layers, reconstructed backbone network, ISPP module, and optimized loss function, respectively.

TABLE V
ABLATION STUDY RESULTS OF EACH MODULE ON THE VISDRONE2019 DATASET

1	2	3	4	mAP50	mAP50:95	Params(M)	FLOPs(G)
×	×	×	×	36.5	21.5	11.1	28.7
√				41.0	24.4	2.3	30.9
√	√			42.5	25.5	**1.4**	27.6
√	√	√		43.6	26.5	1.6	28.2
√	√	√	√	**44.1**	**26.8**	**1.4**	**23.5**

The introduction of the improved detection layers lead to a substantial gain in accuracy, increasing mAP50 by 4.5 points on the VisDrone2019 dataset and by 4.6 points on the AI-TOD dataset. Subsequently, the enhanced backbone reduces the parameter count by 0.9M, retaining only 60% of the original size, while further improving mAP50 by 1.5% and 1.8% on VisDrone2019 and AI-TOD, respectively. Subsequently, the ISPP module is applied, resulting in further improvements of 1.1 and 1.8 percentage points in mAP50 on the VisDrone2019 and AI-TOD datasets, respectively. Finally, the adoption of the optimized loss function MPDIoU reduces the parameter count by 0.2M and the computational cost by 4.7G, while achieving mAP50 values of 44.1% and 43.1% on VisDrone2019 and AI-TOD, respectively, thereby simultaneously lowering resource consumption and enhancing performance.

TABLE VI
ABLATION STUDY RESULTS OF EACH MODULE ON THE AI-TOD DATASET

1	2	3	4	mAP50	mAP50:95	Params(M)	FLOPs(G)
×	×	×	×	34.2	14.5	11.1	28.7
√				38.8	17.1	2.3	30.9
√	√			40.6	18.2	**1.4**	27.6
√	√	√		42.4	19.0	1.6	28.2
√	√	√	√	**43.1**	**19.5**	**1.4**	**23.5**

F. Comparative Analysis of Different Object Detection Models on the VisDrone2019 Dataset

TABLE VII
COMPARATIVE EXPERIMENTAL RESULTS OF DIFFERENT MODELS ON THE VISDRONE2019 DATASET

Model	mAP50	mAP50:95	Params(M)	FLOPs(G)
SSD [19]	23.9	13.1	24.5	87.9
RetinaNet [20]	27.3	15.5	19.8	93.7
RT-DETR [21]	43.8	26.2	19.8	57.0
yolov8n	30.5	17.3	3.0	8.2
yolov8s	36.5	21.5	11.1	28.7
yolov8m	40.5	24.2	25.8	79.1
yolov9t [22]	31.1	17.9	2.0	**7.9**
yolov9s	37.4	22.1	7.2	27.4
yolov9m	41.0	24.8	20.1	77.6
yolov9c	42.0	25.3	25.5	103.7
yolov10s [23]	36.1	21.5	8.0	24.8
yolov11s [24]	36.5	21.5	9.4	21.6
ESOD-YOLO [25]	39.8	23.7	5.4	20.6
Drone-YOLO [26]	41.2	24.6	10.8	37.6
LD-YOLO [27]	36.8	21.8	3.0	23.7
MASF-YOLO [28]	41.6	24.7	12.0	44.3
YOLO-DSC [29]	40.3	24.0	11.1	33.0
DTSSNet [30]	39.9	24.2	10.1	50.4
ours	**44.1**	**26.8**	**1.4**	23.5

As shown in table VII, our proposed model achieves the highest accuracy, with mAP50 is 44.1% and mAP50:95 is 26.8%, while maintaining the lowest parameter count. Compared with YOLOv8s, our model improves mAP50 by 7.6 points and mAP50:95 by 5.3 points, with only 12% of YOLOv8s' parameters and 5.2G fewer FLOPs. Compared with YOLOv10s, our method outperforms it by 8.0 points in mAP50 and 5.3 points in mAP50:95, with only 17.5% of YOLOv10s' parameters. Furthermore, our model surpasses ESOD-YOLO by 4.3% and 3.1% in accuracy, while using less than one-third of its parameters. Our model achieves a 2.5% higher mAP50 than MASF-YOLO while exhibiting substantially lower complexity. Relative to DTSSNet, our approach achieves 4.2% and 2.6% higher mAP50 and mAP50:95, while

DTSSNet has 7.2 times more parameters and over twice the computational cost of our model.

