Task-Wise Sampling Convolutions for Arbitrary-Oriented Object Detection in Aerial Images

Zhancho Huang®, Member, IEEE, Wei Li®, Senior Member, IEEE, Xiang-Gen Xia®, Fellow, IEEE, Hao Wang®, and Ran Tao, Senior Member, IEEE

Abstract—Arbitrary-oriented object detection (AOOD) has been widely applied to locate and classify objects with diverse orientations in remote sensing images. However, the inconsistent features for the localization and classification tasks in AOOD models may lead to ambiguity and low-quality object predictions, which constrains the detection performance. In this article, an AOOD method called task-wise sampling convolutions (TS-Conv) is proposed. TS-Conv adaptively samples task-wise features from respective sensitive regions and maps these features together in alignment to guide a dynamic label assignment for better predictions. Specifically, sampling positions of the localization convolution in TS-Conv are supervised by the oriented bounding box (OBB) prediction associated with spatial coordinates, while sampling positions and convolutional kernel of the classification convolution are designed to be adaptively adjusted according to different orientations for improving the orientation robustness of features. Furthermore, a dynamic task-consistent-aware label assignment (DTLA) strategy is developed to select optimal candidate positions and assign labels dynamically according to ranked task-aware scores obtained from TS-Conv. Extensive experiments on several public datasets covering multiple scenes, multimodal images, and multiple categories of objects demonstrate the effectiveness, scalability, and superior performance of the proposed TS-Conv.

Index Terms—Arbitrary-oriented object detection (AOOD), convolutional neural network (CNN), dynamic label assignment, oriented bounding box (OBB), task-wise sampling strategy.

I. INTRODUCTION

Arbitrary-oriented object detection (AOOD) is widely applied in remote sensing scenes [1].

Manuscript received 4 May 2023; revised 17 December 2023; accepted 15 February 2024. Date of publication 27 February 2024; date of current version 1 March 2025. This work was supported by the National Key Research and Development Program of China under Grant 2021YFB3900502.

Fig. 1. (a) IFS problems of AOOD in different subtasks and orientations. (b) IFS problem is exacerbated by the diverse orientation and dense distribution of objects.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
different orientations problems are coupled. Different from the features and predictions of the localization subtask in AOOD that need to change with IFS of different orientations, the features of the classification subtask should be robust to IFS of different orientations without changing for the same category of objects.

Therefore, when the IFS of different subtasks is coupled with IFS of different orientations, the learning goals of different subtasks for IFS are different. Furthermore, objects usually have more diverse orientations and dense distribution in AOOD tasks [11], which also exacerbates the IFS problem. Since objects of the same category have different orientations in the remote sensing overhead perspective, their IFS is not only between different subtasks and categories but also within the same category and subtasks. Besides, since many objects in remote sensing images may be densely distributed [12], the feature-sensitive areas of different objects may interfere with the feature extraction of vanilla convolution. As shown in Fig. 1(b), since the vanilla convolution neighborhood range is fixed, surrounding objects may interfere with feature extraction within the convolution area, i.e., the difference between the two feature-sensitive regions in the two red boxes in Fig. 1(b) becomes larger. This exacerbates IFS, which is not directly due to subtask or orientation differences. In addition, the IFS problem also leads to ambiguity in label assignment due to the different preferences of each candidate for location sensitivity or classification sensitivity.

In response to the IFS problem of different subtasks, RepPoints [13], Oriented RepPoints [14], and so on, sampled localization and classification features by two deformable convolutional networks (DCNs) share the aligned sampling offsets in two CNN branches. However, the positions suitable for sampling localization features may not be suitable for sampling classification features. The forced alignment of sampling offsets for different subtasks may cause some sampling points that are only suitable for one task to be missed. Furthermore, a candidate position for accurate localization may not be optimal for object classification when CNN makes predictions. In this regard, AutoAssign [15], ATSS [16], GGHL [7], and so on indirectly improve the prediction by adjusting the selection or weights of positive and negative samples to optimize the training supervision. However, these methods do not directly change the feature extraction or alignment process, and the IFS problem that existed in the sampling process of the convolutional layer is not eliminated. However, AOOD not only faces IFS of different subtasks but also faces IFS of different directions, as well as the problem of coupling the two IFS together. Existing research does not consider these and other challenges.

Rethinking the above problems and observing Fig. 1, a straightforward idea is to separately extract more appropriate features from different sampling positions that are suitable for different subtasks and different object orientations by different sampling strategies. Following this idea, two more aspects have to be carefully considered: 1) selecting appropriate sampling strategies for extracting task-wise features and 2) designing association constraints for different sampling offsets since they are not entirely independent. Besides, in the design of the task-wise convolutional sampling scheme, the characteristics of objects, including OBB representation, dense distribution, and diverse orientation, need to be carefully considered. Furthermore, when the task-wise features are mapped to the same candidate position for predictions, it requires a dynamic label assignment strategy to provide feedback to the CNN on which candidate positions are suitable. Therefore, motivated by the above considerations, a task-wise sampling convolutions (TS-Conv) method is proposed in this work. The main contributions are summarized as follows.

1) The IFS problem of different subtasks and different orientations in AOOD is comprehensively analyzed. On this basis, the proposed TS-Conv designs more explicitly supervised sampling strategies to extract the task-wise features from appropriate sensitive regions. Furthermore, the sampling strategies, feature alignment processes, and label assignment are unified into a closely related framework to achieve dynamic feedback optimization.

2) In the designed TS-Conv, the localization sampling offsets are directly associated with the spatial coordinate embeddings and the OBB decoding in the detection head, which change with the orientations and sizes of objects. The classification sampling offsets are rotation invariant and can be used with the designed dynamic circular kernel (DCK) to extract orientation-robust features by adjusting the optimal orientation and weights.

3) Based on the predictions of TS-Conv, a dynamic task-consistent-aware label assignment (DTLA) strategy is developed, in which the aligned features are adaptively mapped to positive samples selected by the ranked task-consistent-aware score. It forms a dynamic closed loop of “assignment-sampling-alignment-reassignment” for supervising CNN training to obtain better performance.

The remainder of this article is organized as follows. Section II reviews the related works. Section III presents the proposed TS-Conv in detail. Section IV evaluates and analyzes the performance of TS-Conv through extensive experiments. Section V draws the conclusions and discussions.

II. RELATED WORKS

A. Arbitrary-Oriented Object Detection

Unlike ordinary OD, the objects in AOOD are positioned by OBBS. Thus, additional CNN localization branches with different OBB representation strategies are designed to predict OBBS. ROI Transform [4], SCR-Det [12], R3Det [6], S2ANet [17], and so on predicted rotation angles based on horizontal bounding boxes (HBBS) to obtain OBBS, while Gliding Vertex [5], GGHL [7], and so on directly predicted the four-vertices of each OBB. Oriented RepPoints [14] represented OBBS by a set of points. Distinguishing from the above angle or vertex regression strategies, CSL [18] predicted the rotation angles of OBBS by discrete angle classification. To get rid of the dependence on anchor boxes, BBAVectors [19], O2DNet [20], GGHL [7], and so on developed different anchor-free label assignment strategies, and AO2DETR [21] predicted OBBS by the sequence model without
anchor boxes. For more accurate OBB predictions, GWD [22] and KLD [23] designed new loss functions based on the distances of Gaussian distributions generated from the predicted OBBs and ground truth. CFA [24] proposed a convex-hull feature adaptation method, which optimizes feature assignment by constructing convex-hull sets and dynamically splitting positive and negative convex hulls. DCFL [25] proposed a dynamic prior along with the coarse-to-fine assigner to leverage the coarse prior matching and finer posterior constraint to dynamically assign labels, which provides appropriate and relatively balanced supervision for diverse instances. In response to the IFS problem in AOOD, GGHL [7], CFC-Net [26], and so on decoupled the CNN branches and reweighted task-wise features by different strategies. Oriented RepPoints [14] predicted initial OBBs and categories through aligned features extracted by shared-offset DCNs and then learned additional offsets to refine the initial OBBs. Different from the existing AOOD methods, the proposed TS-Conv separately samples task-wise features without additional reweighting or refinement operation. Besides, TS-Conv associates the task-wise feature sampling with other parts of the AOOD pipeline, including the OBB representation and label assignment.

**B. Solutions for the IFS Problem**

IoU-Net [8] analyzed the IFS problem for the first time, which predicted an additional localization score and aggregated it with the classification score as the final score. Along with this idea of postprocessing at the prediction side, IoU-aware [27], GGHL [7], and so on designed different strategies to obtain prediction scores that consider both localization and classification contributions by reweighting, reranking, or jointly optimizing the scores and so on. Nevertheless, the feature-sensitive positions are still spatial misalignment. Double-Head R-CNN [28], YOLOX [29], and so on decoupled the detection head of CNN into two branches to extract localization and classification features separately. Based on this feature-decoupling scheme, two-stage methods, TSD [9] and D2Det [30], extracted task-suitable features by task-wise Deformable RoI Pooling [31]. One-stage methods, such as Guided Anchoring [32] and VFNet [27], aligned and refined the decoupled features by DCNs [31], [33] or attention mechanism. Furthermore, RepPoints [13] and Oriented RepPoints [14] extracted localization and classification features from the spatially aligned positions in two branches by two DCNs that share the same sampling offsets. In addition, Oriented RepPoints [14] designed a quality assessment and sample assignment strategy for selecting high-quality sampling points and positions. However, the sampling positions of localization and classification features in the above methods are spatially aligned. Inspired by the above contradiction, this work designs separately sampling strategies to solve the IFS problem in AOOD. Moreover, TS-Conv also associates the OBB representation and dynamic label assignment with task-wise feature sampling, which deals with the IFS problem more comprehensively from different perspectives of the AOOD framework.

