AFDet: Toward More Accurate and Faster Object Detection in Remote Sensing Images

Nanqing Liu, *Graduate Student Member, IEEE*, Turgay Celik, Tingyu Zhao, Chao Zhang, and Heng-Chao Li

***Abstract*—Object detection in remote sensing imagery usually suffers from inaccurate target localization and bounding box regression uncertainty, mainly due to the varying sizes of objects and the complexity of the background. Most detectors address these challenges by adding various feature extraction modules, which increases the size and computational burden of the network. In this article, we propose a more accurate and faster detector named AFDet, which is composed of two parts: a backbone pretrained on ImageNet and a head that includes a center prediction branch (CPB), semantic supervision branch (SSB), and boundary estimation branch (BEB). CPB produces a keypoint heatmap using an elliptical Gaussian kernel to adapt to the ground truth with a large aspect ratio. SSB, which is used only during training, extracts extra keypoint features from boundary and interior points rather than only from the center point, thereby improving the quality of object localization. BEB predicts the distributions of the bounding box in four directions, which is further supervised by the focus loss, and the gather loss raises the box prediction accuracy. To verify the effectiveness and robustness of AFDet, we conduct extensive experiments on three widely used optical remote sensing object detection datasets, i.e., NWPU VHR-10, DIOR, and HRRSD, for which AFDet achieves state-of-the-art results.**

***Index Terms*—Anchor-free method, object detection, optical remote sensing images.**

I. Introduction

OBJECT detection in remote sensing is a fundamental topic in many applications, such as detecting ships, vehicles, and aircraft. As many researchers have great interest in this field, various advanced methods have been proposed in recent years. The traditional object detection methods [1], [2] mainly use classifiers trained on handcrafted image features to distinguish between the foreground (the target) and background. However, they cannot generalize due to the foreground and background variations, which are not well captured by the handcrafted features.

In recent years, as the number of deep learning techniques has exponentially increased, object detectors have become more generalizable than traditional methods. The earliest deep learning-based object detectors, such as the region-based convolutional neural network (RCNN) [3], Fast-RCNN [4], and Faster-RCNN [5], are two-stage methods in which region proposals are generated before the object category, and bounding box location are obtained. Due to a large number of parameters and slow speed of two-stage methods, single-stage anchor-based methods [6]–[8], which can directly predict objects without multistage refinement, have become much more common. However, anchor-based methods excessively rely on anchor designs, which usually depend on the human experience. Moreover, an anchor's size and aspect ratio must constantly change to adapt to different data distributions. To alleviate anchor-related problems, some researchers have proposed anchor-free methods [9]–[11].

Although the algorithms mentioned above can achieve good performance for natural images, employing these methods to detect objects in remote sensing images remains difficult. Compared with natural scene images, the varying sizes of objects and the complexity of the visual appearance in remote sensing images make it difficult to locate the center and regress the object's boundary. To address the challenge of large-scale variation of objects, Cheng *et al.* [12] adopted a cross-scale feature fusion (CSFF) strategy to generate the feature map with one layer of multiscale receptive fields. Huang *et al.* [13] adopted a multilevel feature pyramid [14] and CSFF [12] to obtain more informative features. Xu *et al.* [15] proposed a pseudoanchor proposal module and a context-based feature alignment module to learn adaptive features for objects with a large aspect ratio. Zhao *et al.* [16] proposed channel-wise attention module to fuse features between channels and pixels to obtain a global receptive field and extract more robust features. Wang *et al.* [17] proposed an atrous spatial feature pyramid module to fuse the context information in multiscale features by using feature pyramid and multiple atrous rates. To address the challenge of the complex visual appearance, Fu *et al.* [18] proposed an anchor-free method based on a fully convolutional one-stage object detector (FCOS) [9] using an attention-guided balanced pyramid and a feature-refinement module. Cui *et al.* [19] introduced the spatial shuffle-group enhance attention module into the backbone of CenterNet [10] to suppress inshore and inland interference. Guo

Manuscript received August 18, 2021; revised October 20, 2021; accepted November 4, 2021. Date of publication November 16, 2021; date of current version December 16, 2021. This work was supported by Sichuan Provincial Science and Technology Projects under Grant 2019JDJQ0023 and in part by the National Natural Science Foundation of China under Grant 61871335. *(Corresponding author: Turgay Celik.)*

Nanqing Liu, Tingyu Zhao, and Heng-Chao Li are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China (e-mail: lansing163@163.com; zhaotyu007@163.com; hcli@home.swjtu.edu.cn).

Turgay Celik is with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China, with the School of Electrical and Information Engineering, University of Witwatersrand, Johannesburg 2000, South Africa, and also with the Faculty of Engineering and Science, University of Agder, 4630 Kristiansand, Norway (e-mail: celikturgay@gmail.com).

Chao Zhang is with the Department of Traffic Engineering, Sichuan Police College, Chengdu 610041, China (e-mail: galoiszhang@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2021.3128566

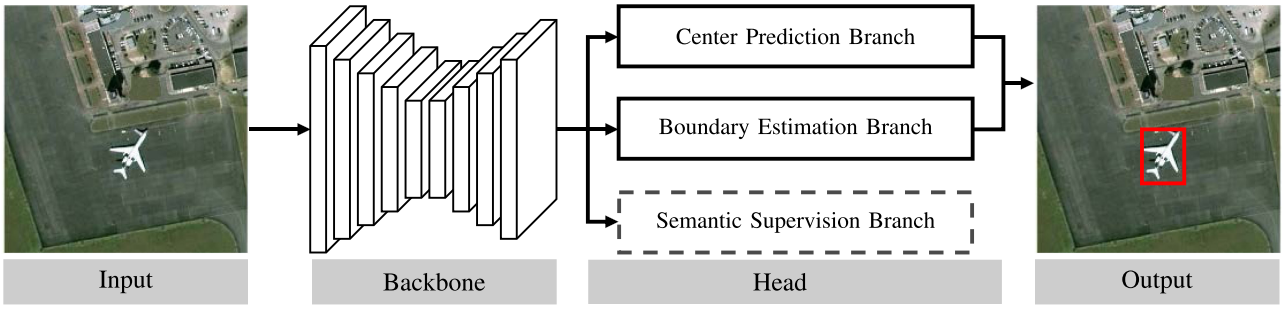

Fig. 1. Overall architecture of AFDet, which contains a backbone and a head. The head consists of the CPB, BEB, and SSB. The CPB and BEB are used to predict the locations of objects and the bounding box estimation parameters, while the SSB is used only during the training of the detector.

et al. [20] designed a feature refinement module, a feature pyramid fusion module, and a head enhancement module to improve the conventional CenterNet for synthetic aperture radar images. Liu *et al.* [21] used a selective gate to acquire reasonable aggregations from different-sized features.

The approaches mentioned above add extra modules to the network or use anchors *a priori*, thereby increasing the computational burden on the detectors. In contrast to these methods, we propose a novel keypoint-based anchor-free detector called AFDet. Our detector does not contain a multiscale prediction architecture or an anchor design and thus does not exhibit a complex predefined setting and network structure. AFDet mainly has a backbone and head, with the head consisting of three branches: the center prediction branch (CPB), boundary estimation branch (BEB), and semantic supervision branch (SSB). The simple representation of the AFDet is shown in Fig. 1. To evaluate our proposed model, we conduct extensive experiments on the NWPU VHR-10 [22], DIOR [23], and HRRSD [24] datasets. The results show that our model performs well on the datasets and outperforms some state-of-the-art object detection approaches.

The main contributions of this article are summarized as follows:

1) We propose an accurate and fast object detector called AFDet, which includes three novel branches named CPB, BEB, and SSB. AFDet can achieve good accuracy while maintaining low computational complexity.
2) In CPB, due to the large aspect ratios of objects in remote sensing images, we apply the elliptical Gaussian kernel (EGK) to encode the training samples. Unlike the conventional circular Gaussian kernel (CGK) [10], EGK can adapt to the shape of objects. Details are provided in Section III-B.
3) In BEB, we treat the bounding box estimation as a classification task rather than a regression task to obtain more fine-grained distance distribution, which is helpful for some objects with blurred boundaries. Moreover, we propose two losses named L_{gather} and L_{focus} to obtain more accurate predictions by controlling the shape of the distribution. The details can be found in Section III-C1.
4) In SSB, we propose to extract semantic keypoints of an object as training targets because the center point usually cannot represent a sufficient number of features. Through more semantic acquisition, the positioning of the center point will be more accurate. The details are provided in Section III-D.

The remainder of this article is organized as follows. In Section II, we first review work related to anchor-free object detectors. We then introduce some common ways to represent boxes and the improvement of our method. In Section III, we present the proposed AFDet and introduce each module in detail. In Section IV, we first describe the dataset and evaluation metrics and then conduct ablation experiments to verify the effectiveness of each module. Finally, we compare our method to other state-of-the-art methods. Finally, Section V concludes this article.