G. Comparative Analysis of Different Object Detection Models on the AI-TOD Dataset

TABLE VIII
COMPARATIVE EXPERIMENTAL RESULTS OF DIFFERENT MODELS ON THE AI-TOD DATASET

Model	mAP50	mAP50:95	Params(M)	FLOPs(G)
SSD	21.7	7.0	24.5	87.9
RetinaNet	23.6	8.7	19.8	93.7
yolov8n	27.5	11.4	3.0	**8.2**
yolov8s	34.2	14.5	11.1	28.7
yolov8m	38.6	16.8	25.8	79.1
yolov9s	33.8	14.4	7.2	27.4
yolov9m	39.2	16.7	20.1	77.6
yolov10s	34.0	14.6	8.0	24.8
yolov11s	34.6	14.7	9.4	21.6
Drone-YOLO	38.0	16.8	10.8	37.6
SOD-YOLO [31]	40.5	17.2	1.7	20.1
MSFE-YOLO [32]	38.1	16.5	14.7	60.2
MHAF-YOLO [33]	36.3	16.0	7.4	26.8
ours	**43.1**	**19.5**	**1.4**	23.5

According to the results in table VIII, our model achieves an mAP50 of 43.1% and an mAP50:95 of 19.5% on the AI-TOD dataset. Compared with YOLOv8s, our method achieves an increase of 8.9 points in mAP50 and 5.0 points in mAP50:95. Relative to YOLOv9s, the proposed model improves mAP50 by 9.3%. Against YOLOv11s, the mAP50 improvement is 8.5%. Meanwhile, our network outperforms Drone-YOLO by 5.1 and 2.7 points in accuracy, with a parameter complexity of only 13% of Drone-YOLO. In comparison with MSFE-YOLO, our method surpasses it by 5.0% and 3.0% in detection performance, while MSFE-YOLO has a very large scale, with 14.7M parameters (10.5 times ours) and 60.2G FLOPs. Compared with MHAF-YOLO, our model achieves 6.8% and 3.5% higher mAP50 and mAP50:95, respectively, while MHAF-YOLO has more than five times our parameter count and 3.3G higher computational cost.

H. Visualization Analysis

Fig. 7 presents the detection results of the baseline and the proposed model. As illustrated, our method identifies a greater number of targets while producing fewer redundant and erroneous bounding boxes, and it achieves more precise localization of predicted boxes, thereby demonstrating superior detection performance. The performance increases of the improved detection model are particularly evident across diverse aerial scenarios, including highways, shopping centers, and densely populated streets. Consequently, it can be concluded that our approach achieves enhanced detection accuracy while simultaneously reducing model complexity.

IV. CONCLUSION

To further improve the accuracy and efficiency of aerial object detection, we propose an efficient aerial object detection model based on Dynamic Feature Cropping and Multi-Level guiding(DFCMNet). To address the redundant computations incurred during feature extraction, a dynamic partitioning strategy is introduced to distinguish important features from redundant ones, thereby guiding the model to emphasize informative representations while suppressing unnecessary computations. In addition, an optimized detection head is designed to enhance the detection performance for small-scale objects. The backbone network is refined to reduce model complexity while strengthening the extraction of shallow features. Furthermore, the ISPP is proposed to model multi-scale contextual features and to guide the network's attention toward regions of interest. Finally, the regression loss is optimized to further reduce computational overhead while maintaining detection accuracy. We conducted a series of experiments on the VisDrone2019 and AI-TOD datasets, and the results demonstrate the effectiveness and feasibility of the proposed model. In future work, we will continue to explore strategies for balancing detection speed and accuracy and extend our approach to a broader range of object detection scenarios to further enhance detector performance.

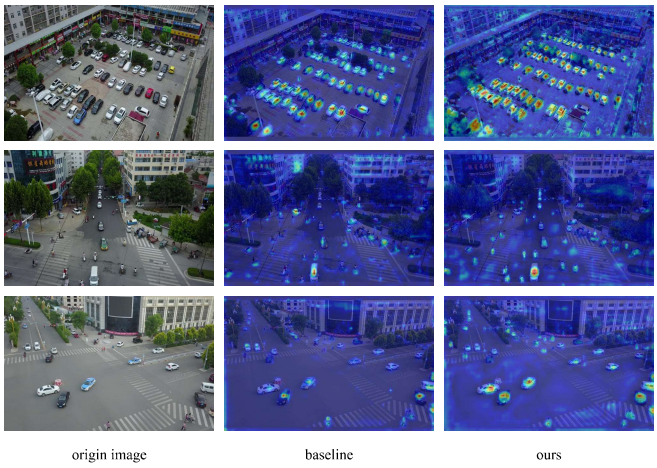

Fig. 6. Visualized heatmaps on the VisDrone2019 dataset.

