IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2024, accepted 11 May 2024, date of publication 15 May 2024, date of current version 23 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3401397

RESEARCH ARTICLE

Effective Anchor Adaptation and Feature Enhancement Strategies for Tiny Object Detection in Aerial Images

HAOGUANG LIU, QIANG TONG, LIN MIAO, AND XIULEI LIU

Laboratory of Data Science and Information Studies, Beijing Information Science and Technology University, Beijing 100101, China
Beijing Advanced Innovation Center for Materials Genome Engineering, Beijing Information Science and Technology University, Beijing 100101, China

Corresponding author: Qiang Tong (tongq85@bistu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB2600600, in part by the Research and Development Program of Beijing Municipal Education Commission under Grant KM202111232003, and in part by the Research Fund of Beijing Information Science and Technology University.

ABSTRACT In recent years, research based on anchor-based two-stage detectors has achieved great performance improvements in aerial object detection tasks. However, they still have two significant problems in the detection of tiny objects: 1) The preset fixed anchor is not conducive to assigning positive and negative samples in RPN when dealing with tiny objects, resulting in low-quality samples. 2) When the detector encounters tiny objects lacking structural details, it fails to accurately represent features, causing divergence in object features and hindering network learning. In this work, we propose the Anchor Adaptation and Feature Enhancement Strategies (AFS) to alleviate the above two problems. AFS contains two optimized modules: Anchor Adaption RPN Head (A^2RH) and Feature Enhanced Attention Module (FEAM). Specifically, A^2RH performs anchor adaptive learning by establishing a new anchor bias learning branch from the feature map, enabling higher-quality positive and negative sample assignments in RPN. FEAM introduces global features and mask attention based on FPN, and presents Gaussian mask supervision for attention to obtain stronger feature representation. Experiments show that our method improves the average precision by 1.8% on the baseline model, and achieves state-of-the-art results on AI-TOD dataset. Moreover, validation on AI-TOD-v2 and VisDrone2019 datasets also confirms the effectiveness of our method. The code will soon be available at https://github.com/gravity-lhg/AFS.

INDEX TERMS Deep learning, aerial images, tiny object detection, anchor adaptation, feature enhancement.

I. INTRODUCTION

Object detection is one of the fundamental tasks in the field of aerial image interpretation. It aims to accurately locate and identify objects that need to be detected in aerial images, such as pedestrians, vehicles, and aircraft, through image algorithms [1]. It is essential in various practical applications such as military reconnaissance, field search and rescue. Recently, with the decrease in cost and increase in the number of high-altitude aircraft such as drones and remote sensing satellites, the number of aerial images has increased exponentially. This has propelled the aerial object detection task into a stage of rapid development.

Nowadays, deep neural networks have demonstrated strong computing representation and learning capabilities and achieved outstanding results in various application scenarios. In the field of aerial object detection, several typical detection networks such as Faster R-CNN [2], RetinaNet [3], YOLO [4], and FCOS [5] have been adapted and utilized, leading to remarkable detection performance. However, there is still a challenge in detecting tiny objects in aerial images. The reason is that most detectors are designed for regular-sized objects. When encountering tiny objects that lack structural details due to their small size, fixed anchors and common feature representations impede network learning.

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

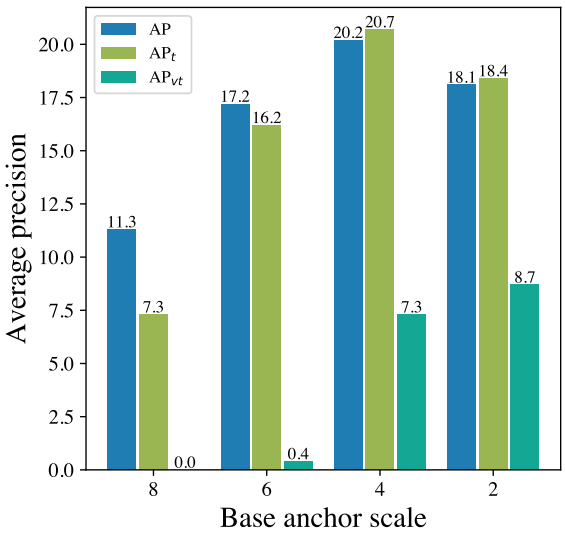


FIGURE 1. The average precision of different base scales anchor on AI-TOD dataset. Faster R-CNN is used as the detector. The base anchor scale is a hyper-parameter used to generate fixed regular anchors. *t* and *vt* represent *tiny* and *very tiny* respectively, which are more detailed classifications of object size by AI-TOD.

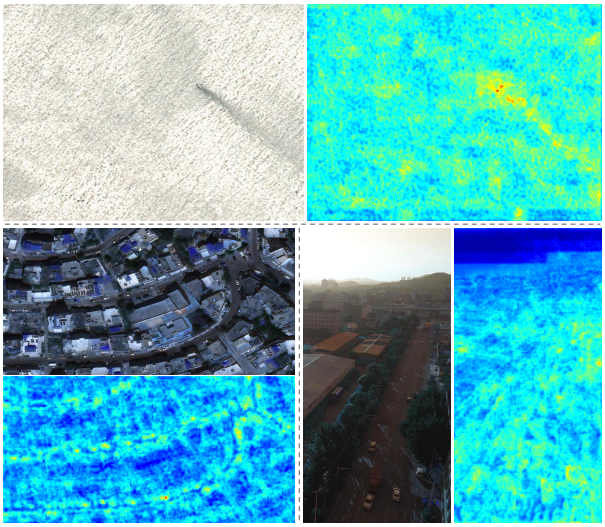


FIGURE 2. Visualization of some sample features in AI-TOD dataset. Faster R-CNN is used as the detector, and the heatmap is obtained by the first layer features output by FPN after channel dimension max pooling.

Specifically, the sensitivity of the intersection and union (IoU) to tiny objects [6], [7], [8] causes the detection network with fixed anchor cannot perform high-quality assignment of samples in Region Proposal Network (RPN). Moreover, since the tiny object has fewer visual features, it is easy to diverge during the feature representation process. Currently, Some works are considered from a metric perspective and enhance detection performance of detector by designing metrics suitable for detecting tiny objects, such as DotD [6], DDR [7], and NWD [8]. There are also some works focused on enhancing the feature representation capability by employing specific strategies to fuse feature maps from different layers, such as Feature Pyramid Network (FPN) [9] and some of its variants PAFPN [10], BiFPN [11], etc.

Our work focuses on the detection of tiny objects in aerial images. As shown in Figure 1. Based on Faster R-CNN, we found through experiments that the presetting of anchor extremely impacts the detection performance of tiny objects. On AI-TOD [12] dataset, we set the base anchor scale to 8, 6, 4, and 2 to conduct comparative experiments. The result shows that the Average Precision (AP) and the AP of *tiny* objects increase from 8-scale to 4-scale but decrease from 4-scale to 2-scale, while the AP of *very tiny* objects gradually increases from 8-scale to 2-scale. This is because smaller anchors can provide higher-quality positive and negative samples for tiny objects. However, this may decrease the quality of samples for larger-scale objects. It indicates that fixed anchors are not optimal for detecting tiny objects. Some works, such as GA-RPN [13] and Cascade RPN [14], focus on optimizing anchors for regular-size objects. But their designs are complex and perform poorly when it comes to tiny objects. Besides, as shown in Figure 2. We observed that when the structural information of the object is ambiguous or when the background of the image is complex, it leads to divergence in object features. The lack of noticeable distinction between foreground and background in the feature map hinders the convergence of the detection network. A series of feature enhancement methods, such as FPN [9], PAFPN [10] and BiFPN [11], all improve detection performance by feature fusion. However, none of them approached the problem from the perspective of feature differentiation.

In response to the above existing problems, we proposed effective anchor adaptation and feature enhancement strategies (AFS) for aerial tiny object detection. It is based on an arbitrary anchor-based two-stage detector and includes two optimized modules: Anchor Adaptation RPN Head (A^2RH) and Feature Enhanced Attention Module (FEAM). A^2RH achieves flexible anchors by designing an anchor adaptation strategy on RPN to mitigate the issue of low-quality positive and negative samples. FEAM improves the feature representation of tiny objects by implementing a feature enhancement strategy on FPN to facilitate effective learning of sub-tasks. In A^2RH, anchor adaptation is accomplished by constructing an anchor bias learning branch from feature maps using a standard 2-D convolution and embodied by learned anchors. Learned anchors are obtained by decoding the original anchor with the bias learned by the anchor bias learning branch, and are used for the assignment of samples in RPN. In FEAM, context enhanced module and mask attention module are designed for feature enhancement to alleviate the problem of feature divergence of tiny objects. They are achieved by global average pooling [15] and deformable convolution [16], respectively. Inspired by SCRDet [17], we provide effective 2-D Gaussian mask supervision for attention, helping the network learn stronger feature representation capabilities for tiny objects. Furthermore, in the experimental section, we perform ablation experiments to validate the effectiveness

of the two modules and achieve state-of-the-art results on AI-TOD dataset. Our method also demonstrates performance improvements on other aerial tiny object datasets.