II. Related Work

A. Anchor-Free Object Detectors

DenseBox [25] was the first anchor-free detector, followed by UnitBox [26], which was an upgrade to DenseBox to achieve better performance. You Only Look Once (YOLO) [27] can be regarded as the first successful anchor-free detector. Nevertheless, because anchor-based methods can achieve high recall rates, anchors were considered indispensable to object detectors for many years. Recently, CornerNet [11] redefined anchor-free methods by detecting an object bounding box as a pair of keypoints (i.e., the top-left and bottom-right corners). Unlike CornerNet, CenterNet [10] represents objects by a single point to reduce false detections. In addition to these approaches, many anchor-free detectors relying on an FPN [28] have been introduced, such as FCOS [9] and Foveabox [29]. Subsequently, some researchers [30], [31] have sought to improve the representation with full instances by extracting extra point features rather than central point features. Recently, detection transformer (DETR) [32] and deformable DETR [33] were developed with a transformer architecture to realize end-to-end detection. However, transformer-based detectors have a relatively large number of parameters, and they are slower than general convolution-based detectors. Unlike the above detectors, AFDet has no complex structure, such as an FPN or a transformer. Moreover, the SSB in AFDet can help the backbone focus more on the contextual or corner information of the object, which is active only during the training. These merits enable AFDet to achieve a good speed-accuracy tradeoff.

B. Representation of Bounding Boxes

In conventional detectors [9], [10], [27], estimating the bounding box can be regarded as a regression task. There are two regression methods: one is to obtain the height and width of the object [10], [27], while the other is to measure the distances from a point to the four sides of the bounding box [9]. The learning objective is a rigid value that usually obeys a Dirac delta distribution in these two ways. Different from the above techniques, some researchers adopt a Gaussian distribution [34], [35] to assess the regression uncertainty, which can reflect the reliability of the bounding box. In addition, the distribution focal loss (DFL) [36] was recently proposed to introduce a more general distribution rather than the Dirac delta or Gaussian distribution for the bounding box. For AFDet, we propose a novel method to estimate the bounding box that treats the bounding box estimation as a classification task rather than a regression task. Compared with regression-based methods, classification through a softmax layer can yield the probability value of each distance. Hence, we can obtain the boundary information by a weighted summation of different distances. In addition, due to the large-scale variation of objects in remote sensing images, the distribution obtained is more dispersed. Thus, by controlling the mean and standard deviation of the distribution, we can make the distribution sharper, leading to more accurate detection.

III. Proposed Method

A. Overall Architecture

In this section, we introduce the overall architecture of AFDet. Fig. 1 presents an overview of the network architecture, which illustrates that AFDet is predominantly composed of a backbone and a head. The modified deep layer aggregation (DLA) [37] network is adopted as the backbone. In addition, we add more skip connections to the original DLA network, which has the same connection style as CenterNet's [10] backbone network.

As mentioned above, the head contains CPB, BEB, and SSB. Given an input image $I \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times 3}$, the output feature $F_b \in \mathbb{R}^{H\times W\times C}$ of the backbone is obtained, where H, W, and C represent the width, height, and channel dimensions of the feature map, respectively. Then, F_b is used as the input to the three branches. CPB and BEB are used to predict the locations of objects and the bounding box estimation parameters, while SSB is used only in training, not in testing. Similar to most detectors, we use soft-nms as our postprocessing operation to produce the final results. More details regarding the core modules of the AFDet are described in the following sections.

B. Center Prediction Branch

CPB, which is shown in Fig. 2, is utilized to categorize all objects in the image and determine their center locations. To do so, CPB takes the feature map F_b from the backbone and predicts the heatmap of the center point. The feature map $F_c \in \mathbb{R}^{H\times W\times S}$ in Fig. 2 is a heatmap that includes the centers of objects in different categories, where H, W, and S represent the width, height, and number of object categories, respectively.

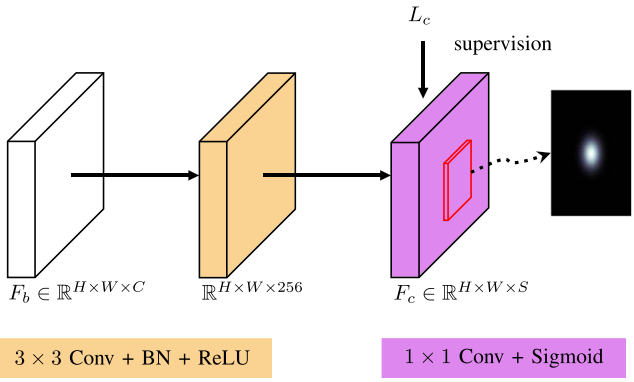

Fig. 2. Architecture of CPB. F_b is the feature map from the backbone. The loss function L_c is used to supervise the output heatmap F_c. The area with an elliptical Gaussian distribution that has different standard deviations for width and height is used to encode a training sample (red rectangle).

The feature map F_c can be defined as follows:

$$F_c = \mathrm{Sig}(\mathrm{C}_{1\times1}(\mathrm{ReLU}(\mathrm{BN}(\mathrm{C}_{3\times3}(F_b))))) \tag{1}$$

where $\mathrm{C}_{1\times1}(\cdot)$ and $\mathrm{C}_{3\times3}(\cdot)$ are 1×1 and 3×3 convolution operations, respectively, $\mathrm{BN}(\cdot)$ is the batch normalization operation, $\mathrm{ReLU}(\cdot)$ is the rectified linear unit activation function, and $\mathrm{Sig}(\cdot)$ is the sigmoid activation function.

To train CPB, the Gaussian heatmap is used as the ground truth. Due to the large aspect ratio of objects in remote sensing images, it is difficult to adapt the CGK [10] to the shapes of the objects. Hence, we adopt EGK in which the standard deviation is determined by the width w and height h of the target. Given the center location (x_0, y_0) of the category $s \in \{1, \ldots, S\}$, the EGK $\mathrm{G_e}(x, y, s) = \exp(-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2})$ is used to produce the target heatmap $F_c' \in \mathbb{R}^{H\times W\times S}$, where $\sigma_x = \frac{\gamma w}{6}$ and $\sigma_y = \frac{\gamma h}{6}$. The selection of the control factor γ is discussed in Section IV-C1. The peak of the Gaussian distribution is selected as the positive sample, while other points are treated as negative samples. Because the points near the center also possess a certain predictive ability, we impose a smaller penalty than for points farther away from the center. We use the modified focal loss L_c as in previous studies [10], [11], i.e.,

$$L_c = -\sum_{s=1}^{S}\sum_{i=1}^{H}\sum_{j=1}^{W} l_{c,\alpha,\beta}\left(F_c(i,j,s), F_c'(i,j,s)\right) \tag{2}$$

where

$$l_{c,\alpha,\beta}(a, a') = \begin{cases} (1-a)^{\alpha}\log(a) & \text{for } a' = 1 \\ (1-a')^{\beta}\, a^{\alpha}\log(1-a) & \text{for } a' \neq 1 \end{cases},$$

α and β are arbitrary constants. We set $\alpha = 2$ and $\beta = 4$.

C. Boundary Estimation Branch

1) Pipeline and Training: In conventional single-stage detectors [6], [8], [10], the bounding box is commonly learned under a Dirac delta distribution, which is a rigid value and may lead to inaccurately estimated boundaries. In BEB, we transform the regression process into a classification task, enabling a finer resolution when generating the boundary. Any regression label

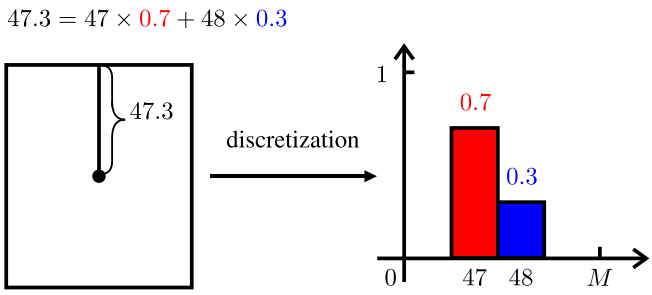

Fig. 3. Given that the continuous distance from the center point to the top side is 47.3, this distance can also be expressed as a weighted summation of two discrete values, i.e., 47 and 48.

$y \in \mathbb{R}^+$ in the task of object detection can be represented as a combination of two discrete labels, y_1 and y_2, as follows:

$$y = W(y_1)y_1 + W(y_2)y_2 \tag{3}$$

where $y_1 = \lfloor y \rfloor, y_2 = \lceil y \rceil$, $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are the floor and ceiling functions, respectively, and $W(y_1)$ and $W(y_2)$ are the corresponding weights of y_1 and y_2, respectively. We set the interval δ between y_1 and y_2 to 1 such that $W(y_1)$ and $W(y_2)$ are computed as

$$W(y_1) = \lceil y \rceil - y \tag{4}$$

$$W(y_2) = y - \lfloor y \rfloor. \tag{5}$$

For example, as shown in Fig. 3, assuming that the distance from the center point to the top side is 47.3, the label can be expressed as a weighted summation of 47 and 48, the weights of which are 0.7 and 0.3, respectively.

Obviously, a softmax layer can be used to obtain the distribution of the distances of the bounding boxes. Suppose that the length of an edge in the bounding box is in the range from 0 to M; then, the range can be divided into M + 1 classes with an integer of 1 as the interval. Hence, the estimated distance from the center to box boundary can be expressed as follows:

$$\hat{y} = \sum_{i=0}^{M} P(y_i)y_i \tag{6}$$

where $P(y_i)$ are the probabilities of locating the target at a distance y_i.