REFERENCES

[1] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.
[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," arXiv preprint arXiv:1506.01497, 2016.
[3] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," arXiv preprint arXiv:2209.02976, 2022.
[4] S. Liu, J. Zha, J. Sun, Z. Li, and G. Wang, "EdgeYOLO: An Edge-Real-Time Object Detector," arXiv preprint arXiv:2302.07483, 2023.
[5] Y. Cao, D. Pang, Q. Zhao, Y. Yan, Y. Jiang, C. Tian, F. Wang, and J. Li, "Improved YOLOv8-GD deep learning model for defect detection in electroluminescence images of solar photovoltaic modules," *Engineering Applications of Artificial Intelligence*, vol. 131, p. 107866, 2024.
[6] Y. Chang, X.-B. Yang, X. Liang, and X. Zheng, "DSFPN-YOLOv8: Dynamic Multi-scale Feature Fusion Model for Small Object Detection," in *Proc. 2024 7th Int. Conf. Artif. Intell. Pattern Recognit. (AIPR)*, 2024.
[7] T. Ning, W. Wu, and J. Zhang, "Small object detection based on YOLOv8 in UAV perspective," *Pattern Analysis and Applications*, vol. 27, p. 103, 2024.

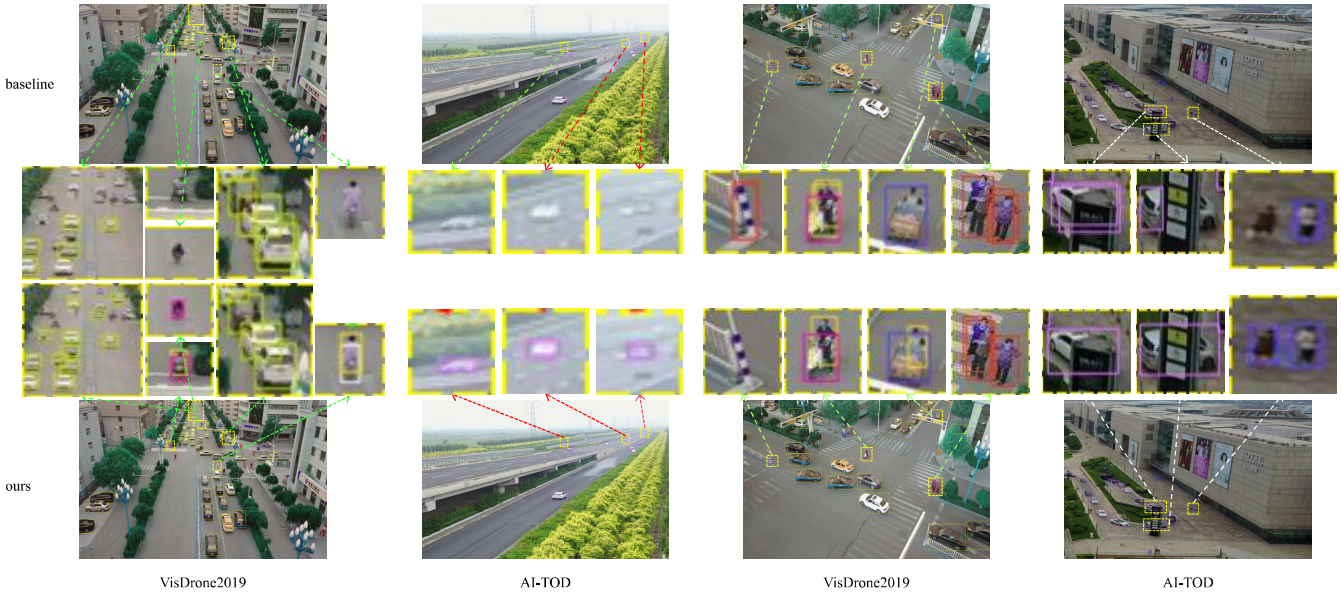

Fig. 7. Comparison of object detection results from aerial perspectives.