To sum up, the main contributions of our work as follows:

- We propose AFS for anchor-based two-stage object detectors to improve prediction precision on aviation tiny object detection task.
- We present the A^2RH, which appends the anchor bias learning branch to obtain flexible learned anchors to alleviate the problem of low-quality positive and negative samples in RPN.
- We devise the FEAM with context enhancement and mask attention, which is supervised by Gaussian masks to achieve stronger feature representation of tiny objects in aerial images.
- Our method achieves state-of-the-art detection precision on AI-TOD dataset and also shows performance improvements on other aerial tiny object datasets.

II. RELATED WORK

A. OBJECT DETECTION

With the development of object detection technology in recent years, detectors based on neural networks have proliferated. Detectors are classified as one-stage detectors [3], [4], [5] and two-stage detectors [2], [18] according to whether they contain region proposals stage. Depending on whether the anchors are used, they are categorized into anchor-based detectors [2], [3], [19] and anchor-free detectors [5], [20], [21]. Since DETR [22] brought the Transformer [23] into the object detection task in 2020, various detectors [24], [25], [26], [27] based on DETR have also been proposed rapidly and have achieved excellent performance in the object detection task.

Detectors in aerial field are mostly designed based on the aforementioned regular detectors. They adapt to the requirements of object detection in aerial scenes by appending or optimizing certain modules on top of regular detectors. For instance, existing works such as RoI transformer [28], Oriented R-CNN [29], Oriented RepPoints [30], etc., build upon regular detectors like Faster R-CNN, RepPoint, etc. by designing various rotation box learning strategies to achieve high-precision oritented object detection in aerial images. KLD [31] and TRD [32] mitigate the significant scale differences in aerial images by designing a new metric and employing Transformer-based feature aggregation, respectively. However, compared with the above problems, there are few attentions on the challenging problem of tiny object detection in aerial images.

B. ANCHOR OPTIMIZATION

The anchor serves as a shape and position hypothesis, which guides the detectors to locate and classify objects. However, the artificially preset anchors are discretely distributed and lack continuity similar to the distribution of ground truth boxes. This results in low-quality samples, thereby affecting the network's detection precision. GA-RPN [13] utilizes semantic features to guide anchoring and alleviates feature inconsistency through a feature adaptation module. Cascade RPN [14] implements multi-stage proposals refinement on a single anchor at each position by designing adaptive convolutions. Yolov5 [33] uses clustering methods in advance to obtain initial anchors on specific datasets and then applies them to network learning. Nevertheless, the methods mentioned above are primarily designed for objects of regular sizes, strongly tied to the dataset, which lead to poor performance when applied to tiny objects. Our proposed anchor adaptation strategy can mitigate the issue of low-quality sample assignment in two-stage detectors when dealing with tiny objects.

C. TINY OBJECT DETECTION

Due to the limited visual information inherent in tiny objects and the restricted adaptability of detectors to different scenes, regular detectors often exhibit insignificant detection performance on tiny objects. Recently, some works [8], [12], [34] constructed favorable tiny object datasets for aerial images, providing new benchmarks for research. Essentially, most model-wise works attempt to address this issue from two approaches: feature enhancement and assignment metrics improvement.

1) FEATURE ENHANCEMENT

Feature enhancement refers to achieving a stronger feature representation of tiny objects through the design of fusion or alignment methods. FPN [9] is a typical solution for feature enhancement. It achieves scale fusion by up-sampling high-level features with richer semantics and integrating them into low-level features with lower semantics, thereby enhancing feature representation. Besides, several variants have been proposed based on FPN, such as PAFPN [10], BiFPN [11], etc., all of which are further explored on the fusion approachs of inter-layer. Besides, Wu et al. proposed a features and spatial alignment network named FSANet [35], which adjusts interpolation to promote feature alignment by similarly learning the spatial transfer information between adjacent feature maps. Wu et al. [36] proposed a divergent activation module to improve the response intensity of low-response areas and a similarity module to improve feature distribution and suppress background noise. YOLOv5Imprv [37] captures small features by adding a new feature fusion layer with a smaller receptive field in the feature pyramid part of YOLOv5 [33]. MA^2-FPN [38] promotes the aggregation of tiny object features through large-kernel convolution and hierarchical mask mechanism. In this work, we propose a feature enhancement strategy to obtains more essential feature representation through mask attention supervision.

2) ASSIGNMENT METRIC IMPROVEMENT

Assignment metric improvement is aimed at addressing the issue of low-quality sample assignment in anchor-based detectors when encountering tiny objects. Due to the

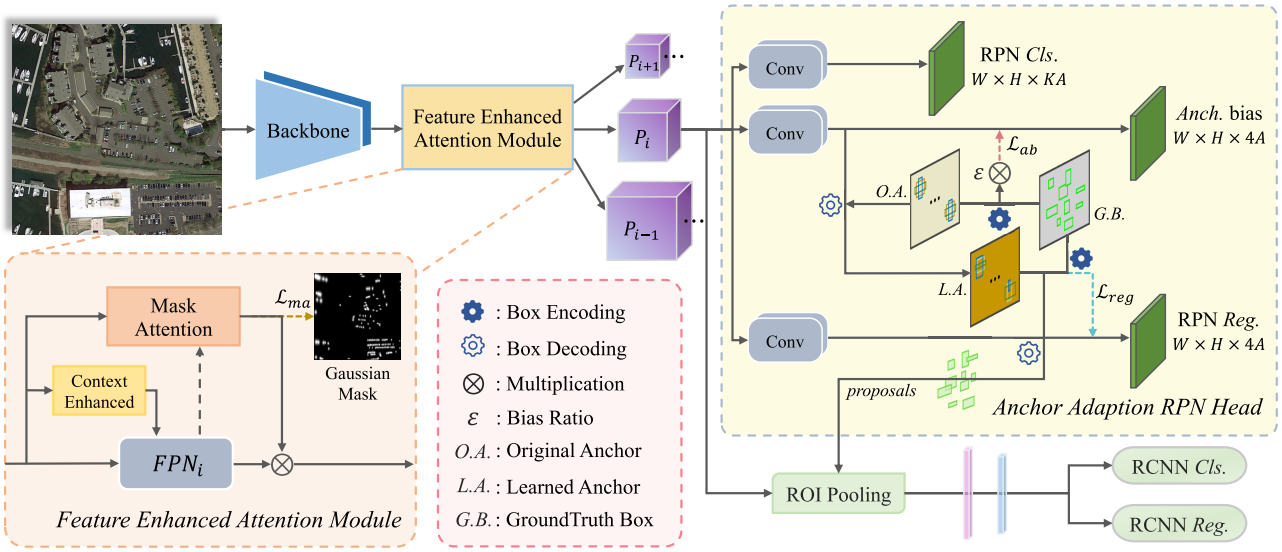


FIGURE 3. The network structure of AFSNet, which is optimized from Faster R-CNN based on our proposed AFS for tiny object detection in aerial images, includes A^{2}RH and FEAM. In A^{2}RH, the anchor bias learning branch provides the anchor with adaptive capabilities and obtains learned anchors. The regression branch predicts the offset from learned anchors to ground truth. In FEAM, features are fed for context enhancement and mask attention to obtain stronger feature representation.

sensitivity of the IoU-based metric to position deviation, subtle position perturbations under a specific threshold may cause sample label inversion, which is very detrimental to stable network learning. DotD [6] simply uses the center points distance between bounding boxes and ground truth boxes to obtain higher-quality samples. NWD-RKA [8] introduces the normalized Wasserstein distance, focusing on more entire bounding box information than DotD. Moreover, it uses the top-K mechanism, which alleviates the problem of inflexible sample assignment. DDR [7] adopts the idea of normalization and uses the line segment ratio metric instead of the area ratio metric, which reduces the sensitivity without introducing any hyper-parameters. However, the above methods all focus on optimizing metrics and do not consider the adverse effects of suboptimal anchors on detector performance. RFLA [39], proposed replacing anchors with Gaussian receptive fields and improved the region proposal capability for tiny objects through hierarchical label assignment. But it requires complex calculation of the receptive field of each layer and discards natural normalized metrics such as IoU-based metric. In this work, we propose an anchor adaptation strategy to mitigate the issue of low-quality sample assignment in RPN by optimizing anchors.