III. PROPOSED GGHL FRAMEWORK

The AOOD framework based on TS-Conv is shown in Fig. 2, which is mainly composed of three parts: the CNN model and training data, the proposed TS-Conv component consisting of the convolution for sampling localization features (LS-Conv) and the convolution for sampling classification features (CS-Conv), and the designed DTLA strategy.

First, the CNN extracts features through the backbone and FPN and outputs them to the dual branches of localization and classification in the TS-Conv component, marked in blue and yellow in Fig. 2(b). It follows the scheme of decoupling branches in Fig. 1(b). Meanwhile, initial OBBs, marked in red in Fig. 2(a), are predicted as the constraint to supervise the learning of sampling positions of TS-Conv. In response to the problems of the shared-offset DCNs scheme in Fig. 1(c), different sampling offsets constrained by initial OBBs are designed for localization and classification. Then, the convolutions for localization and classification use the designed sampling strategies to extract features from their sensitive regions separately. It allows the task-wise sampling positions of CNN to be supervised by ground truth to adapt to different objects dynamically. Next, aiming at the problem that static label assignment strategies cannot give CNN feedback on assigning candidate positions without the IFS problem for predictions during training, a dynamic label assignment strategy DTLA is designed. In DTLA, the static arbitrary-oriented label assignment strategy GGHL [7] is utilized as the initial assignment of DTLA. Based on the initial assignment, the designed dynamic label assignment strategy, i.e., DTLA, is carried out. As shown in green in Fig. 2(c), DTLA adjusts the positive and negative samples preassigned by the static label assignment method GGHL [7] according to the task-consistent-aware score predicted in each iteration of TS-Conv training. Finally, the CNN is trained to make the loss between CNN’s predictions and labels assigned by DTLA converge, and then, the trained CNN is used to detect objects.
Fig. 3. Principle of the proposed TS-Conv. (a) OBB representation of GGHL [7]. (b) CNN structure of the proposed TS-Conv. (c) Sampling positions of the LS-Conv. (d) Sampling positions of the CS-Conv.

A. Convolution for Sampling Localization Features

From the observation in Fig. 1(a) and the analysis of [9], the feature-sensitive regions for localization are mainly distributed at the object boundary, i.e., the regions near the OBB. Inspired by this observation and the star-shaped box of VFNet [27], associating the feature sampling positions of the LS-Conv with the OBB representation is considered. For OBB representation, GGHL [7] proposed a flexible anchor-free strategy based on Gliding Vertex [5], as shown in Fig. 3(a). It predicts an OBB by predicting the distances $l_n, n = 1, 2, 3, 4,$ from the current candidate object position $q_n(x, y)$ to the four edges of the external HBB and the distances $s_n, n = 1, 2, 3, 4,$ from the four vertices $q_0, q_2, q_6,$ and $q_8$ of the HBB to the four vertices $q_1, q_3, q_5,$ and $q_7$ of the OBB on the corresponding edge. This OBB representation has nine key positions, i.e., the four vertices of the HBB, the four vertices of the OBB, and the current candidate object position. A straightforward idea is to directly correspond these key positions to the nine sampling points of a $3 \times 3$ convolution, as shown in Fig. 3(c). The localization convolution no longer explores the features in the current position and its eight neighborhoods fixedly while adjusting the sampling position according to different objects with different OBBs by DCN [34] dynamically. The details and further improvements are presented as follows.

First, the initial OBBs are predicted by the CNN model trained under the supervision of labels assigned by GGHL [7]. The initial estimations of $l_n$ and $s_n, n = 1, 2, 3, 4,$ are denoted as $\hat{l}_n$ and $\hat{s}_n, n = 1, 2, 3, 4,$ respectively. The second, the spatial coordinates of the sampling points $q_i, i = 0, 1, \ldots, 8,$ are calculated according to the predictions $\hat{l}_n, \hat{s}_n$, and the coordinates of the current convolutional position $(x, y)$. In this case shown in Fig. 3(a), the current sample point is defined as $q_0 = (x, y)$. The four vertices of HBB are defined as $q_0 = (x - \hat{l}_4, y - \hat{l}_4), q_2 = (x + \hat{l}_2, y - \hat{l}_3), q_6 = (x - \hat{l}_4, y + \hat{l}_3)$ and $q_8 = (x + \hat{l}_2, y + \hat{l}_3)$, respectively. The four vertices of OBB are defined as $q_1 = (x - \hat{l}_4 + s_1, y - \hat{l}_1), q_3 = (x - \hat{l}_4, y + \hat{l}_3 - \hat{s}_3), q_5 = (x + \hat{l}_2, y - \hat{l}_1 + \hat{s}_2),$ and $q_7 = (x + \hat{l}_2, y + \hat{l}_3)$. Intuitively, the process of manually labeling the bounding box of an object by human is usually to find the most prominent points on the top, bottom, left, and right of an object and then draw the bounding box based on this range, that is, for an ideal bounding box label that closely surrounds an object, at least one point on each edge intersects the object. Thus, the intersection points between the four sides of the bounding box and the object are the key elements for locating the position and range of an object. Since objects have various shapes and each edge may have more than one intersection point with the object, a sampling point on each of the four sides is selected and set to be adaptive sliding. The sliding processes are controlled by the CNN learnable variables $\sigma_i \in (0, 1), i = 0, 2, 6, 8, \ldots$. These four sliding points are defined as $\hat{q}_0, \hat{q}_2, \hat{q}_6,$ and $\hat{q}_8$, and the coordinates them are listed in Table I. For convenience, let $\tilde{q}_i = q_i, i = 1, 3, 4, 5, 7,$ in the following.

Define the feature map input to the localization branch as $I_{\text{loc}} \in \mathbb{R}^{W \times H \times F}$, where $W$, $H$, and $F$ denote the width, height, and the number of feature maps, respectively. Furthermore, inspired by extracting features from keypoint while letting CNN learn their position information, a spatial coordinate embedding operation is employed on LS-Conv. Generate a tensor $I_{\text{coor} \times x} \in \mathbb{R}^{W \times H \times 1}$. All the elements on each column of $I_{\text{coor} \times x}$ are the same, i.e., the index of this column. Similarly, generate a tensor $I_{\text{coor} \times y} \in \mathbb{R}^{W \times H \times 1}$. All the elements on each row of $I_{\text{coor} \times y}$ are the same, i.e., the index of this row. Define $I_{\text{loc}} \in \mathbb{R}^{W \times H \times (F+2)}$ as the tensor of $I_{\text{loc}}, I_{\text{loc} \times x}$, and $I_{\text{loc} \times y}$, i.e., $I_{\text{loc}}(1 : W, 1 : H, 1 : F) = I_{\text{loc}}, I_{\text{loc}}(1 : W, 1 : H, F + 1) = I_{\text{coor} \times x}$, and $I_{\text{loc}}(1 : W, 1 : H, F + 2) = I_{\text{coor} \times y}$. Define the element of $I_{\text{loc}}$ at $(x, y)$, $x \in [1, W]$ and $y \in [1, H]$, in the first two dimensions as $I_{\text{loc}}(x, y) \in \mathbb{R}^{1 \times 1 \times (F+2)}$, which consists of a position and its associated $F$-dimensional feature vector.

| Points | Coordinates | Binding Conditions |
|--------|-------------|--------------------|
| $\tilde{q}_0$ | $\{ x_0 = x_1, y_0 = y_1 \}$ | $x_1 = x_0 \leq x_2$ |
| $\tilde{q}_2$ | $\{ x_2 = x_1, y_2 = y_1, y_2 + 2x_1 \}$ | $x_2 = x_1 \leq x_2$ |
| $\tilde{q}_6$ | $\{ x_6 = x_4, y_6 = y_4, x_6 + 2y_1 \}$ | $x_6 = x_4 \leq x_7$ |
| $\tilde{q}_8$ | $\{ x_8 = x_6, y_8 = y_4, x_8 + 2y_1 \}$ | $x_8 = x_4 \leq x_8$ |
It allows the CNN to learn the spatial coordinates of the sampled positions directly while extracting the features from sensitive regions [35]. In addition, because the size $F$ of the feature vector satisfies $F \gg 2$ in CNN, the spatial coordinate embedding operation does not add much additional computational burden. Thus, the feature information and coordinates of sampling positions are explicitly and spatially correspond to the feature map and its grid coordinates, and the geometric constraints between sampling positions are also clear and can be represented by the offsets of DCNs. Compared with using completely dynamic sampling positions, the learning goal of CNN is clearer.