According to the traditional classification task, we use the cross-entropy loss as the distance classification loss L_{cls}. L_{cls} is a weighted summation of two components, i.e.,

$$L_{cls} = -(W(y_j)\log(P(y_j)) + W(y_{j+1})\log(P(y_{j+1}))) \tag{7}$$

where $P(y_j)$ and $P(y_{j+1})$ are the probabilities of locating an object at y_j and y_{j+1}, $y_j = \lfloor y \rfloor$, and $y_{j+1} = \lceil y \rceil$. Intuitively, L_{cls} aims to increase the probabilities of the values around the label y.

Although L_{cls} can force the network to learn the distribution, it cannot accurately reflect the difference between the predicted value and the label because all the negative classes are treated equally. Therefore, we introduce a regression loss L_{focus} to force the results to focus on the correct value, i.e.,

$$L_{\text{focus}} = |\hat{y} - y|. \tag{8}$$

L_{focus} penalizes the error between the weighted predicted value $\hat{y}$ and ground truth y and thus addresses the drawback that the classification loss L_{cls} cannot measure the accurate distance between prediction values and ground truth.

Moreover, due to the complexity of scenes in remote sensing images, the boundaries of objects are difficult to recognize; as a result, the standard deviation of the distribution is usually ambiguous. Hence, it is reasonable to use a loss to control the scattered distribution. We define a gather loss L_{gather} as follows:

$$L_{\text{gather}} = \sum_{i=0}^{M} P(y_i)(y_i - \hat{y})^2. \tag{9}$$

L_{gather} requires the estimated distribution to be concentrated within a small range around the predicted value. Accordingly, the distribution becomes sharper to eliminate potential confusion regarding the distribution.

The overall architecture of BEB is illustrated in Fig. 4. The feature map $F_e \in \mathbb{R}^{H \times W \times 4(M+1)}$ is transformed from F_b through standard 3×3 convolution and 1×1 convolution, followed by a softmax layer. To supervise BEB, we use L_e, which is defined as

$$L_e = L_{\text{cls}} + \lambda_{\text{focus}} L_{\text{focus}} + \lambda_{\text{gather}} L_{\text{gather}} \tag{10}$$

where λ_{focus} and λ_{gather} are two hyperparameters that we recommend be set to small values because they fluctuate greatly in the early training stage. We choose $\lambda_{\text{focus}} = 0.05$ and $\lambda_{\text{gather}} = 0.01$ for our network, but they are not carefully selected. With the supervision of L_e, we can obtain the distributions from the center point to the four sides (top, left, bottom, and right). The M in F_e denotes the range of distances. The final distance result can be obtained by the weighted summation of the different distances multiplied by the corresponding probabilities. The selection of M depends on the size of the object in the input image, which is roughly between 0 and 400. Because the stride of the backbone is 4, the value of M is set to 100.

2) Analysis: In this section, we mainly analyze the specific impact of L_{focus} and L_{gather}. The specific analysis is as follows.

The output, $P(y_i)$, which is obtained by a softmax activation function, can be computed as

$$P(y_i) = \frac{e^{z_i}}{\sum_{k=0}^{M} e^{z_i}} \tag{11}$$

where z is an $M + 1$ dimensional vector before the softmax function and z_i is one element of z.

According to (6) and (8), the gradient of L_{focus} w.r.t. $P(y_i)$ can be computed as

$$\frac{\partial L_{\text{focus}}}{\partial P(y_i)} = \begin{cases} y_i & \text{if } y < \hat{y} \\ -y_i & \text{if } y > \hat{y}. \end{cases} \tag{12}$$

Based on (11), the gradient of $P(y_i)$ w.r.t z_j can be computed as

$$\begin{aligned} \frac{\partial P(y_i)}{\partial z_j} &= \frac{e^{z_j}}{\sum_{k=0}^{M} e^{z_k}} - \left(\frac{e^{z_j}}{\sum_{k=0}^{M} e^{z_k}}\right)^2 \\ &= P(y_j) - P(y_j)^2, \text{ if } j = i \end{aligned} \tag{13}$$

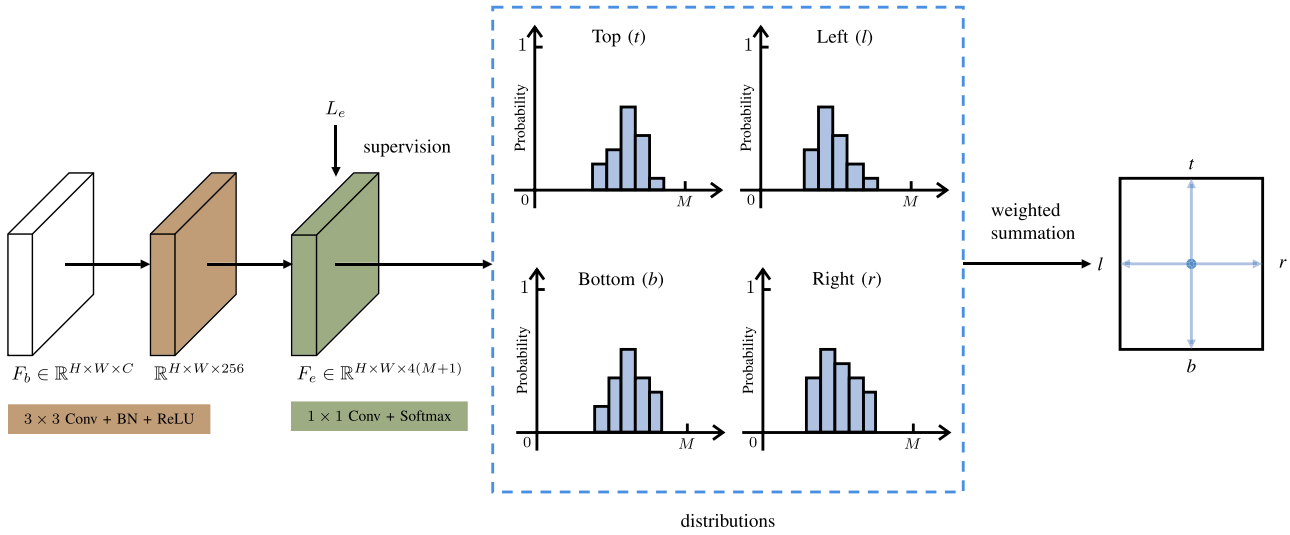

Fig. 4. The architecture of BEB. F_b is the feature map from the backbone. L_e is used to supervise the output feature map F_e, which is the summation of L_{cls}, L_{focus}, and L_{gather}. The distance distribution of an object is extracted from F_e according to the position of the center point. The final distance result is obtained by the weighted summation of the different distances multiplied by the corresponding probabilities in the distributions.

and

$$\frac{\partial P(y_i)}{\partial z_j} = -\frac{e^{z_i}}{\left(\sum_{k=0}^{M} e^{z_k}\right)^2} * e^{z_j} = -P(y_j) * P(y_i), \text{ if } j \neq i. \tag{14}$$

Based on (6), (12), (13), and (14), if $y < \hat{y}$, the gradient of L_{focus} w.r.t. z_i can be written as

$$\begin{aligned}\frac{\partial L_{\text{focus}}}{\partial z_i} &= y_i * (P(y_i) - P(y_i)^2) - \sum_{k=0,k\neq j}^{M} y_k * P(y_k) * P(y_i) \\ &= P(y_i)(y_i - y_i * P(y_i) - \sum_{k=0,k\neq j}^{M} y_k * P(y_k)) \\ &= P(y_i)(y_i - \hat{y}).\end{aligned} \tag{15}$$

Intuitively, from (15), the network will be updated to increase the probabilities of the classes y_i, which are larger than those of $\hat{y}$ due to their positive gradients, and decrease the probabilities of the classes y_i, which are smaller than those of $\hat{y}$ due to their negative gradients. If $y > \hat{y}$, the gradient of L_{focus} w.r.t z_i equals $P(y_i)(\hat{y} - y_i)$. Thus, the network will be updated to increase the probabilities of the classes y_i, which are smaller than those of $\hat{y}$ due to their positive gradients, and decrease the probabilities of the classes y_i, which are smaller than those of $\hat{y}$ due to their negative gradients.