III. METHODOLOGY

A. OVERVIEW

In this work, the two-stage detector Faster R-CNN [2] is selected as the main baseline model. This network is divided into four parts: backbone, neck, RPN, and R-CNN. For the fixed anchor limitations and poor representation ability of the network for tiny objects, we optimize the Faster R-CNN network by AFS, called AFSNet, shown in Figure 3. The optimization is in two parts. First, we design A^{2}RH based on RPN. It eliminates the fixity of preset anchors, making anchors more adaptive for each scale object. Second, we present FEAM based on FPN, which introduces context enhanced module and mask attention module to get stronger feature representation ability for the network. See the following two subsections for specific details.

B. ANCHOR ADAPTION RPN HEAD

1) REGION PROPOSAL NETWORK

The Region Proposal Network [2] (RPN) is a key component in the two-stage object detector, and its main task is to generate candidate object regions to provide effective suggestions in the second stage of object classification and bounding box regression. RPN generates a set of anchor boxes $\mathbb{A}$ in the image dimension based on the downsampling rate of each layer's feature map. Each anchor box $\boldsymbol{a}$ is represented by a 4-tuple form as $\boldsymbol{a} = (a_x, a_y, a_w, a_h)$, where x, y represents the center of the box, w and h are the length and width of the box, respectively. The regression branch aims to predict the transformation $\boldsymbol{\delta}$ from the anchor box $\boldsymbol{a}$ to the ground truth box $\boldsymbol{t}$, which is implemented through the encoding as follows:

$$\begin{aligned} \delta_x &= (t_x - a_x)/a_w, \quad \delta_y = (t_y - a_y)/a_h, \\ \delta_w &= log(t_w/a_w), \quad \delta_h = log(t_h/a_h). \end{aligned} \tag{1}$$

Here, the regression branch $\boldsymbol{f}$ takes image features $\boldsymbol{x}$ as input and outputs predictions $\hat{\boldsymbol{\delta}} = \boldsymbol{f}(\boldsymbol{x})$ to minimize the bounding box loss:

$$\mathcal{L}_{reg}(\hat{\boldsymbol{\delta}}, \boldsymbol{\delta}) = \sum_{k \in \{x,y,w,h\}} L_1(\hat{\delta}_k - \delta_k), \tag{2}$$

where $L_1(\cdot)$ means L_1 loss. According to the decoding box process of inverse transformation of (1), the regression anchor

FIGURE 4. Stochastic visualization of original and learned anchors. The original anchor has strong regularity (first row), and the learned anchor is more flexible and can adapt to the tiny instance (second row). Notice that the visualization is performed on a small random portion of anchors for a clear view.

box can be simply inferred as follows:

$$a'_x = \hat{\delta}_x a_w + a_x, \quad a'_y = \hat{\delta}_y a_h + a_y,$$
$$a'_w = a_w \cdot \exp(\hat{\delta}_w), \quad a'_h = a_h \cdot \exp(\hat{\delta}_h). \tag{3}$$

where $\exp(\cdot)$ represents the exponential function. Then, the set of regression anchors $\mathbb{A}' = \{\boldsymbol{a'}\}$ is filtered by non-maximum suppression (NMS) to produce a sparse set of proposal boxes $\mathbb{P}$:

$$\mathbb{P} = NMS(\mathbb{A}', \mathbb{S}), \tag{4}$$

where $\mathbb{S}$ is the set of proposal box confidences learned by the classification branch.

2) DESIGN OF ANCHOR ADAPTION

We design A^{2}RH based on RPN, which adopt an anchor adaptation strategy to alleviate the problem of low detection performance for tiny objects caused by fixed anchors. Its idea is to generate learned anchors, which can assign higher quality positive and negative samples to tiny objects by learning a certain bias toward the ground truth from the original anchor. The learned anchor participates in the loss calculation and region proposal of the regression branch as a formal anchor. As shown in Figure 3, we introduce the anchor bias learning branch in RPN to generate learned anchors. The input of this branch is the feature of each layer, and the output is the bias of each anchor of the layer, which supervised by a trend bias of ground truth. Specifically, the original anchor, the learned anchor, and the ground truth are represented as $\boldsymbol{a}$, $\boldsymbol{l}$, and $\boldsymbol{t}$, respectively. The anchor bias learning branch predicts the transformation $\boldsymbol{\vartheta}$ from $\boldsymbol{a}$ to $\boldsymbol{l}$, which can be expressed as follows:

$$\vartheta_x = (l_x - a_x)/a_w, \quad \vartheta_y = (l_y - a_y)/a_h,$$
$$\vartheta_w = log(l_w/a_w), \quad \vartheta_h = log(l_h/a_h), \tag{5}$$

where x, y, w, h represents the box's center coordinate, width and height. Similar to the regression branch, anchor bias learning branch $\boldsymbol{f'}$ takes the image feature $\boldsymbol{x}$ as input and outputs the prediction $\hat{\boldsymbol{\vartheta}} = \boldsymbol{f'}(\boldsymbol{x})$ for bias optimization. Since bias learning has a trend towards the ground true, we use $\boldsymbol{\varepsilon}$ times $\boldsymbol{\delta}$ to approximate the target $\boldsymbol{\vartheta}$ for supervised anchor bias stable optimization learning, expressed as fellow:

$$\mathcal{L}_{ab}(\hat{\boldsymbol{\vartheta}}, \boldsymbol{\vartheta}) = \mathcal{L}(\hat{\vartheta}, \varepsilon\delta) = \sum_{k\in\{x,y,w,h\}} L_1(\hat{\vartheta}_k - \varepsilon\delta_k), \tag{6}$$

where $\mathcal{L}_{ab}$ is the loss calculation function for anchor bias. $\boldsymbol{\delta}$ is the transformation from the original anchor to the ground truth (see Section III-B1). $\boldsymbol{\varepsilon}$ is the bias rate of the anchor, and its value range is [0,1]. When $\boldsymbol{\varepsilon}$ is 0, A^{2}RH degenerates into an RPN head, and when $\boldsymbol{\varepsilon}$ is 1, the proposal boxes are completely contributed by the anchor bias learning branch. We set $\boldsymbol{\varepsilon}$ to 0.5 through ablation experiments to obtain the best detection results.

Then the learned anchor is used to replace the original anchor to calculate the transformation $\boldsymbol{\delta^*}$ to the ground truth. The $\boldsymbol{\delta^*}$ is expressed as:

$$\delta^*_x = (t_x - l_x)/l_w, \quad \delta^*_y = (t_y - l_y)/l_h,$$
$$\delta^*_w = log(t_w/l_w), \quad \delta^*_h = log(t_h/l_h). \tag{7}$$

Likewise, the minimum bounding box loss for regression branch prediction $\hat{\boldsymbol{\vartheta}}$ is as follows:

$$\mathcal{L}^*_{reg}(\hat{\boldsymbol{\delta}}, \boldsymbol{\delta^*}) = \sum_{k\in\{x,y,w,h\}} L_1(\hat{\delta}_k - \delta^*_k). \tag{8}$$

where $\mathcal{L}^*_{reg}$ represents the new bounding box regression loss. The other parts of A^{2}RH remain the same as RPN. As shown in Figure 4, the anchor can adapt to tiny objects based on image information and obtain more effective samples after adopting the anchor adaptation strategy. Follow-up experimental results also show that this strategy can improve network detection accuracy effectively.