For notational convenience, we define the correspondence of the nine elements (positions) of the following set:

$$
\{(x, y) \mid (x, y) \in \{(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0), (1, 1)\}
$$

The indices of $(x, y)$ and its eight neighbors are represented as $(x + \Delta x_{u}^{\text{loc}}, y + \Delta y_{u}^{\text{loc}})$, $u = 0, 1, \ldots, 8$. Define the feature vectors at $(x + \Delta x_{u}^{\text{loc}}, y + \Delta y_{u}^{\text{loc}})$ as $\mathbf{F}_{u}^{\text{loc}}$. Define a $3 \times 3$ filter $K^{\text{loc}}$, where $K_{j, u}$, $j = 0, 1, \ldots, 8$, are the elements and also the coefficients of $K^{\text{loc}}$. When using $K^{\text{loc}}$ to perform LS-Conv$^1$ on $\tilde{I}^{\text{loc}}$, the following two cases are considered. When the position $(x, y)$ is a positive position, which will be explained in Section III-C, used to predict the OBB, as shown in Fig. 3(c), the nine sampling points $(x + \Delta x_{u}^{\text{loc}}, y + \Delta y_{u}^{\text{loc}})$, $u = 0, 1, \ldots, 8$, are correspondingly moved to the points $(x_{\hat{q}_{i}}, y_{\hat{q}_{i}})$, $i = 0, 1, \ldots, 8$, by DCN [33].

The sampled feature vectors are denoted as $\tilde{I}_{x_{\hat{q}_{i}}, y_{\hat{q}_{i}}}$, $i = 0, 1, \ldots, 8$. When the position $(x, y)$ is not a positive position, the sampling positions $(x + \Delta x_{u}^{\text{loc}}, y + \Delta y_{u}^{\text{loc}})$, $u = 0, 1, \ldots, 8$, do not move. Thus, the output of LS-Conv at $(x, y)$ is represented as

$$
O_{x, y}^{\text{loc}} = \left\{ \begin{array} { l l } { \sum_{i,j=0}^{N,8} \tilde{I}_{x_{\hat{q}_{i}}, y_{\hat{q}_{i}}} K_{j, i}^{\text{loc}} m_{j}^{\text{loc}}, } & { \text{if } (x, y) \text{ is positive}} \\ { \sum_{u,j=0}^{N,8} \tilde{I}_{x + \Delta x_{u}^{\text{loc}}, y + \Delta y_{u}^{\text{loc}}} K_{j, u}^{\text{loc}} m_{j}^{\text{loc}}, } & { \text{otherwise}} \end{array} \right.
$$

where $O_{x, y}^{\text{loc}} \in \mathbb{R}^{1 \times (F+2)}$ and CNN-learnable scalars $m_{j}^{\text{loc}} \in (0, 1)$, $j = 0, 1, \ldots, 8$, are employed to adjust the contribution of the features sampled from different positions such as DCNv2 [31]. It further enhances the feature learning capability of LS-Conv by simultaneously adjusting the sampling position and amplitude. For some OBB vertices that do not fall on the object, it can suppress the sampling of low-quality features by decreasing the value of $m_{j}^{\text{loc}}$. Finally, the output features are used to predict the corrections $\Delta \hat{I}_{n}$ and $\Delta \hat{s}_{n}$, $n = 1, 2, 3, 4$, to the initial OBB predictions $\hat{I}_{n}$ and $\hat{s}_{n}$, $n = 1, 2, 3, 4$. The refined OBB predictions are

$$
\hat{I}_{n} = \hat{I}_{n} \times \Delta \hat{I}_{n}
$$

$$
\hat{s}_{n} = \hat{s}_{n} \times \Delta \hat{s}_{n}
$$

where $\hat{s}_{1,2} \in [0, \hat{I}_{1,2}]$ and $\hat{s}_{3,4} \in [0, \hat{I}_{1,2}]$. Note that, due to the significant difference in the sizes of different objects, i.e., the values of $\hat{I}_{n}$ and $\hat{s}_{n}$, (3) uses multiplications rather than additions to make the ranges of $\Delta \hat{I}_{n}$ and $\Delta \hat{s}_{n}$ predicted by the CNN less affected by the object size. In addition, the refined OBB predictions are also supervised by the ground truth during the CNN training. Compared to RepPoints [13], Oriented RepPoints [14], and so on, which first obtain the set of dynamic sampling points and then generate bounding boxes from the point set, the sampling positions of LS-Conv are obtained from OBBs directly. Thus, the receptive field of LS-Conv is always adapted to the object size. It makes the supervision of localization feature sampling in CNN more comprehensive to the final task objective.

### B. Convolution for Sampling Classification Features

Unlike localization feature-sensitive regions directly associated with the OBB representation and spatial coordinates, sensitive regions of classification features are more variable for objects with different categories, shapes, and orientations [36], [37]. Therefore, the sampling positions of the classification convolution are only constrained within the OBB for more flexible adjustment according to each object, as shown in Fig. 3(b). Since the OBB is an arbitrary convex quadrilateral, the constraint range is approximated as the minimum external rectangle (MERect)$^2$ of the OBB in this case in order to facilitate parallel computation in CNN. Define the length of MERect’s long side as $S_{1}$, the length of the other side as $S_{2}$, the center point of MERect as $(x_{c}, y_{c})$, and the angle between the long side and the positive direction of $x$-axis as $\alpha$, $\alpha \in [0, \pi]$ in this case. Define the sampling positions of the CS-Conv as $p_{i}$, $i = 0, 1, \ldots, 8$, and the CNN-learnable variables $\alpha_{i}^{(s)}, \omega_{i}^{(s)} \in (0, 1), i = 0, 1, \ldots, 8$, used to adjust the sampling positions. According to the designed constraint, the coordinates of the sampling positions are represented as

$$
\begin{bmatrix}
    x_{p_{i}} \\
    y_{p_{i}}
\end{bmatrix} =
\begin{bmatrix}
    \cos \alpha & -\sin \alpha \\
    \sin \alpha & \cos \alpha
\end{bmatrix}
\times
\begin{bmatrix}
    x_{c} - 0.5S_{1} + \alpha_{i}^{(s)}S_{1} \\
    y_{c} - 0.5S_{2} + \alpha_{i}^{(s)}S_{2}
\end{bmatrix}
$$

where $(x_{p_{i}}, y_{p_{i}})$ denote the coordinates of the sampling position $p_{i}$, $i = 0, 1, \ldots, 8$. Although the above design allows the sampling positions to translate to the sensitive regions within the OBB adaptively, the sensitive regions are variable due to the arbitrary orientations of objects in AOOD. It brings more significant uncertainty in the sampling position offsets, and CS-Conv lacks absolute-coordinate information such as SCE in LS-Conv. Compared to rotating the sampling positions with more complex constraints, rotating the convolution kernel to the appropriate direction is a more straightforward idea. Thus, a DCK is designed.

First, as shown in Fig. 4(a), a circular kernel $K^{\text{cls}} \in \mathbb{R}^{3 \times 3}$ is generated from the square kernel $K^{\text{cls}} \in \mathbb{R}^{3 \times 3}$ by bilinear interpolation to accommodate rotations in different directions. The coefficients of $K^{\text{cls}}$ are defined as $K_{j, j}$, $j = 0, 1, \ldots, 8$, and the coefficients of $K^{\text{cls}}$ are defined as $K_{j, j}$, $j = 0, 1, \ldots, 8$. When $j = 1, 3, 4, 5, 7$, $K_{j, j} = K^{\text{cls}}_{j, j}$. Second, the circular

---

$^1$Note that, it is conventionally called a convolution operation in CNN, but it is actually a filtering operation.

$^2$OpenCV and many computer vision libraries have implemented this function, so the details are not repeated here.
the idea of CondConv [38]. The fused kernels are represented as
\[ K_{\varphi_{k}} = \begin{cases} \lambda_{j} K_{\varphi_{k}}^{cls} + (1 - \lambda_{j}) K_{\varphi_{k}}^{circ} & \text{if } k = 0, 2, 4, 6 \\ K_{\varphi_{k}}^{circ} & \text{if } k = 1, 3, 5, 7 \end{cases} \] (5)
where \( \lambda_{j} \in (0, 1), j = 1, 2, 3, 4 \), denote the CNN-learnable weights for fusion, whose generation module in CNN is shown in Fig. 4(a). Similarly, the coefficients of \( K_{\varphi_{k}}^{cls} \) are defined as
\[ K_{\varphi_{k}}^{cls} = \begin{cases} (\frac{-\sqrt{2}}{2}, \frac{-\sqrt{2}}{2}), (-1, 0), (\frac{-\sqrt{2}}{2}, \frac{-\sqrt{2}}{2}), (0, -1) \\ (0, 0), (0, 1), (\frac{-\sqrt{2}}{2}, \frac{\sqrt{2}}{2}), (1, 0), (\frac{-\sqrt{2}}{2}, \frac{\sqrt{2}}{2}) \end{cases} \] (6)
The indices of \((x, y)\) and its eight circular neighbors are represented as \((x + \Delta x_{v}^{cls}, y + \Delta y_{v}^{cls}), v = 0, 1, \ldots, 8\). Define the feature map input to the classification branch as \( I^{cls} \in \mathbb{R}^{H \times W \times F} \). The elements of \( I^{cls} \) at \((x + \Delta x_{v}^{cls}, y + \Delta y_{v}^{cls}), v = 0, 1, \ldots, 8\), are defined as \( I^{cls}_{\varphi_{k}} \), which are \( F \)-dimensional feature vectors. When the position \((x, y)\) is a positive position, as shown in Fig. 3(d), the nine sampling points \((x + \Delta x_{v}^{cls}, y + \Delta y_{v}^{cls}), v = 0, 1, \ldots, 8\), are correspondingly moved to the points \((x_{p}, y_{p}), i = 0, 1, \ldots, 8\), by CNN [33]. The sampled feature vectors are denoted as \( I_{\varphi_{k}}^{cls, i}, i = 0, 1, \ldots, 8 \). When the position \((x, y)\) is not a positive position, the sampling points \((x + \Delta x_{v}^{cls}, y + \Delta y_{v}^{cls}), v = 0, 1, \ldots, 8\), do not move. Thus, the output of CS-Conv at \((x, y)\) is represented as
\[ O_{x,y}^{cls} = \sum_{k=0}^{M} \sum_{j=0}^{N} \sum_{i,j}^{8} I_{\varphi_{k}}^{cls, i} \tilde{K}_{j}^{cls} \beta_k m_j^{cls}, \] (7)
where \( O_{x,y}^{cls} \in \mathbb{R}^{1 \times 1 \times F} \), and CNN-learnable scalars \( \beta_k, k = 0, 1, \ldots, 7 \), are used to reweight the features extracted by \( \tilde{K}_{\varphi_{k}}^{cls} \) with different orientations, as shown in Fig. 4(b).