According to (6) and (9), the gradient of L_{gather} w.r.t. $P(y_i)$ can be computed as

$$\begin{aligned}\frac{\partial L_{\text{gather}}}{\partial P(y_i)} &= (y_i - \hat{y})^2 - 2 * y_i * \sum_{k=0}^{M} P(y_i)(y_k - \hat{y}) \\ &= (y_i - \hat{y})^2.\end{aligned} \tag{16}$$

Based on (14), (15), and (16), the gradient of L_{gather} w.r.t z_i can be expressed as

$$\begin{aligned}\frac{\partial L_{\text{gather}}}{\partial z_i} &= (y_i - \hat{y})^2 (P(y_i) - P(y_i)^2) \\ &\quad - \sum_{k=0,k\neq j}^{M} (y_k - \hat{y})^2 * P(y_k) * P(y_i)) \\ &= P(y_i)((y_i - \hat{y})^2 - \sum_{k=0}^{M} (y_k - \hat{y})^2 * P(y_k))\end{aligned} \tag{17}$$

For notational convenience, we use Λ to represent the term $\sum_{k=0}^{M}(y_k - \hat{y})^2 * P(y_k)$ in (17). Thus, (17) can be simplified as

$$\frac{\partial L_{\text{gather}}}{\partial z_i} = P(y_i)((y_i - \hat{y})^2 - \Lambda). \tag{18}$$

The gradient in (18) has the following properties:

$$\begin{aligned} &y_i \in \left(\hat{y} - \sqrt{\Lambda}, \hat{y} + \sqrt{\Lambda}\right), \\ &\frac{\partial L_{\text{gather}}}{\partial z_i} < 0. \end{aligned} \tag{19}$$

and

$$\begin{aligned} &y_i \in \left[0, \hat{y} - \sqrt{\Lambda}\right) \cup \left(\hat{y} + \sqrt{\Lambda}, M\right], \\ &\frac{\partial L_{\text{gather}}}{\partial z_i} > 0. \end{aligned} \tag{20}$$

Eq. (19) shows that the network will be updated to increase the probabilities of the classes y_i close to $\hat{y}$ ($y_i \in (\hat{y} - \sqrt{\Lambda}, \hat{y} + \sqrt{\Lambda})$) via their negative gradients. By contrast, (20) shows that the network will be updated to decrease the probabilities of the classes y_i far from $\hat{y}$ ($y_i \in [0, \hat{y} - \sqrt{\Lambda}) \cup (\hat{y} + \sqrt{\Lambda}, M]$) via their positive gradients.

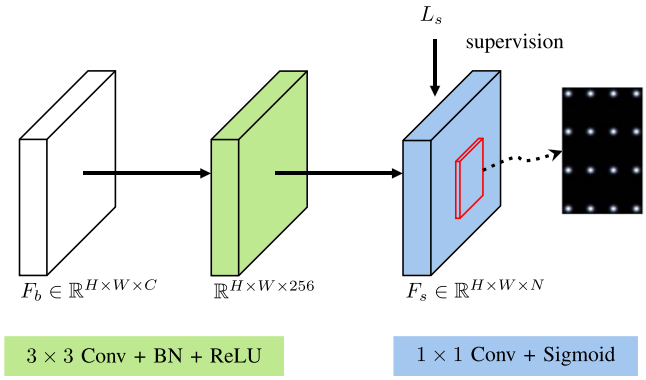

Fig. 5. Architecture of SSB. F_b is the feature map from the backbone. L_s is used to supervise the output feature map F_s. The visualized semantic heatmap for one object (red rectangle) shown on the right is the element-wise addition of all the keypoint heatmaps in the channel dimension. The CGK is used to encode all the keypoints.

D. Semantic Supervision Branch

The feature of a center point cannot accurately represent the entire object; thus, some methods [30], [31] have been proposed to extract an informative corner or border features to enhance the object detection accuracy. Different from these methods, SSB uses keypoints both on the boundary and within the interior of an object to extract the semantic features. Consequently, supervising this branch can force the backbone to learn more semantic information (i.e., in addition to the center point). The semantic heatmap $F_s \in \mathbb{R}^{H\times W\times N}$ shown in Fig. 5 is the output of SSB, where N is the number of keypoints.

To train SSB, we use a CGK $\mathrm{G_c}(x,y) = \exp(-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2})$ to encode the target heatmap, where (x_0, y_0) denotes the location of a keypoint and σ is the standard deviation adapted to the object size [10], [11]. The loss L_s required to supervise SSB is also the modified focal loss, which is the same as L_c. The keypoints are uniformly distributed along the spatial dimensions (rows and columns) of the object. Different from those of the CPB output, all the object categories of the SSB output share the same keypoint heatmap. Specifically, a channel in F_s represents a keypoint in a different location of an instance. For example, the heatmap in Fig. 5 signifies the sum of the keypoint heatmaps in the channel dimension, which includes four keypoints in the horizontal or vertical direction. Note that F_s does not include the center point of an object. Therefore, when there are odd numbers of points in the rows and columns, the center point is deleted. In addition, we verify the influence of the number of keypoints in Section IV-C3.

E. Total Loss

The total loss L_T is the summation of the above three branch losses with different weighting coefficients. Specifically,

$$L_T = \lambda_c L_c + \lambda_s L_s + \lambda_e L_e \tag{21}$$

where $\lambda_c = 1.0$, $\lambda_s = 0.1$, and $\lambda_e = 0.3$, which are empirically set. In the previous sections, we summarize the network structure and training process. To better understand the overall inference process of AFDet, we list the steps in Algorithm 1.

Algorithm 1: Inference of AFDet.

Input: An input image $I \in \mathbb{R}^{\mathcal{H}\times\mathcal{W}\times 3}$.
Output: All the bounding boxes of the detected objects.

Obtain the output feature $F_b \in \mathbb{R}^{H\times W\times 256}$ of the backbone
Transform F_b into output feature $F_c \in \mathbb{R}^{H\times W\times S}$ of CPB and $F_e \in \mathbb{R}^{H\times W\times 4(M+1)}$ of BEB
Find all satisfied N points whose scores are greater than 0.1 in F_c as the detected object centers: C $\leftarrow \{(x_i, y_i) \mid i = 1, 2, \ldots N\}$
for $i \leftarrow 1$ *to* N **do**
    Extract the distance distributions d_t, d_b, d_l and d_r of an object in F_e
    Calculate the distance values t, b, l and r from d_t, d_b, d_l and d_r using Eq. 6
    Calculate bounding box parameters $\{x_i^a, y_i^a, x_i^b, y_i^b\}$ on the input image:
    $x_i^a = 4(x_i - l), \quad y_i^a = 4(y_i - t)$
    $x_i^b = 4(x_i + r), \quad y_i^b = 4(y_i + b)$
end
Apply soft-nms to decoded bounding boxes
Return all the detected bounding boxes

IV. Experimental Results and Analysis

A. Dataset and Implementation Details

We conduct experiments on three extensively used datasets, i.e., the NWPU VHR-10 [22], DIOR [23] and HRRSD [24] datasets. The NWPU VHR-10 dataset contains 650 annotated images distributed into ten categories. There are no partitioned training, validation, and test sets in the original NWPU VHR-10 dataset; thus, in accordance with previous studies, we randomly use 60 % for training, 20 % for validation and the remaining 20 % for testing. For the DIOR dataset, 11 725 remote sensing images are used as the training set, and the remaining 11 738 images are employed as the test set. Likewise, the HRRSD data are divided into three parts (training, validation, and test sets), with 5401, 5417, and 10 913 images, respectively.

For all experiments implemented based on our proposed detector, DLA-34 series networks (DLA-34 and DLA-34-DCN) are used as the backbone networks. DLA-34 series networks are more lightweight than the commonly used ResNet-101 networks. Compared with DLA-34, DLA-34-DCN modifies every convolution layer in upsampling stages to a deformable convolutional layer [38]. We initialize the backbone networks with the weights pretrained on ImageNet. Specifically, rectified Adam (RAdam) [39] is selected as the optimizer for our model. For the HRRSD and DIOR datasets, the model is trained within a total of 30 epochs, and the batch size is 6. The learning rate is set to 0.0005 at the beginning, which is multiplied by 0.1 at epochs 20 and 25. For the NWPU VHR-10 dataset, the model is trained within a total of 140 epochs. The learning rate is set

to 0.0001 at the beginning, which is multiplied by 0.1 at epochs 90 and 120. For all the datasets, the input image is padded with zeros to a size of 800×800 if its size is smaller than 800×800. We use simple data augmentation operations including random horizontal flipping, vertical flipping, and random scaling to enhance the robustness of the detector. We perform the experiments under the PyTorch framework on a PC with an Intel single-core i7 CPU and a GeForce RTX 1080 Ti GPU.

B. Evaluation Metrics

To evaluate the detection performance of our proposed network, we use two types of methods following the evaluation protocol in Common Objects in Context (COCO) [40] and PASCAL Visual Object Classes (VOC) [41]. Among the PASCAL VOC metrics, the mean average precision (mAP) represents the mean value of the average precision of each class ($\mathrm{AP_v}$) at the intersection over union $(\mathrm{IoU}) = 0.50$. $\mathrm{AP_v}$ is the value of the area enclosed by the coordinate axes and a precision-recall curve (PRC). The PRC is plotted by the precision P and recall R, which are defined as

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{22}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{23}$$

where TP, FP, and FN denote the numbers of true-positives, false-positives, and false-negatives, respectively. $\mathrm{AP_v}$ can be defined as

$$\mathrm{AP_v} = \int_0^1 \mathrm{P_m(r)} dr \tag{24}$$

where $\mathrm{P_m(r)}$ is the measured precision when the recall R equals r.

Among the COCO metrics, $\mathrm{AP}, \mathrm{AP_{50}}, \mathrm{AP_{75}}$ are used in this article. Different from the $\mathrm{AP_v}$ metric in PASCAL VOC, the $\mathrm{AP_{50}}$ metric in COCO is defined as the mean precision at a set of 101 equally spaced recall levels [0, 0.01, ..., 1] at $\mathrm{IoU} = 0.5$

$$\mathrm{AP_{50}} = \frac{1}{101} \sum_{r \in \{0, 0.01, \ldots, 1\}} \mathrm{P_m(r)}. \tag{25}$$

$\mathrm{AP_{75}}$ is at $\mathrm{IoU} = 0.75$, and AP corresponds to the mean average precision for IoU from 0.5 to 0.95 with a step size of 0.05.