C. FEATURE ENHANCED ATTENTION MODULE

1) MODULE STRUCTURAL DESIGN

In order to alleviate the problem of feature divergence to tiny objects, we established a feature representation

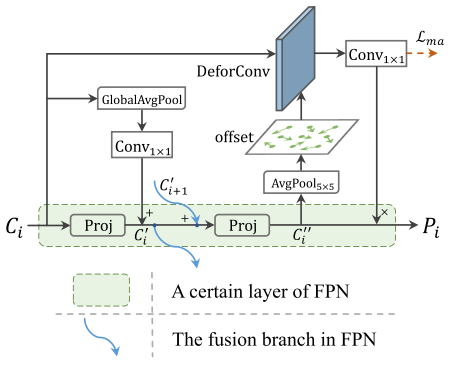
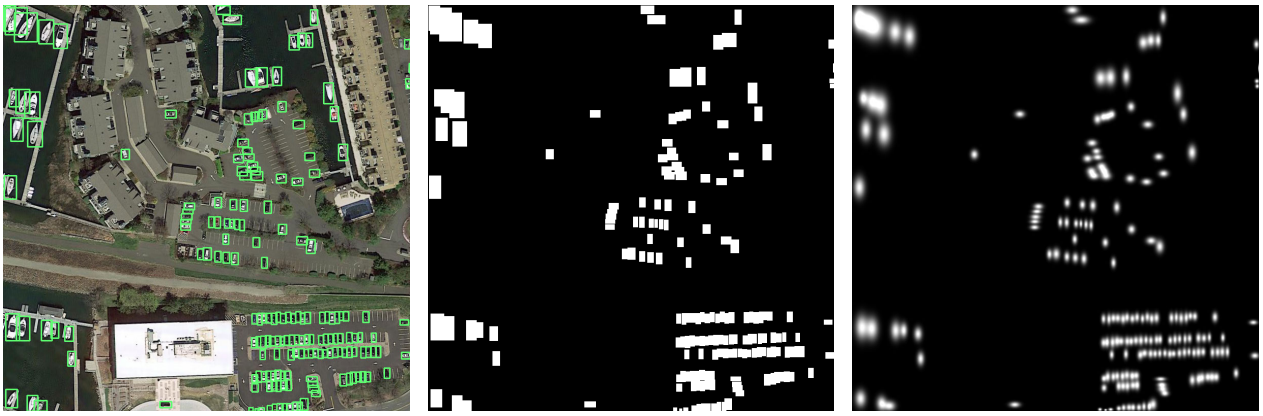


FIGURE 5. Structure of the FEAM, C_i and P_i are the input and output feature maps of the *i-th* layer of FPN, C'_i and C''_i are the middle feature maps in FEAM. "Conv", "DeforConv", "GlobalAvgPool", "AvgPool" and "Proj", respectively represent conventional convolution, deformable convolution, global average pooling, conventional average pooling and projection operation.



(a) GroundTruth Boxes (b) Binary Mask (c) Gaussian Mask

FIGURE 6. Illustration of different masks. (a) is the visualization of the ground truth on the picture, (b) is the Binary mask corresponding to the ground truth of the image, the value is either 0 or 1. (c) is the Gaussian mask corresponding to the ground truth of the image, which value is taken as [0∼1]. In the mask image, black represents the background area and white represents the foreground area.

capability enhancement module based on FPN [9], called FEAM. It comprises two sub-modules: context enhanced and mask attention (see Figure 3). The context enhanced module introduces richer contextual information by globally coordinating features; the mask attention module learns effective mask attention through mask supervision to obtain stronger feature representation. The detail of the module design is shown in Figure 5. In the context enhanced module, we use global average pooling and point convolution to globally coordinate the features of each layer output by the backbone and then add the feature after the first projection layer of FPN. It can be expressed as follows:

$$C_i' \leftarrow proj(C_i) \oplus Conv_{1\times1}(GlobalAvgPool(C_i)), \quad (9)$$

where C_i represents the input feature map, C_i' represents the output feature map by the context enhanced module, $Proj(\cdot)$, $Conv_{1\times1}(\cdot)$ and $GlobalAvgPool(\cdot)$ represent the FPN projection layer, point convolution and global average pooling respectively. In the mask attention module, we use deformable convolution and point convolution to generate mask attention, and then dot-multiply it on the features output by FPN. They can be expressed as follows:

$$C_i'' \leftarrow proj(C_i' \oplus C_{i+1}'), \quad (10)$$

$$offset \leftarrow Conv_{1\times1}^{c=18}(AvgPool_{5\times5}(C_i'')), \quad (11)$$

$$\widehat{MA}_i \leftarrow Conv_{1\times1}^{c=1}(DeforConv(C_i, offset)), \quad (12)$$

$$P_i \leftarrow C_i'' \otimes \widehat{MA}_i, \quad (13)$$

where C_i'' represents the original output feature map of FPN, which obtained by adding the middle feature map C_{i+1}' of the previous layer and C_i' of the current layer through a projection layer. P_i is the output feature map of FEAM. $DeforConv(\cdot)$ represents deformable convolution, and its input is c_i and offset. The offset is obtained by C_i'' through $AvgPool_{5\times5}$ and $Conv$, $AvgPool_{5\times5}(\cdot)$ represents conventional average pooling with a kernel size of 5. $\widehat{MA}_i$ represents the mask attention map obtained by FEAM. It is noteworthy that the c in upper right corner of $Conv$ is the dimension of the convolution output. If not indicated, it means that the input and output dimensions of $Conv$ are the same. Besides, mask attention is supervised by the ground truth mask, which is inspired by SCRDet [17].

2) SUPERVISION FOR MASK ATTENTION

To effectively eliminate the interference of background noise and focus on feature learning for the objects that require attention, we introduced mask supervision for the mask attention module and further improved it. Generally, the binary mask is a common mask form of supervision strategy. It uses two forms, either 1 or 0, to represent the spatial distribution information of the object and background in the image. As shown in Figure 6 (b), where 1 (white) denotes the object and 0 (black) denotes the background, all object area information in the mask is obtained from the corresponding ground truth. However, we found that the detection accuracy gain brought by the binary mask is not significant through experiments. This is likely caused by the introduction of too much background noise in the object boundary area, which seriously affects the feature representation of tiny objects that originally lacked structural information. Therefore, we make improvement to the supervised mask and propose the probability-based 2-D Gaussian mask. It converts the binary mask corresponding to each ground truth into a Gaussian mask through the following formula:

$$\boldsymbol{f}(\boldsymbol{X_j} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{\exp\left[-\frac{1}{2}(X_j-\mu)^T\Sigma^{-1}(X_j-\mu)\right]}{2\pi\,|\Sigma|^{1/2}}, \quad (14)$$

where $\boldsymbol{f}(\boldsymbol{X_j} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \in [0, 1]$. $\boldsymbol{X_j}$ is a vector (x, y) representing j-th 2-D coordinate within the ground truth. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean vector and co-variance matrix of the Gaussian distribution. Therefore, the ground truth mask MA can be

expressed as follows:

$$MA_i' = \sum_{j=1}^{M} (I_i^0 \Leftarrow f(X_j \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})), \tag{15}$$

$$MA \leftarrow MA'[MA' > 1] = 1, \tag{16}$$

where MA_i is *i-th* ground truth mask, MA and $\widehat{MA}$ have the same dimensions and size. M is the number of ground truth for each image. I_i^0 is a template initialized to 0 of the same size as MA_i, and $(I_i^0 \Leftarrow f(X_j \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}))$ means placing the conversion result of each ground truth at the position corresponding to I_i^0. In order to deal with the overlapping of ground truth boxes, we use a selection mechanism to set the mask value ≥ 1 to the maximum value of 1 that the mask should have. Thus, the loss function for optimizing mask attention can be expressed as:

$$\mathcal{L}_{ma} = \frac{1}{N} \sum_{i=1}^{N} CEL(\hat{m}_i, m_i), \tag{17}$$

where $\mathcal{L}_{ma}$ represents the loss of mask supervision, $CEL(\cdot)$ denotes Cross Entropy Loss, N is the number of all mask samples, $\hat{m}_i$ and m_i denote the probability of the predicted attention mask and the ground truth mask respectively. Furthermore, because the ground truth is a horizontal box, it can be modeled as a 2-D Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with

$$\boldsymbol{\mu} = \begin{bmatrix} cx \\ cy \end{bmatrix}, \qquad \boldsymbol{\Sigma} = \boldsymbol{\gamma} \cdot \begin{bmatrix} \frac{w^2}{4} & 0 \\ 0 & \frac{h^2}{4} \end{bmatrix}, \tag{18}$$

where (cx, cy), w, and h indicate the center coordinates, width, and height of the ground truth box, respectively. It is noteworthy that during the experiment, we added an adjustment coefficient $\boldsymbol{\gamma}$ in $\boldsymbol{\Sigma}$ to adjust the size of the Gaussian mask to obtain a more effective supervision mask. After the ablation experiment, we set the $\boldsymbol{\gamma}$ in the experiment to 0.8, and the corresponding Gaussian mask visualization is shown in Figure 6 (c). It can be seen that compared with the binary mask, the Gaussian mask distinguishes instances more obviously and weakens the noise feature contribution of the edge part of the object.