C. Dynamic Task-Consistent-Aware Label Assignment

For the CNN-based OD, an object may correspond to more than one candidate detection position in the feature maps. It is significant to select the appropriate positive and negative positions from them to assign labels for CNN training. The label assignment strategy has been studied in many works such as ATSS [16] and Autoassign [15]. In the field of AOOD, GGHL [7] selected candidate positions based on Gaussian heatmaps and OBB prediction scores, as shown in Fig. 5(a). However, on the one hand, it uses a hard-thresholding selection strategy, i.e., the candidate positions with the value of Gaussian heatmap score \( F_{x,y} \) higher than the threshold \( T \) are positive and the other positions are negative. See Appendix B (Supplementary Materials) for the derivation of \( F_{x,y} \). The fixed candidate regions may lead to misassignment due to the irregularity and variety of object shapes. On the other hand, a location closer to the center of the Gaussian heatmap may not be a better positive position. Only using localization score to adjust the weights of positive positions in GGHL [7] still faces the IFS problem that these positive positions may not be optimal for both localization and classification tasks. In response, the DTLA strategy is designed based on TS-Conv and GGHL [7]. DTLA divides the regions of candidate positions into positive, negative, soft-negative, and ignored regions denoted as the position sets \( R_{pos}, R_{neg}, R_{sneg}, \) and \( R_{ig} \). These regions are dynamically adjusted according to the localization and classification prediction costs to obtain the optimal position combinations for CNN training. The detailed implementation of DTLA is given in Algorithm 1.

1) Positive Positions: Different from GGHL [7], which treats all the positions in a fixed range of a Gaussian ellipse region as positive positions, the proposed DTLA ranks the candidate positions of each object and takes the Top-\( P \) positions as candidate positions. Since the performance of AOOD is usually evaluated by the product of localization and classification scores, the straightforward idea is to rank the positive candidate positions according to the combined score \( D_{x,y} \) of localization and classification.
First, the static arbitrary-oriented label assignment strategy GGHL [7] is utilized as the initial assignment of DTLa. Based on the initial assignment, the designed dynamic label assignment strategy is carried out. If a position $D_{x,y}$ lies in the Gaussian region, $F_{x,y} > T$, and the ranking of $D_{x,y}$ is within the Top-$P$, then this position is positive for assigning the label to predict the object, $(x, y) \in R_{pn}$. In this case, we set $T = 0.3$. Benefiting from the designed TS-Conv, localization and classification features are extracted from their sensitive regions and mapped to the same position, as shown in Fig. 5(b). The optimal task-wise scores are spatially aligned for determining positive candidate positions. Thus, the localization and classification combined score at $(x, y)$ is defined as

$$D_{x,y} = \begin{cases} \hat{\theta} F_{x,y} + (1 - \hat{\theta}) \sqrt{L_{x,y} C_{x,y}}^{(b)}, & \text{if } (x, y) \in R_{gh} \\ 0, & \text{otherwise} \end{cases}$$

where $C_{x,y}^{(b)} \in (0, 1)$ indicates the predicted classification score that the object belongs to the ground truth category $h$ and will be specified later. The localization score is represented as

$$L_{x,y} = e^{-\text{Loss}_{loc}^{x,y}}$$

$$\text{Loss}_{loc}^{x,y} = 1 - \text{GluU}(l_{x,y}, \tilde{l}_{x,y}) + \text{mse}(s_{x,y}, \tilde{s}_{x,y}) + (\hat{a}_{x,y} - \tilde{a}_{x,y})^2$$

where $\text{Loss}_{loc}^{x,y}$ denotes the OBB localization loss at $(x, y)$. $L_{x,y} \in (0, 1)$ is monotonically decreasing with $\text{Loss}_{loc}^{x,y}$, $\tilde{l}_{x,y} = [\tilde{l}_1, \tilde{l}_2, \tilde{l}_3, \tilde{l}_4]$ and $\tilde{s}_{x,y} = [\tilde{s}_1, \tilde{s}_2, \tilde{s}_3, \tilde{s}_4]$ represent the predictions of an OBB based on the output localization features $O_{x,y}^{loc}$ of TS-Conv. $l_{x,y} = [l_1, l_2, l_3, l_4]$ and $s_{x,y} = [s_1, s_2, s_3, s_4]$ represent the ground truth of this OBB. $\hat{a}_{x,y}$ and $\tilde{a}_{x,y}$ denote the ground truth and predictions of the area ratio of the OBB and HBB, respectively. $\text{GluU}()$ is the function to measure the HBB localization accuracy by generalized intersection over union (GluU) [39], $\text{mse}()$ represents the mean square error (mse) function. See GGHL [7] for details of $\text{Loss}_{loc}^{x,y}$.

The above strategy takes full advantage of the proposed TS-Conv that can extract and map the most sensitive features of different tasks to the same spatial location. On the one hand, it helps to select the optimal candidate positions suitable for both localization and classification dynamically according to different objects in different scenes. On the other hand, the low-quality candidate positions due to MERect approximation can be filtered according to DTLa. Thus, it is unnecessary to generate the Gaussian probability density function (pdf) of any convex quadrilateral but only needs to replace it with the Gaussian PDF based on MERect, which makes the algorithm more efficient and concise.

2) Negative Positions: The designed DTLa uses a soft thresholding strategy instead of treating all positions with $F_{x,y} < T$ as negative, as shown in Fig. 5(b). If $(x, y)$ does not lie in the Gaussian region, this position is considered as soft negative position, $(x, y) \in R_{neg}$. The background prediction loss of these positions is multiplied by the weight $w_{neg} = 1 - D_{x,y}$, $w_{neg} \in (0, 1)$. The smaller $D_{x,y}$, the larger the weight, indicating a higher negative attribute for this position.

3) Ignored Positions: If $(x, y)$ lies in the Gaussian region $R_{gh}$, $F_{x,y} > T$, but $D_{x,y}$ is not within the Top-$P$, this position may not satisfy both localization and classification tasks, although it has a high priori score. If $(x, y)$ lies in the Gaussian region $R_{gh}$, $F_{x,y} < T$, but $D_{x,y} > T$, this position is too close to the junction region between the object and the background for a low priori score, although it obtains a high $D_{x,y}$. In the above two cases, the priori score $F_{x,y}$ and the dynamic score $D_{x,y}$ contradict each other. It is not appropriate to treat $(x, y)$ as either positive or negative position, so it is ignored and not used for CNN training, i.e., $(x, y) \in R_{ig}$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
4) Loss Functions: If a position \((x, y)\) is positive, its assigned ground truth of objectness \(\text{obj}_{x, y} = L_{x, y}\); if \((x, y)\) is negative, \(\text{obj}_{x, y} = 0\). Note that the CNN gradient of \(L_{x, y}\) is not backpropagated. Define the CNN predicted objectness score at \((x, y)\) as \(\hat{\text{obj}}_{x, y} \in (0, 1)\). According to the assignments of different candidate positions, the binary loss function for objectness prediction is represented as

\[
\text{Loss}_{\text{obj}} = -\frac{1}{M_{\text{pos}}} \sum_{(x, y) \in R_{\text{pos}}} (L_{x, y} - \hat{\text{obj}}_{x, y})^y \log \hat{\text{obj}}_{x, y}
- \frac{1}{M_{\text{neg}}} \sum_{(x, y) \in R_{\text{neg}}} \hat{\text{obj}}_{x, y} \log(1 - \hat{\text{obj}}_{x, y})
- \frac{1}{M_{\text{neg}}} \sum_{(x, y) \in R_{\text{neg}}} \sum_{h \in C_x} w_{\text{neg}} \hat{\text{obj}}_{x, y} \log(1 - \hat{\text{obj}}_{x, y})
\tag{12}
\]

where \(M_{\text{pos}}, M_{\text{neg}},\) and \(M_{\text{neg}}\) mean the numbers of positive, negative, and soft-negative positions for an input image, respectively. The hyperparameter of Focal Loss \([40]\) \(\gamma = 2\), which is the same as GGHL \([7]\). The OBBs and category labels of the objects are assigned to positive positions filtered by the TOP-P strategy for supervised CNN predictions for object localization and classification. According to \((10)\), the localization loss is calculated as

\[
\text{Loss}_{\text{loc}} = \frac{1}{M_{\text{pos}}} \sum_{(x, y) \in R_{\text{pos}}} (\text{Loss}_{\text{obj}}^{\text{init}} + \text{Loss}_{\text{loc}}^{\text{loc}})
\tag{13}
\]

which represents the OBB localization loss of the initial stage, which is used to supervise the sampling points of TS-Conv.