[8] H. Li and J. Wu, "LSOD-YOLOv8s: A Lightweight Small Object Detection Model Based on YOLOv8 for UAV Aerial Images," Eng. Lett., vol. 32, no. 11, 2024.

[9] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang, and Y. Qiao, "InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 14408–14419.

[10] L. Yang, R.-Y. Zhang, L. Li, and X. Xie, "SimAM: A Simple, Parameter-Free Attention Module for Convolutional Neural Networks," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, 2021.

[11] S. Ma and Y. Xu, "MPDIoU: A Loss for Efficient and Accurate Bounding Box Regression," arXiv preprint arXiv:2307.07662, 2023.

[12] 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," arXiv preprint arXiv:2005.03572, 2021.

[13] D. Du, P. Zhu, L. Wen, *et al.*, "VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 213–226.

[14] J. Wang, W. Yang, H. Guo, R. Zhang, and G.-S. Xia, "Tiny Object Detection in Aerial Images," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 3791–3798.

[15] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More Features From Cheap Operations," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1577–1586.

[16] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, "Slim-neck by GSConv: a lightweight-design for real-time detector architectures," *Journal of Real-Time Image Processing*, vol. 21, no. 3, pp. 785–798, Mar. 2024.

[17] Q. Zhang, Z. Jiang, Q. Lu, J. Han, Z. Zeng, S.-H. Gao, and A. Men, "Split to Be Slim: An Overlooked Redundancy in Vanilla Convolution," arXiv preprint arXiv:2006.12085, 2020.

[18] J. Chen, S.-H. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12021–12031.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.

[21] 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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16965–16974.

[22] C.-Y. Wang, I.-H. Yeh, and H. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," arXiv preprint arXiv:2402.13616, 2024.

[23] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-Time End-to-End Object Detection," arXiv preprint arXiv:2405.14458, 2024.

[24] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," arXiv preprint arXiv:2410.17725, 2024.

[25] X. Xu, Q. Li, J. Pan, X. Lu, H. Wei, M. Sun, and H. Zhang, "ESOD-YOLO: An Enhanced Efficient Small Object Detection Framework for Aerial Images," *Computing*, vol. 107, 2025.

[26] Z. Zhang, "Drone-YOLO: An Efficient Neural Network Method for Target Detection in Drone Images," *Drones*, vol. 7, no. 8, Art. no. 526, 2023.

[27] X. Qiu, Y. Chen, W. Cai, M. Niu, and J. Li, "LD-YOLOv10: A Lightweight Target Detection Algorithm for Drone Scenarios Based on YOLOv10," *Electronics*, vol. 13, no. 16, Art. no. 3269, 2024.

[28] L. Lu, D. He, C. Liu, and Z. Deng, "MASF-YOLO: An Improved YOLOv11 Network for Small Object Detection on Drone View," arXiv preprint arXiv:2504.18136, 2025.

[29] J. Zhao, Z. Zheng, and J. Fan, "YOLO-DSC: A Small Target Detection Method for Unmanned Aerial Vehicle Images," *Engineering Research Express*, vol. 7, no. 4, p. 045254, Nov. 2025.

[30] L. Chen, C. Liu, W. Li, Q. Xu, and H. Deng, "DTSSNet: Dynamic Training Sample Selection Network for UAV Object Detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–16, 2024.

[31] Y. Li, Q. Li, J. Pan, Y. Zhou, H. Zhu, H. Wei, and C. Liu, "SOD-YOLO: Small-Object-Detection Algorithm Based on Improved YOLOv8 for UAV Images," *Remote Sensing*, vol. 16, no. 16, Art. no. 3057, 2024.

[32] S. Qi, X. Song, T. Shang, X. Hu, and K. Han, "MSFE-YOLO: An Improved YOLOv8 Network for Object Detection on Drone View," *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024.

[33] Z. Yang, Q. Guan, Z. Yu, X. Xu, H. Long, S. Lian, H. Hu, and Y. Tang, "MHAF-YOLO: Multi-Branch Heterogeneous Auxiliary Fusion YOLO for Accurate Object Detection," arXiv preprint arXiv:2502.04656, 2025.