D. LOSS FUNCTION IN TRAINING

Any anchor-based two-stage detector based on AFS use multi-task loss to train in an end-to-end manner. The specific expression can be expressed as follows:

$$\mathcal{L}_{det} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{reg}^* + \lambda_2 \sum_{\tau=1}^{C} \mathcal{L}_{ma}^{\tau} + \lambda_3 \mathcal{L}_{ab}, \tag{19}$$

where $\mathcal{L}_{det}$, $\mathcal{L}_{cls}$, $\mathcal{L}_{reg}^*$, $\mathcal{L}_{ma}$ and $\mathcal{L}_{ab}$ are the total detector loss, classification loss, bounding box regression loss, mask supervision loss, and anchor bias loss, respectively. In this work, cross-entropy loss is used to $\mathcal{L}_{cls}$ and $\mathcal{L}_{ma}$ for classification and mask supervision, and L_1 or $SmoothL_1$ [40] loss is used to $\mathcal{L}_{reg}^*$ and $\mathcal{L}_{ab}$ for bounding box regression and anchor bias learning. They are balanced by λ_i, and λ_1, λ_3 are set to 1 and λ_2 is set to 2. In addition, τ is the index of layers where the feature is located, and C is the total layers involved in mask supervision. The maximum of C can be 5. Finally, C is set to 2 to trade-off the performance of detector in this work.

IV. EXPERIMENT

A. DATASETS

1) AI-TOD

AI-TOD [12] is a challenging dataset proposed to advance research in the detection of tiny objects in aerial images. Specifically, this dataset comprises 28,036 aerial images, encompassing 8 categories and 700,621 object instances. AI-TOD dataset is generated by regular sampling from the DOTA [41] and VisDrone [34] datasets, incorporating a significant amount of remote sensing and drone optical images, aligning better with the characteristics of aerial imagery. Moreover, the average size of objects in AI-TOD is 12.8 pixels, with over 80% of instances having an average size of less than 16 pixels, making it more suitable for research in the detection of tiny objects when compared to the DOTA dataset, which has an average object size of 53.3 pixels. Additionally, AI-TOD dataset is partitioned into training, validation, and test sets in a 4:1:5 ratio, facilitating consistent training and testing comparisons.

2) AI-TOD-V2

AI-TOD-v2 [8] is an enhanced version of AI-TOD dataset, maintaining the same number of images and average instance sizes as AI-TOD. It alleviates issues such as missing annotations and positional errors in the dataset, promoting more reliable network training and resulting in an approximately 1% improvement in mAP on the test set.

3) VISDRONE2019

VisDrone2019 [34] dataset was collected by the AISKYEYE team from the Tianjin University Machine Learning and Data Mining Laboratory. It aims to facilitate the rapid deployment of camera-equipped drones in various application fields such as agriculture, aerial photography, rapid delivery, and surveillance. The benchmark dataset comprises 288 video clips, totaling 261,908 frames, and 10,209 still images. These were captured by diverse drone cameras, encompassing a broad spectrum of locations, environments, objects, and density.

B. IMPLEMENTATION DETAILS

In this work, all experiments are conducted using the PyTorch deep learning framework and the MMDetection [42] library. Training and inference tests were performed on a computer equipped with a single NVIDIA RTX 3090 GPU. Specifically, we fine-tune our models using a ResNet-50 [43] backbone pretrained on ImageNet [44]. We adopt the 1x learning strategy from MMDetection, which entails a total training process of 12 epochs, with an initial learning rate set to 0.005. Learning rate decay was applied at the

TABLE 1. Comparative experimental results on the learning way of anchor on AI-TOD dataset. Bold denotes the best result for each AP.

Method	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
Baseline	20.2	48.9	12.7	7.2	20.7	**26.7**	32.7
Learnable Tensor	18.3	44.9	11.6	5.0	18.0	24.9	33.8
+ bias supervision	19.1	46.2	12.6	5.9	18.8	25.2	**33.9**
Conv from Feature	20.1	48.6	13.1	6.2	20.2	26.3	33.8
+ bias supervision	**20.9**	**50.4**	**13.4**	**8.3**	**21.0**	26.1	32.2

TABLE 2. Comparative experimental results on the learning attributes of anchors on AI-TOD dataset. Bold denotes the best result for each AP.

Method	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
Baseline	20.2	48.9	12.7	7.2	20.7	26.7	**32.7**
Scale	20.3	48.7	13.0	6.9	20.7	**26.9**	32.6
Center	20.5	50.1	13.0	7.6	20.9	26.4	32.3
Width & Height	20.6	50.1	13.1	8.0	20.8	26.1	32.5
All of the above	**20.9**	**50.4**	**13.4**	**8.3**	**21.0**	26.1	32.2

TABLE 3. Anchor bias ratio ablation experiments. ε is the bias amplitude of the learned anchor from original anchor to the ground truth. Bold denotes the best result for each AP.

ε	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
1.0	20.9	50.4	13.4	8.3	21.0	26.1	32.2
0.7	21.0	50.3	13.5	7.7	21.5	26.2	31.9
0.6	21.2	**51.0**	13.6	8.3	**21.7**	26.9	32.5
0.5	**21.3**	**51.0**	**13.7**	8.5	21.5	**27.1**	**33.2**
0.4	20.7	50.2	13.3	**9.2**	21.2	26.2	31.6
0.0	20.2	48.9	12.7	7.2	20.7	26.7	32.7

8^{th} and 11^{th} epoch with a decay factor of 0.1. In the early training phase, a warm-up is set to 500 iterations with a warm-up ratio of 0.001. All experiments employ stochastic gradient descent (SGD) as the optimizer, with a momentum parameter of 0.9, a weight decay parameter of 0.0001, and a batch size of 2. During inference, we utilize a predefined threshold of 0.05 to filter out background boxes and apply non-maximum suppression (NMS) with a threshold of 0.5 to generate the top 3000 bounding boxes ranked by confidence. All training on AI-TOD dataset in this work is on *trainval set*, and verification is on *test set*. The aforementioned training strategy, inference strategy and parameters are applied consistently across all experiments unless otherwise specified.

C. EVALUATION METRICS

All experiments in this work are evaluated using the Average Precision (AP), which is associated with the IoU. Generally, the value of AP can vary significantly at different IoU thresholds. Following the COCO [45] standard, AP is calculated as the mean over IoU values sampled within the range of 0.5 to 0.95 with a 0.05 interval. AP_{50} represents the AP value at the IoU of 0.5, and AP_{75} represents the AP value at the IoU of 0.75. AP_s, AP_m, AP_l represent AP values in different object size ranges: [2, 32], [32, 96], [96, $+\infty$], where $+\infty$ means positive infinite. Notably, the COCO standard can not adequately reflect the detection performance of tiny objects. Therefore, AI-TOD provides a more detailed breakdown of object sizes: [2, 8], [8, 16], [16, 32], [32, $+\infty$], referred to as *very tiny*, *tiny*, *small*, *medium*, corresponding to AP_{vt}, AP_t, AP_s, AP_m, respectively. In this work, all evaluations are based on the finer-grained standards of AI-TOD [12]. Meanwhile, we also conducted statistics on the computational cost (FLOPs), number of parameters (#Params), and inference speed in Frames Per Second (FPS) for each detection network to fully demonstrate the overall detection performance of each network.

D. ABLATION STUDY

1) THE LEARNING WAYS OF ANCHOR

In A^2RH, the learning form and attributes of the anchor represent two modes of anchor adaptation. The learning form of the anchor refers to whether the anchor's bias is learned from a bunch of randomly initialized parameters or obtained from features through convolution. In Table 1, ''Learnable Tensor'' denotes the initialization parameters as Tensor, and ''Conv from Feature'' denotes establishing a convolution branch in RPN. They all end up with a bias set of the same length as the anchor set, and anchor adaptation is achieved by decoding the original anchor box. The ''Conv from Feature'' mode is significantly better than the ''Learnable Tensor'' mode but is still slightly inferior to the baseline. Therefore, we added bias supervision for effective learning of anchor adaptation. The supervision information of the bias supervision uses the offset from the original anchor to the ground truth. The results show that bias supervision is effective and improves 0.8%AP. This work adopted the form of ''Conv from Feature'' + bias supervision. In addition, the anchor's learning properties refer to the anchor's scale, center, length, and width. In Table 2, ''Scale'', ''Center'', ''Width & Height'' and ''All of the above'' represent individual and joint attribute learning, respectively. We performed comparative experiments and observed that learning different attributes yields certain performance gains. Moreover, learning all attributes results in the most significant improvement. This work adopts all attributes of anchor for adaptation.