Define the assigned ground truth of the \(h\)th category at \((x, y)\) as \(C_{x, y}^{(h)}\). If the object at \((x, y)\) belongs to the \(h\)th category, \(C_{x, y}^{(h)} = 1\); otherwise, \(C_{x, y}^{(h)} = 0\). Define the CNN prediction of \(C_{x, y}^{(h)}\) as \(\hat{C}_{x, y}^{(h)} \in (0, 1)\). The classification loss is calculated as

\[
\text{Loss}_{\text{cls}} = \frac{1}{M_{\text{pos}}} \sum_{(x, y) \in R_{\text{pos}}} \sum_{h = 1}^{M_C} (C_{x, y}^{(h)} \log(\hat{C}_{x, y}^{(h)}))
+ (1 - C_{x, y}^{(h)}) \log(1 - \hat{C}_{x, y}^{(h)}))
\tag{14}
\]

where \(M_C\) represents the total number of categories of objects. The total loss is stated as

\[
\text{Loss} = \text{Loss}_{\text{obj}} + \text{Loss}_{\text{loc}} + \text{Loss}_{\text{cls}}
\tag{15}
\]

which represents the sum of the objectness loss, localization loss, and classification loss.

IV. EXPERIMENTS AND DISCUSSION

In this section, extensive experiments are conducted on several AOOD datasets to evaluate the performance of TS-Conv comprehensively. First, the experimental conditions, including datasets, evaluation metrics, and implementation details, are introduced. Then, ablations experiments are designed to validate the effectiveness of each component of the TS-Conv model and to evaluate their performance quantitatively. The task-wise samplings, rotation-invariant feature extraction, dynamic label assignment, and other previously claimed issues in AOOD are discussed in detail. Furthermore, the scalability of TS-Conv on lightweight models and for multimodal data is evaluated. After that, comparative experiments with existing AOOD methods on the datasets covering different scenarios are given and analyzed. Besides, the extension of the proposed TS-Conv to improve the performance of lightweight AOOD models on embedded devices is further explored.

A. Experimental Conditions

1) Experimental Platforms: The experiments are performed on a server with an AMD 3950WX CPU, 128-GB memory, and four NVIDIA GeForce RTX 3090 GPU (24 GB). In addition, the performance of the TS-Conv improved lightweight models is evaluated on embedded edge devices NVIDIA Jetson AGX Xavier and Jetson TX2.

2) Datasets: To evaluate the performance of TS-Conv model more comprehensively, several public AOOD datasets are used, which cover different scenes, different shapes and categories of objects, and different data sources.

1) The DOTAv1.0 \([1]\) dataset provides a widely used benchmark to evaluate the performance of AOOD methods in remote sensing scenes. It has more than 188,000 objects covering 15 categories in 2806 images from 800 × 800 pixels to 4000 × 4000 pixels. Due to the huge size of remote sensing images, they are usually cropped into subimages of 800 × 800 pixels with an overlap of 200 pixels on each dimension after being scaled to different sizes with the ratios of 0.5, 1.0, and 1.5 \([41]\). DOTAv2.0 \([42]\) further expands the number of objects to 1,793,658 objects covering 18 categories.

2) HRSC2016 \([43]\) is a ship detection dataset consisting of 436 training images, 181 validation images, and 444 testing images from 300 × 300 pixels to 1500 × 900 pixels.

3) DIOR-R \([44], [45]\) dataset is an aerial AOOD dataset containing 190,288 objects covering 20 categories in 23,463 images with the size of 800 × 800 pixels. In the DIOR-R dataset, 5862 images are used for training, 5863 images are used for validation, and 11,738 images are used for testing.

4) DroneVehicle \([46]\) is an infrared–RGB vehicle detection dataset. After screening and preprocessing, 17,900 infrared–RGB image pairs are used for training, 1469 infrared–RGB image pairs are used for validation, and 8980 infrared–RGB image pairs are used for testing.

5) SSDD+ \([47]\) is a synthetic aperture radar (SAR) dataset for ship detection. It has 1160 ship images, including 2456 instances collected from different sea conditions. The ratio of training, validation, and testing images is 7:1:2.

3) Evaluation Metrics: The mean average precision (mAP) is adopted for evaluating the detection accuracy. The mAP with an IoU threshold of 50% is represented as mAP_{50}. mAP_{50,95} means calculating the mAP when the IoU threshold is 50%–95% at every 5% interval and then taking their mean as mAP_{50,95}. The speed is evaluated by frames per second (fps) and the computational complexity is evaluated by the floating-point operations (FLOPs) for lightweight models.
The initial learning rate for training DOTA, DIOR-R, HRSC2016, and SSDD+ datasets is $5 \times 10^{-4}$, and the final learning rate is $1 \times 10^{-6}$. The optimizer for training the CNN model is stochastic gradient descent (SGD), and the learning rate scheduler is cosine decay. The weight decay and the momentum are set as $5 \times 10^{-4}$ and 0.9, respectively. The batch size is 32 (eight images per GPU). The maximum training epochs for DOTA, DIOR-R, HRSC2016, DroneVehicle, and SSDD+ datasets are 36, 150, 150, 50, and 150, respectively. The nonmaximum suppression (NMS) threshold is 0.4. Random cropping, random flipping, random rotation, and mixup strategies are employed for data augmentation. The experiments of the proposed TS-Conv on the DOTA datasets use multiscale cropped images as the training set but use single-scale cropped images as the testing set. The multiscaling data augmentation is not used during model training and testing. The experiments of the proposed TS-Conv on other datasets all adapt single-scale training and single-scale testing strategies.

B. Ablation Experiments

The results of the ablation experiments for different components of TS-Conv, including LS-Conv, CS-Conv with DCK, and DTLa, are presented in Table II. The anchor-free AOOD method GGHL [7] is chosen as the baseline, and experiments are conducted on the most widely used AOOD dataset DOTA v1.0 [1].

1) Ablation Experiments of Each Component: First, as a control group, the shared-offset DCNs employed by RepPoints [13] and Oriented RepPoints [14] are introduced based on GGHL. This design has the same sampling positions for localization and classification features and maps these features to aligned locations. Although the above scheme somewhat enables more objects to be detected and improves the mAP\textsubscript{50} by 0.82, the improvement of the stricter metrics, i.e., mAP\textsubscript{75} and mAP\textsubscript{50:95}, is slight. It implies that the model may relax its requirements for detection quality to accommodate two different tasks.

Correspondingly, the proposed task-wise DCNs consisting of LS-Conv and CS-Conv are evaluated. With the introduction of LS-Conv, the mAP\textsubscript{75} is improved by 4.96% (+2.19) compared to the baseline and 3.63% (+1.62) compared to the shared-offset DCNs. TOOD [48], which also adopts task-specific feature learning ideas, is also compared. The results show that TS-Conv, which uses more explicit constraints and considers directional robustness, performs better than TOOD, which uses reweighted implicit feature extraction constraints. The designed LS-Conv further improves the localization accuracy compared to the shared-offset DCNs by directly associating the convolutional sampling points with the OBB representation and embedding the spatial coordinates into the features. When only CS-Conv is used without LS-Conv, i.e., the sampling points of convolutions are just constrained in OBB, the mAP\textsubscript{50}, mAP\textsubscript{75}, and mAP\textsubscript{50:95} are also increased, but the improvement, especially for mAP\textsubscript{75}, is minor. The bottleneck of this scheme is mainly in inaccurate localization. Because the spatial location coordinates of the OBB are explicit and essential features for the localization task, the convolutional sampling constraint of CS-Conv is vague for localization, although it is suitable for classification. In addition, since the objects in AOOD have richer orientation variations, performance improvements are also observed after replacing the convolutional kernel of CS-Conv with the designed DCK. Furthermore, when LS-Conv and CS-Conv are combined to obtain the TS-DCN scheme, significant performance gains are seen compared to using them alone as listed in Table II. The mAP\textsubscript{50}, mAP\textsubscript{75}, and mAP\textsubscript{50:95} are increased by 1.53% (+1.18), 5.43% (+2.40), and 4.06% (+1.80), respectively. It illustrates that the LS-Conv and CS-Conv play complementary roles for each other. LS-Conv compensates for the lack of localization accuracy using only CS-Conv, while CS-Conv with DCK further improves the detection performance.