C. Ablation Experiments

We conduct comprehensive experiments to evaluate the contribution of each component in the proposed algorithm. We use the DLA-34 network as the backbone. All the results in this section are presented in the COCO evaluation metric style.

1) Evaluation of CPB: As mentioned in Section III-B, CPB is used to predict the center of an object. To improve the detection accuracy, the positive sample in the heatmap is encoded by an EGK, which is more suitable for objects with a large aspect ratio than a CGK. The EGK $\mathrm{G_e}(x, y) = \exp(-\frac{(x-x_0)^2}{2\sigma_x^2} - \frac{(y-y_0)^2}{2\sigma_y^2})$ has two variances, σ_x and σ_y, which are proportional to the height and width of the object, respectively. The larger the variance, the faster the rate at which the confidence level of the deviation from the center of the object drops. Therefore, we verify the AP under different values of the regulatory factor γ (from 0.1 to 0.9) in σ_x and σ_y. EGK with different γ values are visualized in Fig. 6. The highly responsive region clearly increases with increasing γ. In Table I, we report the results on HRRSD, in which CGK and EGK denote the circular and EGK, respectively. Except for the situations where γ equals 0.1 and 0.9, the AP of EGK is higher than that of CGK. This discrepancy arises because small values of γ lead to few training samples, and a large γ value introduces confusing location information. Therefore, the value of AP first increases and then decreases with increasing γ. When γ equals 0.5, the values of AP and $\mathrm{AP_{75}}$ reach their maximums; however, $\mathrm{AP_{50}}$ does not reach its maximum because $\mathrm{AP_{50}}$ is a relatively loose metric compared with AP and $\mathrm{AP_{75}}$.

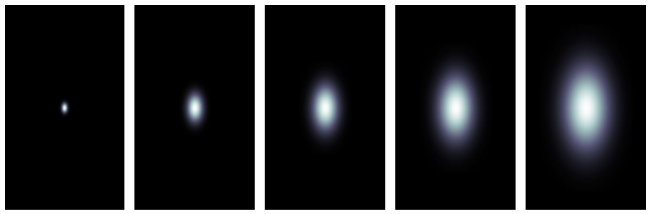

Fig. 6. Visualization of EGK with different values of γ varying from 0.1 to 0.9. From left to right, the highly responsive region increases with increasing γ.

TABLE I
PERFORMANCE OF AFDET WITH A CGK (CGK) AND EGK OF VARYING γ FROM 0.1 TO 0.9 USING THE HRRSD DATASET

Method	AP	$\mathrm{AP_{50}}$	$\mathrm{AP_{75}}$
CGK	56.99	86.42	64.79
EGK (γ=0.1)	56.56	86.31	64.34
EGK (γ=0.3)	58.35	88.20	66.60
EGK (γ=0.5)	**58.40**	88.82	**66.69**
EGK (γ=0.7)	57.54	88.84	65.46
EGK (γ=0.9)	56.81	**89.01**	64.07

TABLE II
PERFORMANCE OF AFDET WITH DIFFERENT BOX ESTIMATION METHODS USING THE HRRSD DATASET

Method	AP	$\mathrm{AP_{50}}$	$\mathrm{AP_{75}}$
$wh + offset$	55.57	88.24	61.44
$tblr$	54.22	88.36	59.62
BEB (L_{cls})	56.90	88.59	64.87
BEB ($L_{cls} + L_{focus}$)	57.77	**88.95**	65.60
BEB ($L_{cls} + L_{focus} + L_{gather}$)	**58.40**	88.82	**66.69**

2) Evaluation of BEB: As mentioned in Section III-C1, BEB is used to predict the distances from the center point to the four sides through the corresponding distributions. Moreover, we can obtain a more accurate boundary by supervising the distributions using the focus loss and gather loss. First, to verify the effectiveness of the classification method, we compare it with conventional regression methods using HRRSD. As shown in Table II, the $wh + offset$ method denotes the width and

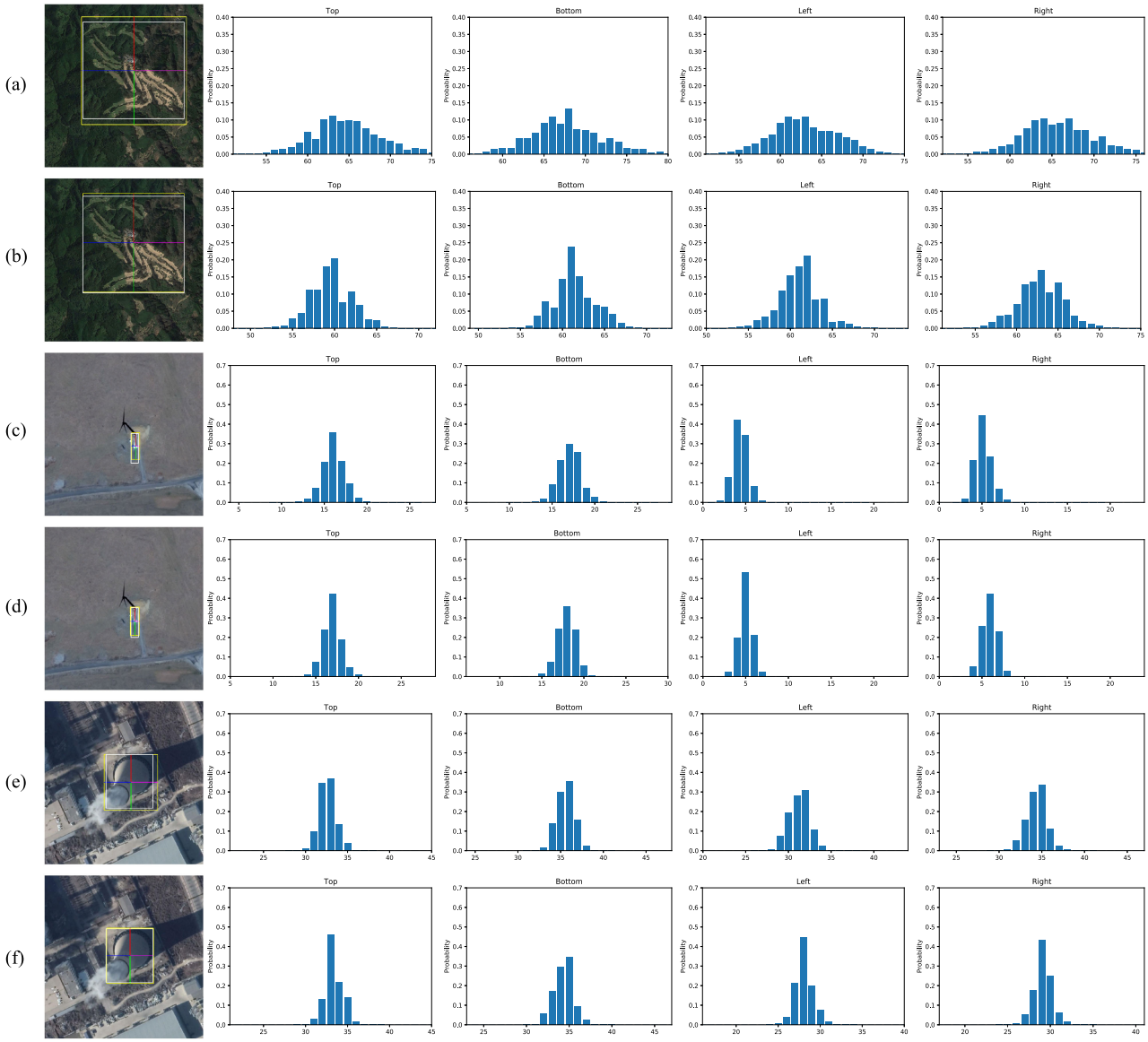

Fig. 7. Comparison of detection results and visualized output distribution in BEB (L_{cls}) and BEB ($L_{\text{cls}} + L_{focus} + L_{\text{gather}}$). Note that the yellow box represents the predicted box and the white box is the ground-truth box. The distances from the predicted center point to the top (red line), bottom (green line), left (blue line), and right (pink line) are shown in the predicted figures. (a), (c), (e) Detection results in BEB (L_{cls}). (b), (d), (f) Detection results in BEB ($L_{\text{cls}} + L_{\text{focus}} + L_{\text{gather}}$).

height of a bounding box and the offset from the center point, while the $tblr$ method denotes the distances from the center to the top, bottom, and left and right sides. Our method clearly outperforms these two commonly used regression methods on all evaluation metrics. Second, we investigate the effect of L_{focus} and L_{gather}. Table II reveals that BEB ($L_{\text{cls}} + L_{\text{focus}}$) performs better than BEB (L_{cls}) on all metrics. With the help of L_{gather}, BEB ($L_{\text{cls}} + L_{\text{focus}} + L_{\text{gather}}$) outperforms BEB ($L_{\text{cls}} + L_{\text{focus}}$), except for AP_{50}. To further intuitively highlight the effectiveness of L_{focus} and L_{gather}, we visualize the distance distributions from the center point to the four sides. As shown in Fig. 7, the figures in the first column include a golf field, windmill, and chimney in a complex scenario. (a), (c), (e) present the detection results and distributions in BEB (L_{cls}), while (b), (d), (f) present the results in BEB ($L_{\text{cls}} + L_{\text{focus}} + L_{\text{gather}}$). As shown in Fig. 7(a) and (b), the golf field has an irregular shape, and the boundaries are difficult to regress. Although BEB (L_{cls}) performs a relatively accurate estimation, the results can be further optimized with the help of L_{focus} and L_{gather}. The distance distributions in the first row have a relatively large standard deviation due to the ambiguous boundaries. However, the distance distributions of BEB ($L_{\text{cls}} + L_{\text{focus}} + L_{\text{gather}}$) in the second row have a sharper appearance, enabling more accurate boundary estimation. In Fig. 7(c) and (d), the windmill has some subtle structures similar to the background area. The detection results in BEB ($L_{\text{cls}} + L_{\text{focus}} + L_{\text{gather}}$) are more accurate than those in BEB(L_{cls}). For the chimney with thick smoke and shadow in Fig. 7(e) and (f), BEB ($L_{\text{cls}} + L_{\text{focus}} + L_{\text{gather}}$) can obtain the same boundaries as the ground truth due to the more focused and accurate distributions.