2) ANCHOR BIAS RATIO

In anchor bias learning, bias supervision is a very important component. It can provide a certain directionality for anchor learning and make anchor adaptation stable and effective. Initially, we used the offset from the original anchor to the ground truth for supervision. However, this way will weaken the ability of the regression branch, causing the anchor bias learning branch to take on more offset learning. This is inconsistent with the original intention of the anchor bias learning to serve only as an auxiliary branch. To balance the capabilities of the two branches and make the anchor bias learning branch gain adaptability toward the ground truth. We introduce the anchor bias ratio parameter $\boldsymbol{\varepsilon}$ and multiply it by the bias from the original anchor to the ground truth to obtain the supervision information of the anchor bias learning branch. In Table 3, we found that when the offset learning of

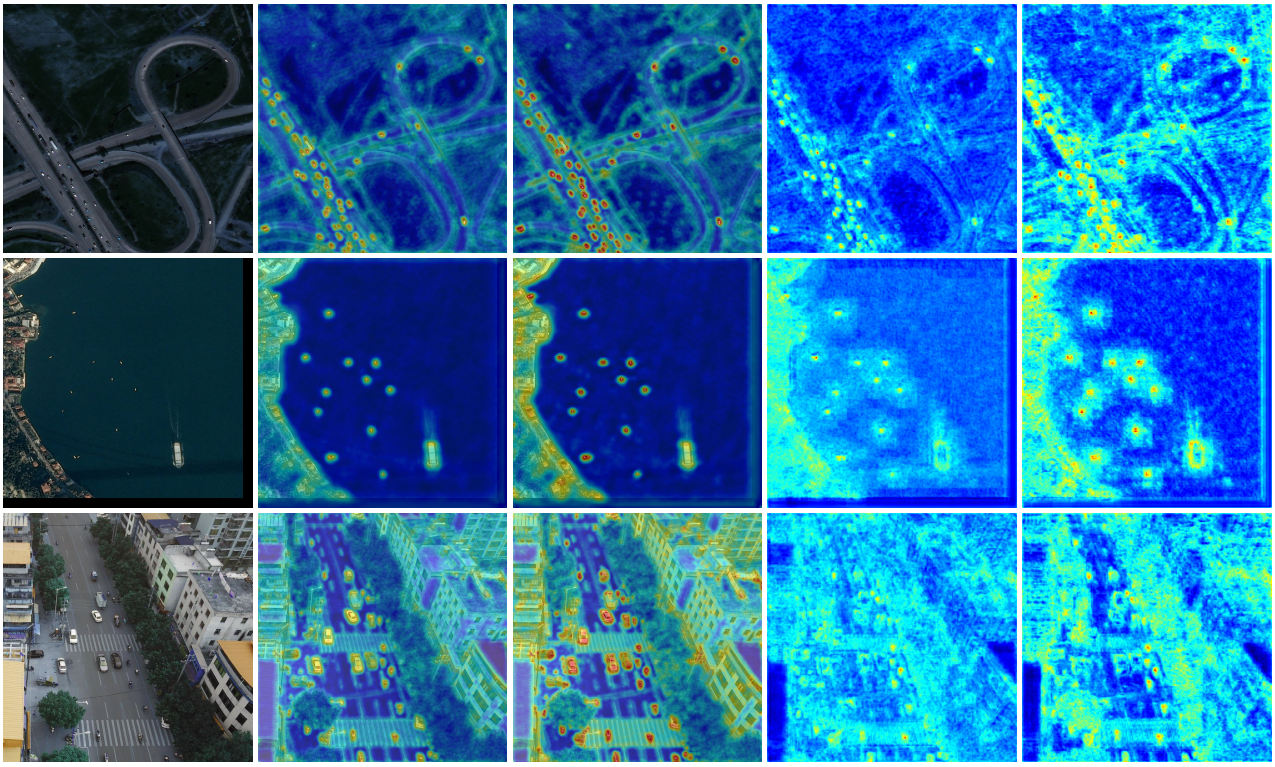


FIGURE 7. Comparative analysis visualization. Includes attention visualization using Binary Mask Supervision (second column) and Gaussian Mask Supervision (third row), feature visualization of the baseline model (fourth column), and baseline model with FEAM. Feature visualization is obtained by the first layer features output by FPN after channel dimension max pooling.

TABLE 4. Results of Gaussian mask factor ablation experiments. γ is the factor of the Gaussian mask co-variance matrix. They are conducted based on the baseline with A^{2}RH. Bold denotes the best result for each AP.

Method	γ	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
Binary Mask	–	21.3	51.2	13.6	7.8	22.0	26.9	32.6
Gaussian Mask	1.5	21.6	52.0	14.1	7.3	22.2	27.2	32.8
	1.0	21.8	51.5	14.6	7.7	22.5	26.8	32.0
	0.8	**22.0**	**52.5**	**14.8**	**8.6**	**22.8**	**28.0**	**33.1**
	0.6	21.9	52.3	14.5	8.0	22.5	26.8	32.7

TABLE 5. Ablation study of each module in the proposed AFS on AI-TOD dataset. A^{2}RH means Anchor Adaption RPN Head and FEAM means Feature Enhanced Attention Module. Bold denotes the best result for each AP.

A^{2}RH	FEAM	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
		20.2	48.9	12.7	7.2	20.7	26.7	32.7
✓		21.3	51.0	13.7	8.5	21.5	27.1	**33.2**
	✓	21.4	51.3	13.9	7.9	22.2	26.8	32.4
✓	✓	**22.0**	**52.5**	**14.8**	**8.6**	**22.8**	**28.0**	33.1

the two branches is divided equally, ε is 0.5, the maximum accuracy gain of 0.4%AP can be obtained. Finally, the value of ε in this work is set to 0.5.

3) GAUSSIAN MASK FACTOR

Since the performance impact brought by binary mask supervision is not significant, we design a 2-D Gaussian mask through analysis. It is a supervised mask containing probability, which can provide a more effective supervision signal for mask attention. Considering the influence of boundary noise, we introduce the Gaussian mask factor γ, an adjustment factor attached to the Gaussian co-variance matrix that can control the size of the Gaussian mask area. In Table 4, we set γ to 1.5, 1.0, 0.8, and 0.6 for experiments, respectively, and found that their precision is higher than the binary mask method. When γ is 0.8, we achieve a performance improvement of 0.7% AP. As shown in Figure 7, the first column are images. The second and third columns indicate Binary Mask Supervision (BMS) and Gaussian Mask Supervision (GMS). The attention using GMS is stronger than the attention using BMS, which can effectively alleviate the interference problem of noise features. The fourth and fifth columns represent the features before and after using FEAM. It can be seen that after using FEAM, the background noise is relatively less, and the features of the object are more prominent.

4) THE EFFECTIVENESS OF A^{2}RH AND FEAM MODULES

In this part, we apply A^{2}RH and FEAM to the baseline model separately or jointly to verify the effectiveness of each module and the combination of modules. As shown In Table 5, A^{2}RH and FEAM each have a performance improvement of more than 1% AP when used in the baseline model. From the data point of view, A^{2}RH is more friendly to *very tiny* objects, and

TABLE 6. AFS is experimentally compared with a anchor-based two-stage network that also optimizes anchor. Bold denotes the best result for each AP.

Method	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
Faster R-CNN [2]	20.2	48.9	12.7	7.2	20.7	26.7	32.7
GA-RPN [13]	19.0	47.2	11.4	8.1	18.1	24.2	**34.3**
Cascade-RPN [14]	16.5	39.1	10.8	4.1	15.9	22.4	29.5
Faster R-CNN w/AFS (ours)	**22.0**	**52.5**	**14.8**	**8.6**	**22.8**	**28.0**	33.1

FEAM is more friendly to *tiny* objects. Moreover, combining the two modules has a performance superposition, and the total AP increase can reach 1.8%, which means that the two modules have the ability to promote each other and improve detectors performance.

5) COMPARISON OF OTHER TWO-STAGE ANCHOR OPTIMIZATION METHODS

To validate the performance advantages of this work, we compared experimentally with a similarly targeted anchor optimization method for two-stage detectors, including GA-RPN [13] based on guidance boxes and Cascade-RPN [14] based on multi-stage refinement. As shown in Table 6. Since GA-RPN and Cascade-RPN are designed for regular scales, the detection precision on AI-TOD dataset is worse than that of Faster R-CNN. However, we propose that AFS achieves highly adaptable anchors and a stronger feature representation for tiny objects, demonstrating excellent precision improvement in Faster R-CNN with AFS on AI-TOD.

E. COMPARE WITH OTHER METHODS ON AI-TOD

To verify the detection performance of the AFS-based two-stage detector, we conducted several comparative experiments with other detectors on AI-TOD dataset, including some regular detectors, such as RetinaNet [3], RepPoints [46], FCOS [5], YOLO [33], SSD [19], CenterNet [21]. And M-CenterNet [12] and FASNet [35], which are optimized for tiny objects. We also compared the state-of-the-art detectors for tiny objects based on Faster R-CNN, Cascade R-CNN, and DetectoRS metric improvements. In Table 7, entire table is divided into two groups. The first group (upper part) shows the performance of regular object detection networks, while the second group (lower part) presents the performance of anchor-based two-stage detection networks, which were utilized as the base networks in our work.