Then, the performance of the two label assignment strategies, GGHL [7] and the proposed DTLa, is evaluated. The proposed DTLa increases the mAP\textsubscript{50}, mAP\textsubscript{75}, and mAP\textsubscript{50:95} without introducing additional CNN structure and adding inference cost. Fig. 6 shows the positive candidate positions selected by GGHL [7] and DTLa for predicting objects. As seen in Fig. 6, there is a significant difference between the positive candidate positions dynamically selected by DTLa according to $\tilde{D}_{x,y}$ and the positions statistically assigned by

| Methods                        | Decoupled Head | DCNs | Task-wise DCNs (TS-DCN) | Label Assignment | mAPs on the DOTA v1.0 Dataset | Speed (fps) |
|--------------------------------|----------------|------|-------------------------|------------------|-------------------------------|-------------|
| Baseline (GGHL [7])            | ✔              |      |                         |                  | mAP\textsubscript{50}       |             |
| TOOD + DCNs [48]               | 1-head         |      |                         |                  | mAP\textsubscript{75}       |             |
| Shared-offset DCNs             | ✔              | shared-offsets    |                         |                  | mAP\textsubscript{50:95}   |             |
| DCNs without shared-offsets    | ✔              | w/o shared-offsets|                         |                  | mAP\textsubscript{75}       |             |
| LS-Conv                        | ✔              |      |                         |                  | mAP\textsubscript{50}       |             |
| CS-Conv                        | ✔              |      |                         |                  | mAP\textsubscript{75}       |             |
| CS-Conv with DCK               | ✔              |      |                         |                  | mAP\textsubscript{75}       |             |
| DTLa                           | ✔              |      |                         |                  | mAP\textsubscript{75}       |             |
| TS-DCN                         | ✔              |      |                         |                  | mAP\textsubscript{75}       |             |
| TS-Conv                        | ✔              |      |                         |                  | mAP\textsubscript{75}       |             |

Note: Bold indicates the best result. 'Shared-offset DCN' represents that both branches of the decoupled head use DCN [33] and their sampling offsets are shared as in the case of RepPoints [13] and Oriented RepPoints [14]. The convolutions of TOOD’s detection head also use DCNs for fair.
Fig. 6. (a) Comparison of DTLA and GGHL label assignment strategies. (b)–(d) Positive candidate positions and their scores statically assigned by the GGHL strategy at three different scales. (e)–(g) Positive candidate positions and their scores dynamically assigned by the proposed DTLA strategy at three different scales. The closer the color is to red, the higher the score.

Fig. 7. Visualization results of task-wise prediction scores.

Table III
Evaluations of Different Label Assignment Strategies on the DOTA Datasets

| Models    | Label Assign. | mAP50 | mAP75 | mAP50:95 | Inference Speed (fps) |
|-----------|---------------|-------|-------|----------|-----------------------|
| GGHL[7]  | Anchor[49]    | 74.64 | 38.77 |          |                       |
| GGHL[7]  | Centerness[50] | 73.48 | 42.39 |          |                       |
| GGHL[7]  | ATSS[16]      | 75.15 | 39.06 |          |                       |
| GGHL[7]  | AutoAssign[15] | 75.34 | 42.39 |          |                       |
| GGHL[7]  | TAL+TAP of TOOD[48] | 77.11 | 42.39 |          |                       |
| GGHL[7]  | DTLA          | 77.98 | 42.39 |          |                       |
| LO-Det[41] | Anchor        | 66.17 | 60.01 |          |                       |
| LO-Det[41] | Centerness[50] | 69.65 | 62.07 |          |                       |
| LO-Det[41] | ATSS[16]      | 70.01 | 60.03 |          |                       |
| LO-Det[41] | AutoAssign[15] | 70.74 | 62.07 |          |                       |
| LO-Det[41] | GGHL[7]       | 71.26 | 62.07 |          |                       |
| LO-Det[41] | TAL+TAP of TOOD[48] | 71.08 | 62.07 |          |                       |
| LO-Det[41] | DTLA          | 73.36 | 62.07 |          |                       |

GGHL [7] according to the Gaussian prior. The Gaussian peak position is not necessarily the optimal candidate position. It is more reasonable for DTLA to select positive candidate positions adaptively according to different objects and training stages. Besides, Table III also compares the proposed DTLA with other label assignment strategies used for GOD. The experimental results show that DTLA brings more benefits compared to other label assignment strategies. Furthermore, Fig. 7 shows the visualization results of task-wise prediction scores, from which it can be seen that the proposed TS-Conv with DTLA has better localization and classification prediction consistency.

Finally, the TS-Conv model combining the above components is obtained. Compared with the GGHL model, the mAP50, mAP75, and mAP50:95 of TS-Conv are improved by 2.34% (+1.80), 5.45% (+2.41), and 4.47% (+1.98). The improvement in mAP75 and mAP50:95 metrics representing higher detection quality is more pronounced. In general, the ablation experiments validate the effectiveness of each component in the proposed TS-Conv. In addition, Table IV lists the experimental results of different hyperparameter settings. The hyperparameter $\vartheta$ is used to measure the weight of statically preallocated labels versus dynamically allocated labels. The larger its value is, the more dependent it is on the label assignment of GGHL, which conforms to the Gaussian prior; the smaller its value is, the greater its reliance on the CNN adaptive learning score for label assignment. According to Table IV, we set $T = 0.3$ and $\vartheta = 0.5$ in TS-Conv to obtain the best results.

2) Analysis of the IFS Problem: To more intuitively analyze the IFS problem faced by existing AOOD models such as GGHL [7], Fig. 8 visualizes the feature-sensitive regions of the localization and classification tasks. Fig. 8(a-1), (a-2), (b-1), and (b-2) Visualization results of feature sensitivity regions before using the designed TS-Conv. (a-3), (a-4), (b-3), and (b-4) Visualization results of feature sensitivity regions after using the designed TS-Conv.

Table IV
Evaluations of Detection Performance for Different Values of Hyperparameters $T$ and $\vartheta$, and Experiments of Using DCK and RRDA on the DOTAV1.0 Dataset

| $T$  | mAP50 | $\vartheta$ | mAP75 | RandomRota | DCK | mAP50:95 |
|------|-------|-------------|-------|------------|-----|----------|
| 0.2  | 74.45 | 0.3         | 75.75 |            |     | 77.92    |
| 0.3  | 78.75 | 0.4         | 78.52 |            |     | 78.47    |
| 0.4  | 78.60 | 0.5         | 78.50 |            |     | 78.75    |

Note: Bold indicates the best result. RandomRota represents using the random rotation data augmentation operation during the CNN training. When evaluating one variable, the other variables are fixed.
using the proposed TS-Conv. Sensitive features of different tasks located at different locations before input to TS-Conv are extracted and mapped to the same spatial location. The sensitive features of localization and classification are spatially aligned after TS-Conv’s mapping. Then, the optimal candidates are found among these spatially aligned features to predict the objects. It further demonstrates the effectiveness of the designed TS-Conv. Furthermore, Fig. 9 visualizes the sampling positions (white circles in the figure) of the model for localization and classification subtasks after using the designed sampling strategies. It can be seen from the visualization results that the feature sampling positions extract features from the corresponding task-sensitive regions, which shows the effectiveness of the task-wise sampling strategies in extracting more appropriate features. Besides, the localization feature sampling positions under explicit constraints also match the sensitive areas, indicating that the designed explicit constraints can effectively associate the feature-sensitive positions with the representation of OBB. In addition, it is found that for large-scale objects, such as the ship in Fig. 9, the range of explicitly constrained sampling positions may be smaller than the object OBB during testing (not directly supervised by the ground truth of OBB during training), although these sampling positions still fit the shape and sensitive region of the OBB. This may be due to insufficient receptive field of the localization branch, which needs further study in the future.

3) Analysis of Orientation-Robust Features Extracted by DCK: Fig. 10(a)–(h) shows the feature-sensitive regions of different oriented convolutional kernels in DCK (see details in Fig. 4). The different feature-sensitive regions indicate that the feature extraction is not robust to arbitrary orientations. Since objects’ orientations in AOOD are more diverse than those in the ordinary OD, this problem further constrains the existing AOOD models’ performance. Fig. 10(i) shows the feature-sensitive regions of the designed DCK, which adjusts the optimal orientation and weights of eight-oriented kernels according to different inputs adaptively. Furthermore, Fig. 10(i)–(l) shows the feature-sensitive regions of DCK when the input images are rotated in different orientations. The results indicate that the feature-sensitive regions of DCK are robust to arbitrary-oriented inputs, which do not fluctuate significantly with the objects’ rotations. In addition, the performance of using the random rotation data augmentation (RRDA) strategy and DCK is compared in Table IV. The performance of the model using only DCK exceeds that of the model using only the RRDA strategy. As shown in Table IV, when using both the RRDA strategy and DCK, the mAP is 0.28 higher than that of using DCK only and 0.83 higher than that of using RRDA only. The performance further confirms the designed DCK’s ability to enhance the model’s robustness to extract features with arbitrary orientations.

