remote sensing

MDPI

Article

Vector Decomposition-Based Arbitrary-Oriented Object Detection for Optical Remote Sensing Images

Kexue Zhou [1,†], Min Zhang [1,†], Youqiang Dong [1], Jinlin Tan [1,2], Shaobo Zhao [1] and Hai Wang [1,*]

[1] School of Aerospace Science & Technology, Xidian University, Xi'an 710126, China; kxzhou@stu.xidian.edu.cn (K.Z.); minzhang@xidian.edu.cn (M.Z.); yqdong_2@stu.xidian.edu.cn (Y.D.); linsheng@stu.xidian.edu.cn (J.T.); shaobozhao@stu.xidian.edu.cn (S.Z.)
[2] Shaanxi Academy of Aerospace Technology Application Co., Ltd., Xi'an 710199, China
* Correspondence: wanghai@mail.xidian.edu.cn
† These authors contributed equally to this work.

Abstract: Arbitrarily oriented object detection is one of the most-popular research fields in remote sensing image processing. In this paper, we propose an approach to predict object angles indirectly, thereby avoiding issues related to angular periodicity and boundary discontinuity. Our method involves representing the long edge and angle of an object as a vector, which we then decompose into horizontal and vertical components. By predicting the two components of the vector, we can obtain the angle information of the object indirectly. To facilitate the transformation between angle-based representation and the proposed vector-decomposition-based representation, we introduced two novel techniques: angle-to-vector encode (ATVEncode) and vector-to-angle decode (VTADecode). These techniques not only improve the efficiency of data processing, but also accelerate the training process. Furthermore, we propose an adaptive coarse-to-fine positive–negative-sample-selection (AdaCFPS) method based on the vector-decomposition-based representation of the object. This method utilizes the Kullback–Leibler divergence loss as a matching degree to dynamically select the most-suitable positive samples. Finally, we modified the YOLOX model to transform it into an arbitrarily oriented object detector that aligns with our proposed vector-decomposition-based representation and positive–negative-sample-selection method. We refer to this redesigned model as the vector-decomposition-based object detector (VODet). In our experiments on the HRSC2016, DIOR-R, and DOTA datasets, VODet demonstrated notable advantages, including fewer parameters, faster processing speed, and higher precision. These results highlighted the significant potential of VODet in the context of arbitrarily oriented object detection.

Keywords: vector decomposition; arbitrarily oriented object detection; remote sensing; adaptive sample matching; YOLOX

Citation: Zhou, K.; Zhang, M.; Dong, Y.; Tan, J.; Zhao, S.; Wang, H. Vector Decomposition-Based Arbitrary-Oriented Object Detection for Optical Remote Sensing Images. *Remote Sens.* **2023**, *15*, 4738. https://doi.org/10.3390/rs15194738

Academic Editor: Pedro Melo-Pinto

Received: 3 August 2023
Revised: 13 September 2023
Accepted: 25 September 2023
Published: 27 September 2023

1. Introduction

In the field of remote sensing image object detection, there exists an alternative method for object detection, which involves predicting bounding boxes with angles to detect arbitrarily oriented objects. With the rapid advancements in deep learning, significant progress has been made in the field of arbitrarily oriented object detection in recent years [1–4]. However, direct angle prediction presents challenges related to boundary discontinuity and angular periodicity [1,5–7]. To address these issues, some researchers have explored turning the angle prediction into a vector prediction, which has been shown to be feasible and effective. Nevertheless, there is still room for the improvement of such methods.

In the vector-decomposition-based methods, BBAVectors [8] was the first work to propose the idea of using vector decomposition for arbitrarily oriented object detection. It represents the oriented bounding box by adopting four boundary vectors ($\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{b}, \boldsymbol{l}$) originating from the center of the oriented bounding box to its four sides. However, in addition