In the first group, it can be seen that FCOS using the P2 layer has the best performance among regular detectors, especially in *tiny* and *very tiny* scales. M-CenterNet is based on CenterNet and optimizes the center point sampling method. FASNet is a one-stage detector based on FCOS and optimizes the alignment of features and spatial. Both of them work on tiny objects and have made significant performance improvements but still at a low level. In the second group, (s4) indicates the results after parameter tuning for tiny object detection. DotD [6], NWD [8], NWD-RKA [8], and RFLA [39] are relatively outstanding methods in the aviation tiny object detection task in recent years. DotD, NWD, and NWD-RKA are metrics for tiny objects on a two-stage detector to alleviate the problem of low-quality RPN positive and negative sample assignment caused by IoU sensitive to tiny objects. They all have significant performance improvements compared to the baseline with a base anchor scale of 8, but they are all solved from the metric perspective and do not consider the suboptimal anchor issue. RFLA replaces anchors with receptive field calculations to avoid problems caused by IoU. This concept is commendable, but it is complex and requires calculating and converting the receptive field in advance and transforming the box into a 2-D Gaussian representation. For tiny objects, the preset anchors of detectors have a particularly extreme impact on the detector's performance, and the suboptimal anchor is a significant factor influencing the assignment of positive and negative samples in RPN. The AFS we proposed can make the anchor adaptive to tiny objects and enhance the feature representation of the network. On the baseline model with a base anchor scale of 4, our method improved 1.8%, 2.0%, and 1.1% AP on Faster R-CNN, Cascade R-CNN, and DetectoRS, respectively. Looking at the entire results, our method has achieved the best results except for AP_m, especially in *very tiny* and *tiny* scales, which have an improvement of about 1% in AP compared to the ranked second method. Noticeably, due to the complex design of the backbone and neck of the DetectoRS, we only adopted the A^2RH module on it. Nevertheless, it achieved state-of-the-art detection results on the AI-TOD dataset.

As shown on the right side of Table 7, It is apparent that the inference speed of detection networks such as YOLOv5 is particularly fast, but the detection accuracy is too low. On the other hand, the anchor-based two-stage detection networks do not have advantages in terms of computation amount, network parameters, and inference speed, but the detection AP can reach about twice than regular detection network. Furthermore, based on our proposed AFS detection network, while the detection AP is significantly improved, the inference speed has not been greatly reduced, indicating that the AFS strikes a good trade-off between precision and speed. Moreover, Figure 8 displays sample inference visualizations on the AI-TOD dataset. The first row showcases results from the baseline detection network, while the second row presents results from the detection network based on AFS. It is evident that compared to the baseline network, the AFS-based network detects more tiny objects and reduces the number of false negatives (FN). Additionally, it demonstrates better detection performance in oblique scenarios.

F. VALIDITY VERIFICATION OF AFS ON OTHER DATASETS

To verify the generalization ability of our proposed AFS, we conducted comparative experiments on two aerial tiny object datasets, AI-TOD-v2 and VisDrone2019. In Table 8, for AI-TOD-v2, the precision of the detector increases after adopting the AFS. Faster R-CNN, Cascade R-CNN, and DetectoRS have 0.7%, 0.6%, and 0.5% AP improvements,

TABLE 7. Comparison of the proposed AFS with previous state-of-the-art methods on AI-TOD. Training used AI-TOD *trainval set* and validation used AI-TOD *test set*. "FR", "CR", "DR", respectively indicate Faster R-CNN, Cascade R-CNN, and DetectoRS. Bold denotes the best result for each AP. Underline denotes the ranked second result for each AP. P2 represents that the lowest layer of FPN is used. s4 indicates that the base scale of the anchor is 4. – means the data was not obtained. The test image size for FPS is 800 × 800, and experimental settings unchanged.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m	FLOPs	#Params	FPS
RetinaNet [3]	ResNet-50	8.9	21.9	5.1	2.3	8.9	12.5	16.8	129.8G	36.3M	30.0
RetinaNet (P2)		13.4	31.2	9.3	4.8	14.4	17.0	21.7	353.9G	32.2M	22.6
RepPoints [46]	ResNet-50	11.8	28.7	7.8	2.9	13.1	15.4	20.3	119.0G	36.7M	32.8
FCOS [5]	ResNet-50	9.8	24.1	5.9	1.4	8.0	15.1	17.4	123.2G	31.9M	33.9
FCOS (P2)		15.4	35.8	11.2	5.4	17.3	18.9	21.6	339.2G	31.9M	23.8
Faster R-CNN [2]	ResNet-50	11.3	26.7	7.6	0.0	7.3	23.7	33.6	134.4G	41.1M	27.2
YOLOv5 [33]	CSP-DarkNet	11.8	28.7	7.8	2.9	13.1	15.4	20.3	49.1G	21.2M	61.8
Cascade R-CNN [18]	ResNet-50	13.6	31.2	10.5	0.0	10.4	26.5	31.5	162.2G	69.0M	18.6
SSD-512 [19]	VGG-16	7.2	22.1	2.7	1.1	5.9	11.6	14.2	89.8G	27.3M	42.3
DetectoRS [47]	ResNet-50	14.9	33.5	11.6	0.0	11.3	28.1	34.2	159.7G	123.2M	10.0
CenterNet [21]	DLA-34	13.4	39.2	5.0	3.8	12.1	17.8	18.9	58.9G	24.5M	43.8
M-CenterNet [12]		14.5	40.7	6.4	6.1	15.0	19.4	20.4	59.1G	24.5M	43.1
FASNet [35]	ResNet-50	16.3	41.1	9.8	4.4	14.6	23.4	33.3	125.2G	31.9M	–
Faster R-CNN											
FR (s4)	ResNet-50	20.2	48.9	12.7	7.2	20.7	26.7	32.7	134.4G	41.1M	27.2
FR w/DotD [6]		14.6	38.5	9.3	7.2	16.1	17.9	23.7	134.5G	41.1M	27.4
FR w/NWD [8]		17.8	43.8	11.0	2.5	17.0	26.1	34.3	134.5G	41.1M	27.1
FR w/NWD-RKA [8]		19.5	49.2	11.7	8.3	19.6	24.5	31.9	134.5G	41.1M	27.0
FR w/RFLA [39]		20.8	51.2	12.9	6.9	21.1	26.7	32.2	134.5G	41.1M	27.3
FR w/AFS (ours)		22.0	52.5	14.8	8.6	22.8	28.0	33.1	136.7G	43.2M	24.8
Cascade R-CNN											
CR (s4)	ResNet-50	21.0	48.7	14.6	6.8	21.6	27.0	32.8	162.2G	69.0M	18.6
CR w/DotD [6]		16.1	39.2	10.6	8.3	17.6	18.1	22.1	162.2G	69.0M	18.8
CR w/NWD [8]		18.7	44.2	12.9	3.6	17.4	26.5	35.6	162.3G	69.0M	18.6
CR w/NWD-RKA [8]		20.5	48.7	13.8	8.1	20.6	25.6	34.0	162.3G	69.0M	18.5
CR w/RFLA [39]		21.9	51.3	15.4	8.2	22.0	27.2	34.7	162.3G	69.0M	18.6
CR w/AFS (ours)		23.0	52.7	16.2	8.2	23.7	28.2	34.9	164.4G	71.0M	16.7
DetectoRS											
DR (s4)	ResNet-50	24.2	54.1	18.1	8.1	25.0	29.9	36.8	159.7G	123.2M	10.0
DR w/DotD [6]		18.9	43.8	13.3	8.8	19.4	27.3	32.5	159.7G	123.2M	10.1
DR w/NWD [8]		20.8	49.3	14.3	6.4	19.7	29.6	38.3	159.7G	123.2M	10.0
DR w/NWD-RKA [8]		23.2	53.5	16.7	8.0	23.3	29.7	**38.6**	159.8G	123.2M	9.9
DR w/RFLA [39]		24.6	55.5	18.0	10.3	24.8	30.2	37.9	159.7G	123.2M	10.1
DR w/A^2RH (ours)		**25.3**	**57.0**	**19.1**	**11.1**	**25.7**	**30.8**	37.8	159.8G	123.2M	9.9

FIGURE 8. Visualization of sample test results on AI-TOD dataset, comparing the baseline detector (first row) with the AFS-based detector (second row). True positive (TP), false positive (FP), and false negative (FN) predictions are represented by green, blue, and red boxes in images, respectively.