C. Experiments for the Scalability of TS-Conv

1) Scalability of TS-Conv on Lightweight Models: Although TS-Conv improves detection performance and robustness, the introduction of DCNs also brings additional computational burdens compared to the baseline model GGHL [7]. For the benefits of TS-Conv to be applied in the lightweight model, the experiments on different embedded devices are designed and validated, as listed in Table V. The lightweight AOOD model LO-Det [41] is chosen as the baseline. First, the anchor-based label assignment strategy of LO-Det is improved to GGHL and the proposed DTLA for comparison. The results in Table V demonstrate that using the proposed dynamic label assignment strategy DTLA further improves the performance of the lightweight model by 2.95% (+2.10) without losing model inference efficiency and increasing model complexity compared to using the static label assignment strategy GGHL. Second, knowledge distillation is adopted to make DTLA take full advantage of the task-wise sensitive features learned by TS-Conv without complicating the lightweight model structure. Specifically, the LO-Det+DTLA is utilized as the student model, and the TS-Conv is used as the teacher model. RGB images and IR images are used to train one teacher model each, and then, two teacher models are used to distill knowledge to the student model according to the

| Modules | Speed1 (fps) | Speed2 (fps) | Speed3 (fps) | Speed4 (fps) | FLOPs (G) | Parameters (MB) |
|---------|-------------|-------------|-------------|-------------|-----------|----------------|
| YOLOXNano [29] | 64.26 | 64.70 | 6.95 | 52.50 | 3.59 | 15.87 | 0.91 |
| NanoDet-M [29] | 63.84 | 52.87 | 7.51 | 23.24 | 3.49 | 15.11 | 0.99 |
| YOLOv6Nano [29] | 70.82 | 49.53 | 6.40 | 19.72 | 3.01 | 15.87 | 4.30 |
| LO-Det [41] (Baseline) | 66.17 | 60.01 | 6.99 | 22.12 | 3.71 | 6.42 | 6.93 |
| LO-Det [41] + GGHL | 71.26 | 62.07 | 7.68 | 23.72 | 4.04 | 6.30 | 6.72 |
| LO-Det [41] + DTLA | 73.36 | 62.07 | 7.68 | 23.72 | 4.04 | 6.30 | 6.72 |
| TS-Conv Lite | 73.96 | 62.07 | 7.68 | 23.72 | 4.04 | 6.30 | 6.72 |

Note: The unit G is Giga, which represents $1 \times 10^9$. The unit MB represents $1 \times 10^{10}$ bytes. Speed1, Speed2, Speed3 and Speed4 are the detection speed on the RTX 3090 GPU, NVIDIA Jetson TX2, Jetson AGX Xavier, and Jetson Nano, respectively. The inference speed only includes the network inference speed without post-processing. TS-Conv Lite: TS-Conv Distilled LO-Det [41] + DTLA.
The results are shown in Table VI and Fig. 11. The performance of the proposed TS-Conv and the baseline model GGHL is evaluated on the multimodal AOOD dataset DroneVehicle. The results show that the proposed TS-Conv Lite has a better performance and faster detection speed on embedded devices. TS-Conv Lite's mAP50 is further improved without additional inference cost. The performance of TS-Conv Lite’s mAP50 improves 11.77% (+7.79) compared to the baseline model with fewer model parameters and faster inference because it does not rely on anchor boxes. In addition, experiments also compare the performance of TS-Conv Lite and other lightweight models. The results show that TS-Conv Lite has significantly better performance than state-of-the-art methods.

D. Comparison Experiments

In this section, the performance of the proposed TS-Conv and state-of-the-art methods is compared on several datasets, including DOTA [1, 42], DIOR-R [45], HRSC2016 [43], SSD+ [47], and SKU-110KR [34].

1) Comparison Experiments on the DOTA Datasets: Table VII provides the performance comparison results of the different methods on the most widely used AOOD dataset DOTAv1.0. The experimental results show that the performance and speed of the proposed TS-Conv outperform most of the AOOD methods (single scale: mAP50 = 78.75 and multiscale: mAP50 = 80.97), further validating the effectiveness of TS-Conv. Although the performance of TS-Conv (single-scale testing) is slightly lower than that of the two-stage method Oriented R-CNN [53] and the refine-stage method S^2ANet (with the larger backbone ResNet-101) [17] when single-scale testing is used, the detection speed of TS-Conv is much faster than those of Oriented R-CNN [53] and S^2ANet [17]. Furthermore, after using multiscale training and testing, the proposed TS-Conv has better mAP performance than RDet+GWD (refine-stage), R^2Det-KLD (refine-stage), CGCDet (two-stage), DODet (two-stage), and other methods. Its mAP is only slightly lower than that of the two-stage method Oriented R-CNN [53] and the refine-stage method S^2ANet [17]. Furthermore, after using multiscale training and testing, the proposed TS-Conv has better mAP performance than RDet+GWD (refine-stage), R^2Det-KLD (refine-stage), CGCDet (two-stage), DODet (two-stage), and other methods. Its mAP is only slightly lower than RDet+KFIoU (ResNet152) by 0.06%, but TS-Conv is significantly faster than these comparison methods. Besides, the proposed TS-Conv is an anchor-free method, which is more flexible and does not rely on many hyperparameters of anchor boxes. In addition, the performance of the lightweight model TS-Conv Lite (mAP50 = 73.96) can reach the level of many larger models and has a detection speed that far exceeds that of other methods. Furthermore, the performance evaluation on the latest versions of the DOTA datasets, i.e., DOTAv1.5 and DOTAv2.0 [42], is listed in Table VIII. These datasets cover a wider category of objects and more small objects that are difficult to detect. The results also demonstrate the performance advantage of the proposed TS-Conv over existing methods.

2) Comparison Experiments on Other Datasets: The results of comparison experiments on the HRSC2016 dataset are listed in Table IX. TS-Conv also performs better than the existing methods in ship detection, where the aspect ratio of
TABLE VII

| Methods                  | Backbone | Stage | Anchor | BD | GTF | SV | LH | ST | VC | RC | BC | STB | RA | HI | IC | mAP 75% | mAP 50% | mAP 30% | Speed (fps) |
|--------------------------|----------|-------|--------|----|----|----|----|----|----|----|----|-----|----|----|----|--------|--------|--------|-------------|
| R-101                    | AB       |       |        |    |    |    |    |    |    |    |    |     |    |    |    | 58.88  | 56.80  | 54.10  | 38.36      |
| R-152                    | AB       |       |        |    |    |    |    |    |    |    |    |     |    |    |    | 57.25  | 55.26  | 52.75  | 36.15      |
| R-200                    | AB       |       |        |    |    |    |    |    |    |    |    |     |    |    |    | 55.54  | 53.54  | 51.05  | 34.26      |
| O-DNet [30]              | H-104    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 59.28  | 57.24  | 54.52  | 37.24      |
| BBAVectors [19]         | R-101    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| CPC [11]                 | R-101    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| Red [56]                | R-101    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| GGHL [7]                | D-53     | Three |        |    |    |    |    |    |    |    |    |     |    |    |    | 55.64  | 52.97  | 36.11  | 31.20      |
| PointNet [37]           | R-101    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| GWDV [52]               | R-152    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| KITL [58]               | R-50     | Two   |        |    |    |    |    |    |    |    |    |     |    |    |    | 49.53  | 47.38  | 44.71  | 30.99      |
| R-Det [6]               | R-152    | Retine|        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| S’A-Net + [17]          | R-101    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| R’-50                   | Retine   |        |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| G-Reps [59]             | R-101    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| OrientedR [14]          | R-101    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| RepPoints* [14]         | R-101    | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| R-Det+GWDV [22]         | R-152    | Retine|        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| R-Det+KLD [58]          | R-152    | Retine|        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| R-Det+KIVGD [58]        | R-152    | Retine|        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| TS-Conv                  | D-53     | Retine|        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| TS-Conv*                | D-53     | Retine|        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |
| TS-Conv Lite            | Mobile2  | One   |        |    |    |    |    |    |    |    |    |     |    |    |    | 63.82  | 61.83  | 59.15  | 38.43      |

Note: Bold font indicates the best results. In order to make a fair comparison with the methods in the DOTAV2.0 benchmark [44], the experiments above do not use data augmentation and other tricks like these comparison methods. mAP 0.5@v1.0, mAP 0.5@v1.5, and mAP 0.5@v2.0 denote the results on the DOTAV1.0, DOTAV1.5, and DOTAV2.0 datasets [42], respectively.

objects is significant. Fig. 12 shows that TS-Conv predicts more accurately and with fewer false alarms for OBBs compared to GGHL [7]. As shown in Table X, TS-Conv also outperforms the existing methods on the DIOR-R dataset [45], where objects have more scale variations and categories. In particular, the improvements are more significant for mAP 75 and mAP 50:

The results on the SAR dataset SSDD+ are given in Table XI. The results demonstrate that TS-Conv performs better on the datasets with other data modality. More visualized results are shown in Fig. 13. In summary, extensive comparison experiments are conducted on datasets covering multiple scenes, multimodal images (RGB, infrared, SAR, and panchromatic images), multiple categories of objects, and different lighting conditions (daytime and nighttime). The state-of-the-art results demonstrate the effectiveness and generality of TS-Conv.

Note: Bold font indicates the best results. AP represents anchor-free methods, and AB represents anchor-based methods. The mAP 0.5@0.95 and mAP 0.5@0.50 represent the mAP calculated on standard of VOC07 and VOC12, respectively.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
of each improvement designed in TS-Conv has been confirmed on public datasets demonstrating the following: 1) the effectiveness in performance and robustness. Extensive experiments on several datasets covering multiple scenes, multimodal images, and so on, demonstrating its generality further. Nonetheless, the computational complexity of using DCN is still high, and the impact of OBB annotation error on explicit constraints of localization on TS-Conv is unclear. In the future, it would be interesting to investigate these issues.

The proposed TS-Conv samples the task-wise features from the IFS problem faced by the existing AOOD methods.

TS-Conv on the SSDD dataset. (a) Embedded edge devices, including Nvidia Jetson TX2, Nano, and AGX Xavier, for evaluating the performance of lightweight models. (b) Visualization results of the proposed TS-Conv on the SSD++ dataset.