to predicting the four boundary vectors ($\boldsymbol{t, r, b, l}$), the method also needs to predict the external width (w_e) and height (h_e) of the oriented bounding box. This requirement results in predicting a total of ten bounding box parameters ($\boldsymbol{t, r, b, l}, w_e, h_e$). In practical scenarios, when an object's bounding box is close to a horizontal orientation, the BBAVectors detector struggles to differentiate between the four boundary vectors, leading to corner cases and problematic results. To address this issue, the bounding boxes of objects are grouped into two categories: horizontal bounding boxes and rotated bounding boxes, which are processed separately. Consequently, BBAVectors needs to predict the types of bounding boxes. This approach introduces complexity and increases the number of bounding box parameters that need to be predicted. Considering the implementation process of BBAVectors, the model becomes complicated due to the excessive number of bounding box parameters, leaving ample room for improvement. ProjBB [9] is an arbitrarily oriented object detector based on vector decomposition, which uses a simplified approach with only six parameters to describe the oriented bounding box. This is in contrast to BBAVectors [8], which employs a more-complex representation with ten bounding box parameters. Specifically, ProjBB uses the following six parameters $(x, y, |u|, |v|, \rho, \alpha)$ to represent an oriented bounding box, where (x, y) are the coordinates of the center point, $(|u|, |v|)$ are the two components of the selected vertex projected onto the diagonal of the oriented bounding box, ρ: is the projection ratio, and α is the quadrant label. However, a challenge arises when dealing with square bounding boxes, as there can be multiple representations: either $(x, y, |u|, 0, 0, \alpha)$ or $(x, y, 0, |v|, 0, \alpha)$. This can lead to uncertainty in square bounding box regression. To address this issue, ProjBB imposes additional constraints. Firstly, the chosen vertex is constrained not to appear on the Y-axis. Secondly, the value of ρ is restricted to the range $[0, 1)$. In an effort to improve detection performance, ProjBB also adopts the strategy of label smoothing. However, there is still considerable room for further improvement based on the detection results. RIE [10] is another arbitrarily oriented object detector that also uses vector decomposition. It employs six parameters $(x, y, \delta_x, \delta_y, b, \psi)$ to describe the oriented bounding box, where (x, y) are the coordinates of the center point of the oriented bounding box, (δ_x, δ_y) are the two components of the long half-axis of the oriented bounding box, b is the short half-axis of the oriented bounding box, and ψ is the orientation label. The method for determining the long half-axis is reported to be too complicated, and the constraints applied to the parameters appear to be incomplete. For a detailed understanding of the constraints, further information is available in the paper [11]. Additionally, RIE introduces the concept of eccentricitywise orientation loss to achieve more-accurate orientation estimation. This loss function likely contributes to enhancing the model's performance when dealing with arbitrarily oriented objects. In [12], Jiang et al. proposed a different approach for representing oriented bounding boxes called long edge decomposition. They utilized ten parameters $(x, y, l, s, w, h, v_x, v_y, o, d)$ to describe the oriented bounding box, where (x, y) are the coordinates of the center point of the oriented bounding box, (l) and (s) are the long edge and short edge of the oriented bounding box, respectively, w and h are the width and height of the circumscribed rectangle of the oriented bounding box, (v_x, v_y) are the two components of the long edge decomposition, o is the bounding box type selection field, and d is the orientation label. Compared to ProjBB and RIE, Jiang et al.'s method employs a larger number of parameters to describe the oriented bounding box. In the context of synthetic aperture radar (SAR) ship detection, Zhou et al. proposed a new vector-decomposition-based arbitrarily oriented object-detection method called AEDet [11]. AEDet employs six parameters $(x, y, |u|, |v|, m, \alpha)$ to describe the oriented bounding box. (x, y) are coordinates of the center point of the oriented bounding box; $(|u|, |v|)$ are the horizontal component and vertical component of the focal vector in the inscribed ellipse of the oriented bounding box; (m) is the difference between the long edge and the focal length; (α) is the orientation label. Compared to other vector-decomposition-based arbitrarily oriented object-detection methods described previously, AEDet stands out for its simplicity in representing oriented bounding boxes and the overall design of the object-detection network. Despite its simplicity, AEDet has demonstrated excellent detection performance. However, AEDet faces a

challenge when dealing with square bounding boxes. The problem of focus disappearing can occur, leading to an object having multiple predicted oriented bounding boxes during the training phase. This issue may limit the performance improvement of AEDet and requires further attention and improvement.

Both from the perspective of vector-decomposition-based arbitrarily oriented object-detection network design and the detection results, there is still room for the improvement of the aforementioned algorithms. After conducting a thorough analysis of these algorithms, we propose a simpler and more-efficient vector-decomposition-based arbitrarily oriented object detector called VODet. To address the focal disappearing problem caused by using square bounding boxes in AEDet [11], we adopted the long edge decomposition approach to describe the oriented bounding boxes. This method retains the angle information, avoiding the issues present in square bounding boxes. Unlike BBAVectors [8], our VODet does not suffer from corner case problems. This is because the representation of oriented bounding boxes in VODet can accurately describe both horizontal and rotated bounding boxes, eliminating the need to group them separately. We maintained the use of the orientation label to determine the orientation of the oriented bounding box, similar to the approaches used in ProjBB [9] and RIE [10]. The specific representation of the oriented bounding box in our method is denoted as $(x, y, l_x, l_y, s, \alpha)$, where (x, y) represents the center point, (l_x, l_y) are the two components of the oriented bounding box's long edge decomposition, s is the short edge of the oriented bounding box, and α is the orientation label. The orientation label α determines whether the oriented bounding box falls in the first-third quadrant or the second-fourth quadrant. Specifically, $\alpha = 0$ indicates that the oriented bounding box is in the second-fourth quadrant, while $\alpha = 1$ indicates that it is in the first-third quadrant. We found that, for simplicity, setting $\alpha = 1$ also signifies that the orientation of the oriented bounding box coincides with the coordinate axis. Our experiments showed that this simplification is effective and does not compromise accuracy. This representation $(x, y, l_x, l_y, s, \alpha)$ is concise and requires fewer parameters compared to BBAVectors [8] and the methods proposed by Jiang et al. [12]. The key contributions of our work are summarized as follows:

1. We propose a novel vector-decomposition-based representation for an oriented bounding box, which requires fewer parameters $(x, y, l_x, l_y, s, \alpha)$. Compared to similar algorithms, the proposed representation method is significantly simpler, making it easier to implement and understand. Moreover, it addresses the issues of corner cases in BBAVectors and the problem of focal disappearing in AEDet, which are common in other existing methods.
2. We propose the angle-to-vector encode (ATVEncode) and vector-to-angle decode (VTADecode) modules to improve the implementation of converting between angle-based representation and the proposed representation. The conversion process of the ATVEncode and VTADecode modules converts all oriented bounding boxes of a batch of images simultaneously into the form of a matrix, eliminating the need for one-by-one processing. This significantly shortens the data-processing time and accelerates the training of the object-detection network
3. We propose an AdaCFPS module to dynamically select the most-suitable positive samples. The AdaCFPS module initially identifies coarse positive samples based on the ground-truth-oriented bounding box. Subsequently, the Kullback–Leibler divergence loss [13] is utilized to assess the matching degree between the ground-truth-oriented bounding box and the coarse-positive-oriented bounding box. Finally, the positive samples that exhibit the highest matching degree are dynamically selected.
4. We developed the anchor-free vector-object-detection (VODet) model based on the proposed representation method and modules. VODet's outstanding performance in object detection was demonstrated through experiments on the HRSC2016 [14], DIOR-R [15], and DOTA [16] datasets, showcasing its effectiveness. Additionally, the experimental results revealed that VODet boasts several advantages, including a fast processing speed, fewer parameters, and high precision. When compared to

similar algorithms, VODet achieved the best results, highlighting the superiority of our vector-decomposition-based arbitrarily oriented object-detection method.

2. Related Work

There are many types of remote-sensing-image-processing tasks, such as change detection [17–19], object detection [20–22], anomaly detection [23–25], etc. The object-detection methods can be divided into bounding-box-based arbitrarily oriented object detection and horizontal-bounding-box-based general object detection. Before the release of the DOTA dataset in 2018 [16], remote sensing object detection using rotating bounding boxes had already been applied in various subdivisions such as vehicle detection [26] and ship detection [14,27]. There are several approaches based on different object-regression techniques in remote-sensing-object-detection methods that employ rotating bounding boxes, which can be categorized as methods based on angle regression, methods utilizing keypoint regression, and other related techniques.

2.1. Angle-Based Methods

Ding et al. [28] proposed the RoI Transformer to address the issue of inconsistency between regional features and objects caused by the use of horizontal candidate frames in the two-stage arbitrarily oriented object-detection-method, which performs spatial transformations on the region of interest (RoI) while being supervised by the labeled rotation bounding box information. This rotation RoI, along with the feature map, is used for geometrically robust feature extraction and ultimately employed for the classification and regression of the rotated RoI. Han et al. [29] proposed a single-stage aligned arbitrarily oriented object-detection network comprising a feature alignment module (FAM) and an oriented detection module (ODM), which can address the significant misalignment issue between anchor boxes and axis-aligned convolutional features, leading to improved performance in arbitrarily oriented object detection tasks. Yang et al. [30] introduced a feature-refined module (FRM) to tackle the challenge of feature misalignment in a single-stage object detector. This module leverages a regression and classification sub-network to generate finely tuned bounding boxes. Additionally, the authors proposed an approximate SkewIoU loss to address the issue of SkewIoU non-guidance. This loss function is designed to enhance object-location-estimation accuracy, thereby contributing to the overall performance improvement of the single-stage object detector. In the work of Yang et al. [5,13], the Gaussian-Wasserstein-distance-based and Kullback–Leibler divergence-based joint unified loss were proposed to address the issue of decoupled regression loss for object angle and size; by leveraging this joint unified loss, the accuracy of arbitrarily oriented object detection can be improved without significant changes to the overall design of the algorithm. Chen et al. [31] introduced a pixel-level IoU metric, called pixel-IoU (PIoU), which can leverage both angle and IoU information to achieve precise regression in arbitrarily oriented object detection. By employing the PIoU loss, the performance of object detection is enhanced, especially in scenarios with high aspect ratios and complex backgrounds. The conventional arbitrarily oriented object-detection methods based on five parameters and eight parameters suffer from issues such as boundary discontinuity and angle periodicity; in order to address these issues, Yang et al. [1,7,32] cleverly transformed angle regression prediction into angle classification, which can effectively address the issues of boundary discontinuity and angle periodicity, leading to significant improvements in the detection accuracy of arbitrarily oriented object detection. Ming et al. [2] made an observation that the positive anchor boxes selected in some arbitrarily oriented object-detection methods may not always maintain accurate detection after regression, while certain negative anchor boxes can achieve precise positioning. To address this issue, they proposed the dynamic-anchor-learning (DAL) method by introducing a novel matching degree that comprehensively evaluates the localization potential of anchor boxes, enabling a more-efficient label-assignment process. Indeed, in the field of remote sensing, several other methods based on angle regression have been extensively researched. The