TABLE 8. Comparison of detection results on AI-TOD-v2 dataset. Training used AI-TOD-v2 *train set* and validation used AI-TOD-v2 *val set*. Bold denotes the best result for each AP.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s	AP_m
Faster R-CNN [2]	ResNet-50	22.6	51.4	13.9	4.7	23.3	27.6	37.4
Cascade R-CNN [18]		23.2	53.8	15.5	5.6	22.7	29.2	39.7
DetectoRS [47]		26.0	57.9	18.8	6.8	25.6	**31.5**	43.8
Faster R-CNN w/AFS	ResNet-50	23.3	54.7	14.3	6.7	23.4	28.6	37.4
Cascade R-CNN w/AFS		23.8	54.9	16.5	**7.9**	23.7	29.0	39.4
DetectoRS w/A^2RH		**26.5**	**60.0**	**18.9**	7.4	**26.0**	31.4	**43.9**

TABLE 9. Comparison of detection results on VisDrone2019. The *train* and *val* sets of VisDrone2019 are used for training and validation, respectively.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_{vt}	AP_t	AP_s
Faster R-CNN [2]	ResNet-50	25.5	43.6	25.9	0.4	7.6	22.9
Cascade R-CNN [18]		27.7	45.2	28.9	0.3	8.7	25.6
DetectoRS [47]		28.4	46.4	29.5	0.3	9.2	26.4
Faster R-CNN w/AFS	ResNet-50	$27.6^{+2.1}$	$49.1^{+5.5}$	$26.9^{+1.0}$	$5.5^{+5.1}$	$14.4^{+6.8}$	$24.5^{+1.6}$
Cascade R-CNN w/AFS		$29.8^{+2.1}$	$50.0^{+4.8}$	$30.4^{+1.5}$	$4.4^{+4.1}$	$15.0^{+6.3}$	$26.9^{+1.3}$
DetectoRS w/A^2RH		$30.1^{+1.7}$	$51.2^{+4.8}$	$30.5^{+1.0}$	$4.0^{+3.7}$	$14.3^{+5.1}$	$27.2^{+0.8}$

respectively. In Table 9, for VisDrone2019, the precision of the detector based on AFS is significantly improved. The improvement of Faster R-CNN, Cascade R-CNN, and DetectoRS can reach 2.1% AP, 2.1% AP, and 1.7% AP, respectively. Same as above, DetectoRS only uses A^2RH. Judging from the experimental results, our method effectively improves precision at *tiny* and *very tiny* scales.

V. CONCLUSION

In this work, we propose AFS to mitigate the issues of suboptimal anchors and poor feature representation of tiny objects in aerial images. First, we build the A^2RH based on RPN. By appending the anchor bias learning branch, the originally fixed anchors become flexible and have stronger adaptability, which helps RPN obtain higher-quality samples of tiny objects. Second, we present the FEAM based on FPN, which is implemented by designing Context Enhanced and Mask Attention modules. It uses Gaussian masks for attention supervision and can improve the network's feature representation ability for tiny objects. We have proven the effectiveness of the above two improvements through ablation experiments. Meanwhile, our method improves the baseline AP by 1.8% on AI-TOD dataset and achieves state-of-the-art detection performance. And it also shows performance improvements on the two other aviation tiny object datasets.

REFERENCES

[1] V. Goyal, R. Singh, M. Dhawley, A. Kumar, and S. Sharma, "Aerial object detection using deep learning: A review," in *Comput. Intell., Select Proc. InCITe*, 2022, pp. 81–92.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, Jun. 2015, pp. 1137–1149.

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[4] 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.

[5] 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.

[6] C. Xu, J. Wang, W. Yang, and L. Yu, "Dot distance for tiny object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 1192–1201.

[7] H. Liu, Q. Tong, L. Qi, Y. Hao, and X. Liu, "A double diagonal ratio metric for tiny object detection in aerial images," in *Proc. Int. Conf. Comput. Eng. Distance Learn. (CEDL)*, Jul. 2023, pp. 11–19.

[8] 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.

[9] 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.

[10] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[11] 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.

[12] J. Wang, W. Yang, H. Guo, R. Zhang, and G.-S. Xia, "Tiny object detection in aerial images," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 3791–3798.

[13] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2960–2969.

[14] T. Vu, H. Jang, T. Pham, and C. Yoo, "Cascade RPN: Delving into high-quality region proposal network with adaptive convolution," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 1–7.

[15] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.

[16] 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.

[17] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8231–8240.

[18] 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.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37.

[20] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," *Int. J. Comput. Vis.*, vol. 128, no. 3, pp. 642–656, Mar. 2020.

[21] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6568–6577.

[22] 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.*, 2020, pp. 213–229.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1–6.

[24] 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*.

[25] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection," 2022, *arXiv:2203.03605*.

[26] Z. Zong, G. Song, and Y. Liu, "DETRs with collaborative hybrid assignments training," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 6748–6758.

[27] S. Zhang, X. Wang, J. Wang, J. Pang, C. Lyu, W. Zhang, P. Luo, and K. Chen, "Dense distinct query for end-to-end object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7329–7338.

[28] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning RoI transformer for oriented object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2844–2853.

[29] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3500–3509.

[30] W. Li, Y. Chen, K. Hu, and J. Zhu, "Oriented RepPoints for aerial object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1819–1828.

[31] X. Yang, G. Zhang, X. Yang, Y. Zhou, W. Wang, J. Tang, T. He, and J. Yan, "Detecting rotated objects as Gaussian distributions and its 3-D generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4335–4354, Apr. 2023.

[32] Q. Li, Y. Chen, and Y. Zeng, "Transformer with transfer CNN for remote-sensing-image object detection," *Remote Sens.*, vol. 14, no. 4, p. 984, Feb. 2022.

[33] G. Jocher et al., "Ultralytics/YOLOV5: V7.0—YOLOV5 sota realtime instance segmentation," Zenodo, Nov. 2022. Accessed: Nov. 2023. [Online]. Available: https://zenodo.org/records/7347926

[34] D. Du et al., "VisDrone-DET2019: The vision meets drone object detection in image challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 213–226.

[35] J. Wu, Z. Pan, B. Lei, and Y. Hu, "FSANet: Feature-and-spatial-aligned network for tiny object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Sep. 2022, Art. no. 5630717.

[36] Z.-Z. Wu, J. Xu, Y. Wang, F. Sun, M. Tan, and T. Weise, "Hierarchical fusion and divergent activation based weakly supervised learning for object detection from remote sensing images," *Inf. Fusion*, vol. 80, pp. 23–43, Apr. 2022.

[37] I. Singh and G. Munjal, "Improved YOLOV5 for small target detection in aerial images," *SSRN Electron. J.*, vol. 3, Jul. 2022, Art. no. 4049533.

[38] S. Li, Q. Tong, X. Liu, Z. Cui, and X. Liu, "MA2-FPN for tiny object detection from remote sensing images," in *Proc. 15th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Nov. 2022, pp. 1–8.

[39] C. Xu, J. Wang, W. Yang, H. Yu, L. Yu, and G.-S. Xia, "RFLA: Gaussian receptive field based label assignment for tiny object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 526–543.

[40] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[41] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.

[42] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[45] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. 13th European Conference*, 2014, pp. 740–755.

[46] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9656–9665.

[47] S. Qiao, L.-C. Chen, and A. Yuille, "DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10208–10219.

HAOGUANG LIU received the B.S. degree from the School of Computer Science and Technology, Wuhan Institute of Technology University, in 2020. He is currently pursuing the degree with the Computer School, Beijing Information Science and Technology University, China. His research interests include machine learning and aviation object detection.

QIANG TONG received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, in 2012. Since August 2018, he has been a Lecturer with the Computer School, Beijing Information Science and Technology University, China. His research interests include image recognition, computer vision, and machine learning.

LIN MIAO received the Ph.D. degree from the Ben-Gurion University of the Negev, in 2022. She is currently a Lecturer with Beijing Information Science and Technology University. Her research interests include artificial intelligence, pattern recognition, and knowledge graph.

XIULEI LIU received the Ph.D. degree in computer science from Beijing University of Posts and Telecommunications, in March 2013. He was a Visiting Ph.D. Student with CCSR, University of Surrey, from October 2008 to October 2010. Since January 2022, he has been a Professor with the Computer School, Beijing Information Science and Technology University, China. His research interests include semantic sensor, semantic web, knowledge graph, and semantic information retrieval.