Table X
Comparative Performance of Different Methods on the DIOR-R Dataset

| Methods       | Anchor | Backbone | mAP_{50} | mAP_{60} | mAP_{75} |
|---------------|--------|----------|----------|----------|----------|
| RetinaNet-O   | AB     | RenNet50 | 57.35    | -        | -        |
| Faster RCNN-O | AB     | RenNet50 | 59.54    | -        | -        |
| Gliding Vertex | AB     | RenNet50 | 60.06    | -        | -        |
| RoI-Transformer | AB    | RenNet50 | 63.87    | -        | -        |
| AOPC         | AB     | RenNet50 | 64.41    | -        | -        |
| Oriented RepPoints [14] | AB     | RenNet50 | 66.71    | -        | -        |
| GGHL          | AF     | DarkNet53 | 66.48    | 36.99    | 37.44    |
| TS-Conv       | AF     | DarkNet53 | 64.47 (±4.10) | 42.69 (±6.79) | 41.38 (±3.42) |

Note: Bold font indicates the best results. AF represents anchor-free methods, and AB represents anchor-based methods.

Table XI
Comparative Performance of Different Methods on the SSD++ Dataset

| Methods       | Anchor | Backbone | mAP_{50} | mAP_{50} | mAP_{75} |
|---------------|--------|----------|----------|----------|----------|
| DRBox-v1 [63] | AB     | VGG16    | 86.41    | 82.40    | -        |
| DRBox-v2 [64] | AB     | VGG16    | 92.81    | 85.17    | -        |
| GGHL          | AF     | DarkNet53 | 95.10    | 90.22    | 22.18    |
| TS-Conv       | AF     | DarkNet53 | 94.34 (±4.50) | 92.44 (±5.38) | 34.16 (±6.99) |
| LO-DeT + GGHL [41] (Baseline52) | AB     | MobileNet2 | 93.87    | 85.90    | 16.64    |
| LO-DeT [41] + DTLX [42] (Baseline52) | AB     | MobileNet2 | 95.89 (±4.65) | 97.86 (±4.55) | 35.75 (±4.28) |

Note: The testing image size is 800 × 800 pixels. To be consistent with the comparison method, the confidence threshold is set to 0.2. AF represents anchor-free methods, and AB represents anchor-based methods.

Fig. 12. Comparison of visualization results between GGHL and TS-Conv on the HRSC2016 dataset. (a-1)–(a-4) Detection results of GGHL. (b-1)–(b-4) Detection results of TS-Conv.

Fig. 13. Presentation of other experiments. (a) Embedded edge devices, including Nvidia Jetson TX2, Nano, and AGX Xavier, for evaluating the performance of lightweight models. (b) Visualization results of the proposed TS-Conv on the SSD++ dataset.

V. CONCLUSION

This article proposes a TS-Conv method to cope with the IFS problem faced by the existing AOOD methods. The proposed TS-Conv samples the task-wise features from their respective sensitive regions and maps them together in alignment to guide a dynamic label assignment for better performance and robustness. Extensive experiments on several public datasets demonstrate the following: 1) the effectiveness of each improvement designed in TS-Conv has been confirmed and the claims made for each component have been verified; 2) TS-Conv has good scalability on lightweight models and for multimodal data and it can improve the performance of the lightweight model without extra inference cost and can be extended to sample modality-wise features with positive results; and 3) the proposed TS-Conv has achieved advanced performance and speed compared to existing AOOD methods on the datasets covering multiple scenes, multimodal images, and so on, demonstrating its generality further. Nonetheless, the computational complexity of using DCN is still high, and the impact of OBB annotation error on explicit constraints of localization on TS-Conv is unclear. In the future, it would be interesting to investigate these issues.

The code is available at https://github.com/Shank2358.

ACKNOWLEDGMENT

The code is available at https://github.com/Shank2358.

REFERENCES

[1] G.-S. Xia et al., “DOTA: A large-scale dataset for object detection in aerial images,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 3974–3983.
[2] 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.
[3] J. Bai et al., “Localizing from classification: Self-directed weakly supervised object localization for remote sensing images,” IEEE Trans. Neural Netw. Learn. Syst., early access, pp. 1–15, 2023, doi: 10.1109/TNNLS.2023.3309889.
[4] 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), Long Beach, CA, USA, Jun. 2019, pp. 2849–2858.
[5] Y. Xu et al., “Gliding vertex on the horizontal bounding box for multi-oriented object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 4, pp. 1452–1459, Apr. 2021.
[6] X. Yang, J. Yan, Z. Feng, and T. He, “R3Det: Refined single-stage detector with feature refinement for rotating object,” 2019, arXiv:1908.05612.
[7] Z. Huang, W. Li, X.-G. Xia, and R. Tao, “A general Gaussian heatmap label assignment for arbitrary-oriented object detection,” IEEE Trans. Image Process., vol. 31, pp. 1895–1910, 2022.
[8] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, “Acquisition of localization confidence for accurate object detection,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 784–799.
[9] G. Song, Y. Liu, and X. Wang, “Revisiting the sibling head in object detector,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11563–11572.
[10] Z. Piao, L. Tang, and B. Zhao, “Unsupervised domain-adaptive object detection via localization regression alignment,” IEEE Trans. Neural Netw. Learn. Syst., early access, pp. 1–12, 2023, doi: 10.1109/TNNLS.2023.3282958.
[11] J. Han, J. Ding, N. Xue, and G.-S. Xia, “ReDet: A rotation-equivariant detector for aerial object detection,” 2021, arXiv:2103.07733.
[12] X. Yang et al., “SCRDet: Towards more robust detection for small, cluttered and rotated objects,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Seoul, South Korea, Oct. 2019, pp. 8232–8241.
[13] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, “RePoints: Point set representation for object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9657–9666.
[14] 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. 1829–1838.
[15] B. Zhu et al., “AutoAssign: Differentiable label assignment for dense object detection,” 2020, arXiv:2007.03496.
[16] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, “Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9759–9768.
[17] J. Han, J. Ding, J. Li, and G.-S. Xia, “Align deep features for oriented object detection,” IEEE Trans. Geosci. Remote Sens., vol. 60, 2021, Art. no. 5602511.
[18] X. Yang and J. Yan, “Arbitrary-oriented object detection with circular smooth label,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 677–694.

[19] J. Yi, P. Wu, B. Liu, Q. Huang, H. Qu, and D. N. Metaxas, “ Oriented object detection in aerial images with box boundary-aware vectors,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., Dec. 2020, pp. 50–50.9.

[20] H. Wei, Y. Zhang, Z. Chang, H. Li, H. Wang, and X. Sun, “ Oriented objects as pairs of middle lines,” ISPRS J. Photogramm. Remote Sens., vol. 169, pp. 268–279, Nov. 2020.

[21] L. Dai, H. Liu, H. Tang, Z. Wu, and P. Song, “AO2-DETR: Arbitrary-oriented object detection transformer,” 2022, arXiv:2205.12785.

[22] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, and Q. Tian, “Rethinking rotated object detection with Gaussian Wasserstein distance loss,” 2021, arXiv:2101.11352.

[23] X. Yang et al., “Learning high-precision bounding box for rotated object detection via Kullback–Leibler divergence,” 2021, arXiv:2106.01883.

[24] Z. Guo, C. Liu, X. Zhang, J. Jiao, X. Ji, and Q. Ye, “Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 86–87.

[25] X. Yang et al., “Dynamic course-to-fine learning for oriented tiny object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2023, pp. 7318–7328.

[26] Q. Ming, L. Miao, Z. Zhou, and Y. Dong, “CFC-Net: A critical feature capturing network for arbitrary-oriented object detection in remote sensing images,” 2021, arXiv:2101.06849.

[27] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, “VarifocalNet: An IoU-aware dense object detector,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 8514–8523.

[28] Y. Wu et al., “Rethinking classification and localization for object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10186–10195.

[29] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021,” 2021, arXiv:2107.08430.

[30] J. Cao, H. Cholakkal, R. M. Anwer, F. S. Khan, Y. Pang, and L. Shao, “D2Det: Towards high quality object detection and instance segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Aug. 2020, pp. 11485–11494.

[31] X. Zhu, H. Hu, S. Lin, and J. Dai, “Deformable ConvNets v2: More deformable, better results,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 9308–9316.

[32] 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. 2965–2974.

[33] J. Dai et al., “Deformable convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 764–773.

[34] X. Pan et al., “Dynamic refinement network for oriented and densely packed object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11207–11216.

[35] R. Liu et al., “An intriguing failing of convolutional neural networks and the coordconv solution,” in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 1–12.

[36] S.-C. Huang, Q.-V. Hoang, and T.-H. Le, “SFA-Net: A selective features absorption network for object detection in rainy weather conditions,” IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 8, pp. 5122–5132, Aug. 2023.

[37] Z. Shao, J. Han, D. Marnerides, and K. Debattista, “Region-proposal-based detection network for oriented target detection in aerial images,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11207–11216.

[38] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, “CondConv: Conditionally parameterized convolutions for efficient inference,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–12.

[39] H. Rezatofighi, N. Tosoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Long Beach, CA, USA, Jun. 2019, pp. 658–666.

[40] 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), Venice, Italy, Oct. 2017, pp. 2999–3007.