3) Evaluation of SSB: As mentioned in Section III-D, SSB is used to force the backbone network to focus on the semantic

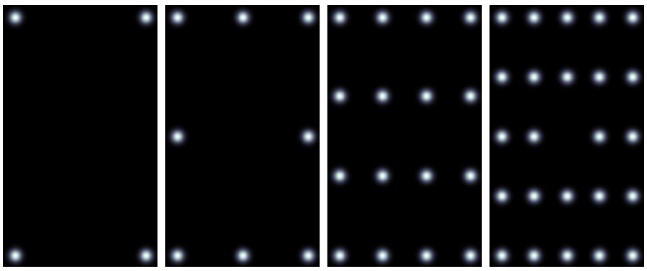

Fig. 8. Visualization of semantic heatmaps for one object in SSB with different values of the number of keypoints N. The visualized heatmap is the elementwise addition of N keypoint heatmaps in the channel dimension.

TABLE III
PERFORMANCE OF AFDET WITH DIFFERENT VALUES OF THE NUMBER OF KEYPOINTS N USING THE HRRSD DATASET

Method	AP	AP_{50}	AP_{75}
w/o SSB	56.90	88.58	64.87
w/ SSB ($N = 4$)	57.90	88.92	66.27
w/ SSB ($N = 8$)	58.11	**88.99**	66.53
w/ SSB ($N = 16$)	**58.40**	88.82	**66.69**
w/ SSB ($N = 24$)	57.94	88.84	66.21

features of the entire instance. In addition to the center point, the keypoints of other locations are the learning objectives of the network. Hence, the use of different numbers of keypoints N affects the final accuracy. Heatmaps with different values of N are shown in Fig. 8. When N equals 4, the network can learn the corner information of the entire object, and for N equals 8, the network can learn the corner and middle point features. For the value of N larger than 8, information on both boundary and internal keypoints can be obtained. In Table III, we report the performance of AFDet with different values for the number of keypoints N using the HRRSD dataset, in which w/ SSB and w/o SSB indicate that the SSB is and is not used, respectively. The networks under the supervision of SSB all clearly perform better than those without the supervision of SSB, especially regarding the AP and AP_{75}. The AP achieves its maximum when N equals 16, suggesting that the feature information inside objects is as crucial as the information along the boundary. However, when N equals 24, the accuracy declines, which may be due to the introduction of confusing information among different keypoints. To better understand the benefits of SSB for object detection, we visualize the output feature maps of CPB under different conditions using the DIOR dataset. Fig. 9 presents the w/ SSB and w/o SSB heatmaps. Fig. 9(a) shows the input images, which include a car, three harbors, and densely distributed ships, while Fig. 9(b) and (c) present the harbor category heatmap and ship category heatmap, respectively, and Fig. 9(d) shows the final detection results. Fig. 9(b) indicates that the w/ SSB method can accurately locate the center of the harbor because the supervision of extra points plays a positive role in locating the center of the harbor. By contrast, the harbor heatmap of the w/o SSB method is ambiguous, as this approach leads to missed detections and errors in the final results. Moreover, the w/ SSB method detects almost all ships and cars compared with the w/o SSB method, which yields some errors regarding the ships and cars in the images.

D. Comparisons With State-of-The-Art Methods

In this section, we compare AFDet with several state-of-the-art object detectors, including two-stage methods such as HRCNN [24], GACL [45], and CSFF [12]; single-stage anchor-based methods such as SSD [7], YOLOv3 [6], MFPNet [43], HawkNet [44], and RetinaNet [8]; and single-stage anchor-free methods such as CenterNet [10], FCOS [9], RepPoints [30], and O^2-DNet [42]. First, we compare the mAPs of the above methods and our methods using the DIOR, HRRSD and NWPU VHR-10 datasets, and we then compare the number of parameters and frames per second (FPS) using the NWPU VHR-10 dataset.

Table IV shows a comparison of the performance of our proposed method with that of other methods using the DIOR dataset. Our method performs well for almost all categories, especially airplanes, basketball courts, bridges, and tennis courts. The scale distribution of objects in this dataset varies greatly, so the performance of the anchor-based methods is limited by the rigid setting of anchors. Moreover, RepPoints performs better than FCOS because it uses multiple points to capture the geometric information of objects. Different from RepPoints and FCOS, our method and O^2-DNet do not adopt the multilayer prediction strategy of an FPN. However, our method using only the DLA-34 backbone can achieve similar performance to O^2-DNet using the Hourglass-104 backbone, which has many more parameters than DLA-34. The addition of DCN to DLA-34 can further improve the network, yielding a gain of 2.8% over AFDet using DLA-34, which also performs better than MFPNet and HawkNet.

The quantitative results of applying different methods to the HRRSD dataset are shown in Table V. Most of the methods achieve good results because this dataset is not as complex as the DIOR dataset and has fewer object categories. The anchor-free methods still perform better than the anchor-based methods, and it is worth mentioning that AFDet (w/ DLA-34) is even better than the strong anchor-free detector RepPoints (w/ ResNet-101). AFDet (w/ DLA-34-DCN) achieves the best mAP and AP for some categories, such as vehicles and ships. AFDet appears to be good at detecting objects with fine textural features, such as basketball and tennis courts.

Table VI shows the mAPs of different methods using the NWPU VHR-10 dataset, demonstrating that the performance of anchor-free methods is similar performance to that of other methods because this dataset is relatively small compared with the HRRSD and DIOR datasets. However, our method achieves a very high mAP, which verifies that our method can maintain good performance with both large and small datasets. Fig. 10 shows several detection results of our proposed method. The images in the first, second and third rows are from the Dior, HRRSD, and NWPU datasets, respectively.

In addition to precision, the speed and number of parameters of detectors are important indicators. Hence, we report the FPS and numbers of parameters of commonly used methods and

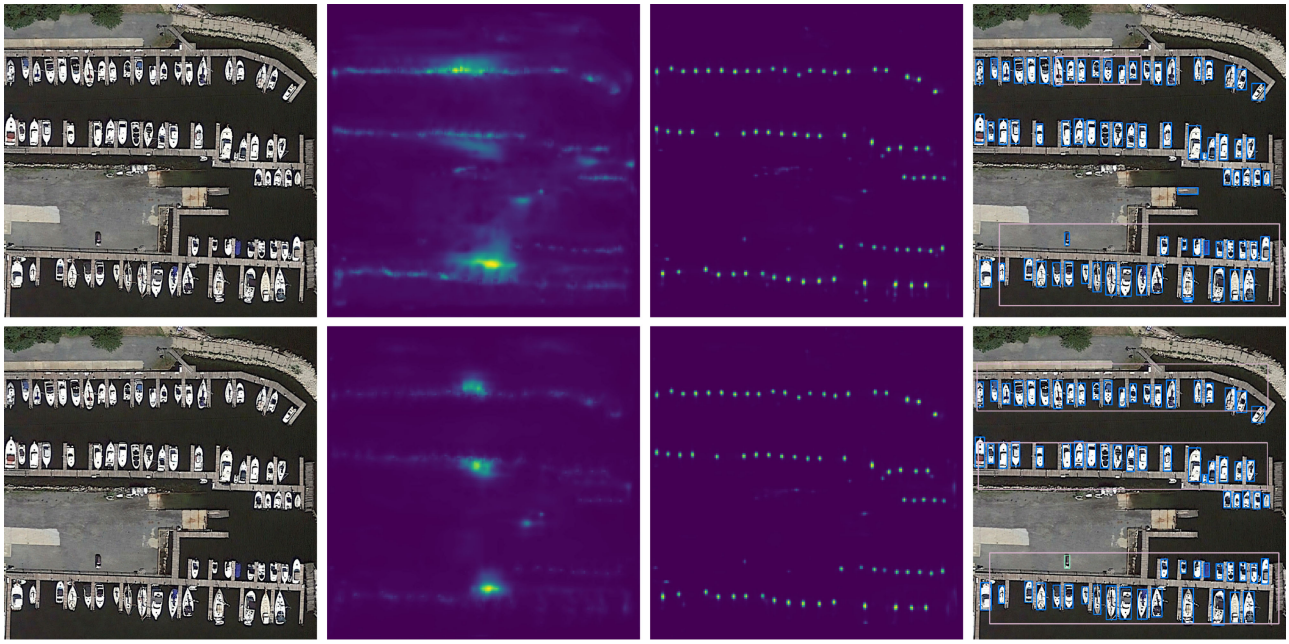

Fig. 9. Comparison of detection results and visualized output heatmaps in CPB without SSB and with SSB using sample images from the DIOR dataset. Note that the blue boxes represent detected ships, pink boxes represent detected harbors, and green boxes represent cars. The first row presents the results without SSB, and the second row presents the results with SSB. (a) Original image. (b) Heatmap in CPB corresponding to harbor category. (c) Heatmap in CPB corresponding to ship category. (d) Detection results.

TABLE IV
PERFORMANCE OF OBJECT DETECTORS USING THE DIOR DATASET

Model	Backbone	AP	AI	BF	BC	BR	CH	DA	ESA	ETS	GF	GTF	HA	OV	SH	SD	ST	TC	TS	VE	WM	mAP
YOLOv3[6]	Darknet-53	60.9	55.6	65.1	82.8	22.5	68.5	40.2	58.6	55.2	61.8	61.6	50.6	44.9	82.2	54.1	56.7	80.1	34.8	32.9	70.1	57.0
SSD[7]	VGG16	59.5	72.7	72.4	75.7	29.7	65.8	56.6	63.5	53.1	65.3	68.6	49.4	48.1	59.2	61.0	46.6	76.3	55.1	27.4	65.7	58.6
CenterNet[10]	DLA-34	67.8	68.8	75.0	83.2	44.6	67.6	44.7	71.0	67.7	66.6	72.5	41.6	56.3	72.0	54.2	70.3	85.3	48.6	51.6	79.2	64.4
RetinaNet[8]	ResNet-101	53.3	77.0	69.3	85.0	44.1	73.2	62.4	78.6	62.8	78.6	76.6	49.9	59.6	71.1	68.4	45.8	81.3	55.2	44.4	85.5	66.1
CSFF[12]	ResNet-101	57.2	79.6	70.1	87.4	46.1	76.6	62.7	82.6	73.2	78.2	81.6	50.7	59.5	73.3	63.4	58.5	85.9	61.9	42.9	86.9	68.0
FCOS[9]	ResNet-101	63.2	81.9	75.3	85.9	40.7	77.7	63.6	85.1	65.0	**83.8**	74.7	54.4	57.4	70.3	65.1	59.0	83.8	56.2	37.9	83.1	68.2
O^2-DNet[42]	Hourglass-104	61.2	80.1	73.7	81.4	45.2	75.8	64.8	81.2	**76.5**	79.5	79.7	47.2	59.3	72.6	70.5	53.7	82.6	55.9	49.1	77.8	68.4
RepPoints[30]	ResNet-101	64.2	**85.7**	76.0	88.2	42.5	**78.0**	67.8	**86.0**	60.9	82.1	**81.9**	57.7	59.8	72.7	**71.0**	62.6	87.1	**66.3**	43.2	81.4	70.8
MFPNet[43]	VGG16	76.6	83.4	80.6	82.1	44.3	75.6	**68.5**	85.9	63.9	77.3	77.2	**62.1**	58.8	77.2	76.8	60.3	86.4	64.5	41.5	80.2	71.2
HawkNet[44]	ResNet-50	65.7	84.2	76.1	87.4	45.3	79.0	64.5	82.8	72.4	82.5	74.7	50.2	59.6	**89.7**	66.0	70.8	87.2	61.4	52.8	**88.2**	72.0
AFDet	DLA-34	74.7	69.0	81.6	**89.8**	46.1	73.7	42.5	75.1	74.7	72.0	80.6	34.9	58.4	77.4	62.0	76.0	90.0	53.9	56.7	85.1	68.7
AFDet	DLA-34-DCN	**82.4**	81.5	**81.9**	**89.8**	**51.7**	74.9	58.7	84.2	73.3	79.5	81.0	44.2	**62.0**	77.8	63.2	**76.9**	**91.0**	62.5	**59.3**	87.1	**73.2**

The abbreviations for the categories are AP-Airplane, AI-Airport, BF-Baseball Field, BC-Basketball Court, BR-Bridge, CH-Chimney, DA-Dam, ESA-Expressway Service Area, ETS-Expressway Toll Station, GF-Golf Field, GTF-Ground Track Field, HA-Harbor, OV-Overpass, SH-Ship, SD-Stadium, ST-Storage Tank, TC-Tennis Court, TS-Train Station, VE-Vehicle, and WM-Windmill.

TABLE V
PERFORMANCE OF OBJECT DETECTORS USING THE HRRSD DATASET

Method	Backbone	AP	BD	BC	BR	CR	GTF	HA	PL	SH	ST	TJ	TC	VE	mAP
HRCNN [24]	ResNet-101	82.93	72.11	24.94	28.31	32.26	80.57	61.57	21.35	57.64	78.76	10.25	74.83	42.84	51.43
GACL [45]	ResNet-101	90.8	87.2	49.7	85.6	88.2	94.8	89.7	65.3	88.5	89.2	75.0	80.8	86.8	82.1
SSD [7]	VGG16	90.68	77.27	35.64	79.57	86.81	90.32	84.63	42.23	76.99	89.06	66.11	76.65	81.97	75.22
YOLOv3 [6]	Darknet-53	98.08	80.73	56.49	79.99	88.15	95.34	87.62	55.62	87.19	88.17	70.14	90.07	83.44	81.62
RetinaNet [8]	ResNet-101	96.11	88.91	61.41	85.67	87.47	96.75	91.75	43.46	81.90	94.35	70.99	90.02	93.24	83.16
CenterNet [10]	DLA-34	98.42	85.09	68.92	83.61	92.35	94.20	91.12	61.39	89.21	94.68	78.51	92.84	94.54	86.53
FCOS [9]	ResNet-101	96.82	91.21	54.10	89.69	94.42	97.45	95.05	63.15	90.46	94.91	82.23	87.85	91.82	86.86
RepPoints [30]	ResNet-101	97.49	**91.90**	60.83	91.04	**95.10**	**98.22**	**95.70**	**71.78**	90.05	94.36	82.26	89.83	94.25	88.71
AFDet	DLA-34	98.38	88.48	75.65	89.07	94.23	95.10	93.59	65.40	91.98	**95.97**	82.62	**93.92**	96.20	89.28
AFDet	DLA-34-DCN	**99.06**	91.60	**75.92**	**91.66**	95.07	96.06	94.74	71.30	**94.11**	95.01	**83.79**	93.62	**96.64**	**90.68**

The abbreviations for the categories are defined as Follows: AP-Airplane, BD-Baseball Diamond, BC-Basketball Court, BR-Bridge, CR-Crossroad, GTF-Ground Track Field, HA-Harbor, PL-Parking Lot, SH-Ship, ST-Storage Tank, TJ-T Junction, TC-Tennis Court, and VE-Vehicle.

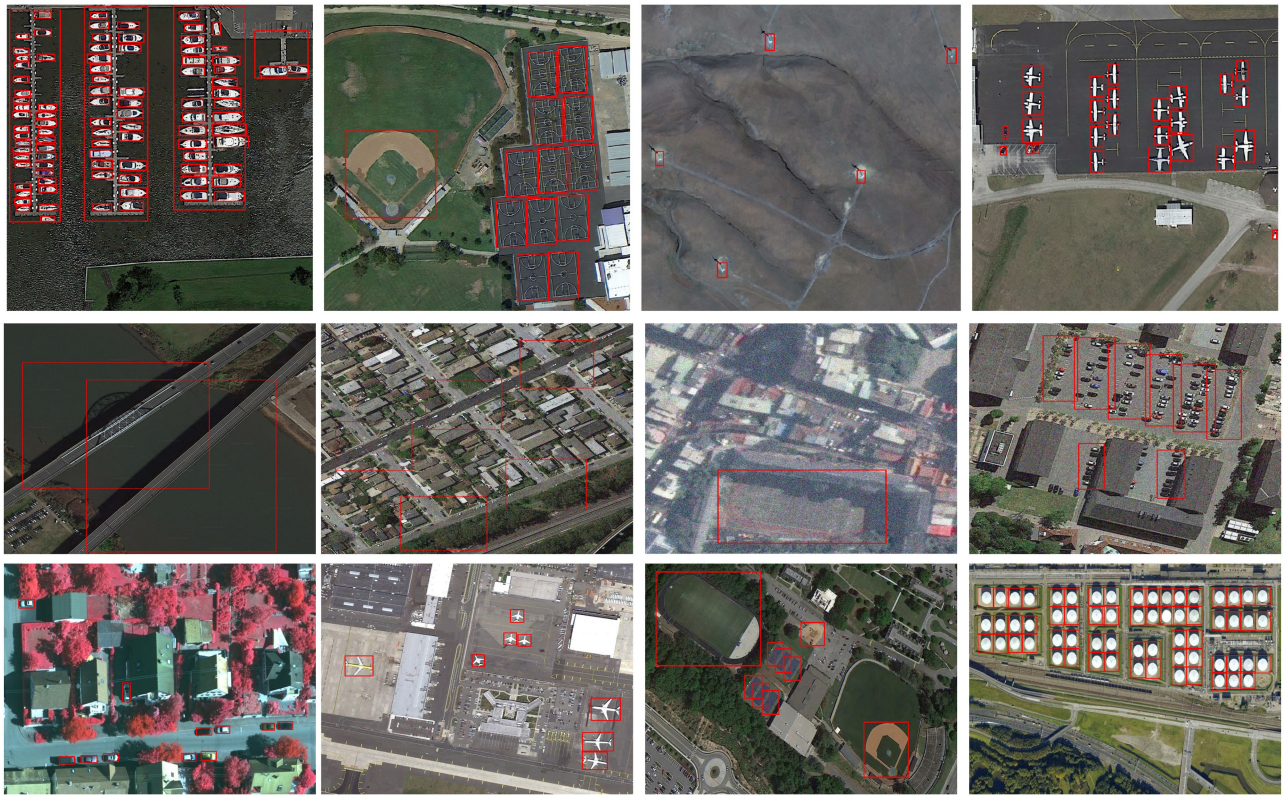

Fig. 10. Detection results of the AFDet for sample images. The images in the first, second and third rows are from the Dior, HRRSD and NWPU datasets, respectively.

TABLE VI
PERFORMANCE OF OBJECT DETECTORS WITH THE NWPU VHR-10 DATASET USING 60 % FOR TRAINING, 20 % FOR VALIDATION AND 20 % FOR TESTING

Method	Backbone	mAP
SSD [7]	VGG16	89.69
YOLOv3 [6]	Darknet-53	94.74
CenterNet [10]	DLA-34	95.59
RetinaNet [8]	ResNet-101	95.95
FCOS [9]	ResNet-101	93.49
RepPoints [30]	ResNet-101	95.24
AFDet	DLA-34	97.01
AFDet	DLA-34-DCN	**97.45**

TABLE VII
COMPUTING TIME AND NUMBER OF PARAMETER COMPARISONS OF DIFFERENT METHODS AS THE AVERAGE NUMBER OF PROCESSED IMAGES PER SECOND USING THE NWPU VHR-10 DATASET

Method	Backbone	Input Size	FPS	#param.
SSD[7]	VGG16	512 × 512	20.0	25.51M
YOLOv3[6]	Darknet-53	608 × 608	29.7	61.63M
CenterNet[10]	DLA-34	800 × 800	22.5	18.48M
RetinaNet[8]	ResNet-101	800 × 800	12.8	55.61M
FCOS[9]	ResNet-101	800 × 800	14.1	51.23M
RepPoints[30]	ResNet-101	800 × 800	13.1	55.93M
AFDet	DLA-34	800 × 800	20.3	18.58M
AFDet	DLA-34-DCN	800 × 800	17.1	20.29M

our method in Table VII. AFDet (w/ DLA-34) can process 20.3 images per second, while AFDet (w/ DLA-34-DCN) can process 17.1 images per second; this slightly reduced FPS is due to the use of the DCN. The computing performance and numbers of parameters are comparable to those of CenterNet. Although YOLOv3 yielded the best speed performance, the number of parameters and precision using the testing datasets are not satisfactory. FPN-based methods (RetinaNet, FCOS, and RepPoints) yield good performance with respect to accuracy, but they are limited by their parameters and speed. Compared with the above methods, our methods achieve a good speed-accuracy tradeoff.

V. CONCLUSION

In this article, we propose a more accurate and faster detector named AFDet, which consists of a backbone network, CPB, SSB, and BEB. CPB is used to predict the center point of an object. SSB is used to force the network to capture more semantic information of an object but is active only in the training process. BEB can obtain the distance distributions from the center point to the four sides of an object under the supervision of a classification loss. Moreover, the focus loss and gather loss can ensure that the distance distributions are more focused on the correct values, thereby retrieving sharper distributions and leading to higher detection accuracy. Ablation experiments

using the HRRSD dataset verify the effectiveness of the different components in AFDet. Our methods achieve state-of-the-art performance on large and small datasets, such as DIOR, HRRSD, and NWPU VHR-10. Furthermore, AFDet performs better than most commonly used methods due to its simple structure. Hence, our method can achieve a good speed-accuracy tradeoff. In the future, we will consider reducing the redundancy of the head of AFDet to further improve its performance.

REFERENCES

[1] S. Xu, T. Fang, D. Li, and S. Wang, "Object classification of aerial images with bag-of-visual words," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 2, pp. 366–370, Apr. 2010.

[2] H. Sun, X. Sun, H. Wang, Y. Li, and X. Li, "Automatic target detection in high-resolution remote sensing images using spatial sparse coding bag-of-words model," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 1, pp. 109–113, Jan. 2012.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[4] R. Girshick, "Fast R-Cnn," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[6] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv preprintarXiv:1804.02767*.

[7] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Euro. Conf. Comput. Vis.*, 2016, pp. 21–37.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2980–2988.

[9] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.

[10] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[11] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. Euro. Conf. Comput. Vis.*, 2018, pp. 734–750.

[12] G. Cheng, Y. Si, H. Hong, X. Yao, and L. Guo, "Cross-scale feature fusion for object detection in optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 431–435, Mar. 2021.

[13] W. Huang, G. Li, Q. Chen, M. Ju, and J. Qu, "CF2PN: A cross-scale feature fusion pyramid network based remote sensing target detection," *Remote Sens.*, vol. 13, no. 5, 2021, Art. no. 847.

[14] Q. Zhao *et al.*, "M2det: A single-shot object detector based on multi-level feature pyramid network," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 9259–9266.

[15] T. Xu, X. Sun, W. Diao, L. Zhao, K. Fu, and H. Wang, "ASSD: Feature aligned single-shot detection for multiscale objects in aerial imagery," *IEEE Trans. Geosci. Remote Sens.*, early access, 2021, doi: 10.1109/TGRS.2021.3089170.

[16] T. Zhao, N. Liu, T. Celik, and H.-C. Li, "An arbitrary-oriented object detector based on variant gaussian label in remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, early access, 2021, doi: 10.1109/LGRS.2021.3087492.

[17] P. Wang, X. Sun, W. Diao, and K. Fu, "FMMSD: Feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3377–3390, May 2020.

[18] J. Fu, X. Sun, Z. Wang, and K. Fu, "An anchor-free method based on feature balancing and refinement network for multiscale ship detection in sar images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1331–1344, Feb. 2021.

[19] Z. Cui, X. Wang, N. Liu, Z. Cao, and J. Yang, "Ship detection in large-scale SAR images via spatial shuffle-group enhance attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 379–391, Jan. 2021.

[20] H. Guo, X. Yang, N. Wang, and X. Gao, "A centernet model for ship detection in SAR images," *Pattern Recognit.*, vol. 112, 2021, Art. no. 107787.

[21] N. Liu, T. Celik, and H.-C. Li, "Gated ladder-shaped feature pyramid network for object detection in optical remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, early access, 2021, doi: 10.1109/LGRS.2020.3046137.

[22] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS J. Photogrammetry Remote Sens.*, vol. 98, pp. 119–132, 2014.

[23] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS J. Photogrammetry Remote Sens.*, vol. 159, pp. 296–307, 2020.

[24] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu, "Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5535–5548, Aug. 2019.

[25] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "Densebox: Unifying landmark localization with end to end object detection," 2015, *arXiv:1509.04874*.

[26] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 516–520.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[28] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.

[29] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, "Foveabox: Beyound anchor-based object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 7389–7398, 2020.

[30] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "Reppoints: Point set representation for object detection," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 9657–9666.

[31] H. Qiu, Y. Ma, Z. Li, S. Liu, and J. Sun, "Borderdet: Border feature for dense object detection," in *Proc. Euro. Conf. Comput. Vis.*, Springer, 2020, pp. 549–564.

[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Euro. Conf. Comput. Vis.*, Springer, 2020, pp. 213–229.

[33] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Representations*, pp. 1–16, 2021.

[34] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 502–511.

[35] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2888–2897.

[36] X. Li *et al.*, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," *NeurIPS*, vol. 33, 2020, pp. 21002–21012.

[37] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2403–2412.

[38] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 764–773.

[39] L. Liu *et al.*, "On the variance of the adaptive learning rate and beyond," 2019, *arXiv:1908.03265*.

[40] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Euro. Conf. Comput. Vis.*, Springer, 2014, pp. 740–755.

[41] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[42] D. Hwabc, Z. Yue, D. Zcabc, L. Hao, B. Hwa, and S. Xian, "Oriented objects as pairs of middle lines," *ISPRS J. Photogrammetry Remote Sens.*, vol. 169, pp. 268–279, 2020.

[43] Z. Yuan, Z. Liu, C. Zhu, J. Qi, and D. Zhao, "Object detection in remote sensing images via multi-feature pyramid network with receptive field block," *Remote Sens.*, vol. 13, no. 5, 2021, Art.no. 862.

[44] H. J. Lin, Y. Zhou, C.-M Gan. Vong, and Q. Liu, "Novel up-scale feature aggregation for object detection in aerial images," *Neurocomputing*, vol. 411, pp. 364–374, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220309784

[45] X. Lu, Y. Zhang, Y. Yuan, and Y. Feng, "Gated and axis-concentrated localization network for remote sensing object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 1, pp. 179–192, Jan. 2020.