studies conducted by [15,33–35] explored diverse aspects, including feature extraction, rotation-invariant features, anchor box setting, and more. The efforts from these studies have enhanced the accuracy, efficiency, and overall performance of object detection in remote sensing imagery, making it a thriving area of research with promising prospects for further improvements.

2.2. Keypoint-Based Methods

The angle-regression-based methods might suffer from a lack of spatial information in object prediction, which can limit the performance of object positioning. To address this issue, Fu et al. [3] proposed a keypoint-based arbitrarily oriented object-detection algorithm instead of directly predicting angles, which can generate a rotating bounding box by predicting keypoints that are uniformly distributed in the object area. Sampling keypoints in the object area helps avoid problems associated with direct angle prediction and effectively utilizes the spatial information of the object. Lu et al. [36] also employed uniform keypoints to represent arbitrarily oriented objects and proposed an oriented sensitive keypoint-based rotated detector (OSKDet). In contrast to Fu et al. [3], Lu et al. introduced an oriented sensitive heatmap (OSH) to enhance OSKDet's ability to model arbitrarily oriented objects and implicitly learn their shapes and orientations. The object keypoint representation designed by Fu et al. [3] and Lu et al. [36] requires keypoints to be evenly distributed within the object area. While this method effectively represents regularly shaped objects, it may not be suitable for irregularly shaped objects. Therefore Li et al. [37] presented an effective adaptive keypoint learning method called Oriented RepPoints, which can utilize adaptive keypoint representation for arbitrarily oriented objects, whether they are regular or irregular in shape, and effectively captures the geometric information of these objects. To obtain the orientation bounding box, three orientation transformation functions were proposed based on the arrangement of the adaptive learning keypoints. In order to address the issue of spatial feature aliasing in object-detection methods that rely on horizontal bounding boxes, caused by variations in the object orientation and dense distribution in remote sensing images, Guo et al. [38] proposed a convex-hull feature-adaptation (CFA) method, which aims to achieve optimal feature allocation by constructing a convex hull set and dynamically splitting positive or negative convex hulls. Dai et al. [39] introduced a novel method for arbitrarily oriented object detection based on keypoint detection, called anchor-free corner evolution (ACE), which predicts the offset of each keypoint relative to the four vertices; by minimizing the offset values of the keypoints to the four vertices, the horizontal bounding box gradually evolves into a rotated bounding box. The imaging mechanism of synthetic aperture radar (SAR) images often results in strong scattering of ships, making them distinguishable in SAR images. To address this phenomenon, Sun et al. [40] and Fu et al. [41] devised a strong scattering point detection network for arbitrarily oriented ship object detection in SAR images. The approach involved first detecting the strong scattering points of ships in SAR images and, then, gathering the positions of these scattering points to obtain an arbitrarily oriented bounding box. Keypoint-based object-detection methods have proven to be flexible and hold significant potential in the field of computer vision. Numerous excellent related works such as [42–47] have made significant contributions to the advancement of keypoint-based arbitrarily oriented object detection research. The collective efforts in this area are accelerating the progress of computer vision technologies and bringing keypoint-based methods closer to practical implementations in various real-world scenarios.

2.3. Other Methods

There are diverse approaches for arbitrarily oriented object detection, each offering unique advantages and insights. Wei et al. [48] proposed a method based on predicting paired midlines inside the object, resulting in a single-stage model without anchor frames and non-maximum suppression. The experimental results demonstrated the effectiveness of this approach for arbitrarily oriented object detection. Polar coordinates have been em-

ployed in various works, such as those by He et al. [49], Zhou et al. [50], and Zhao et al. [51], to achieve the detection of arbitrarily oriented objects. Similarly, rotating bounding boxes and minimum circumscribed horizontal bounding boxes were utilized in the methods by Xu et al. [4] and Xie et al. [52] for detecting such objects. Yang et al. [53] introduced weak supervision and self-supervision learning methods to enable object-detection networks to learn angle information from objects initially marked as horizontal boxes. The experimental results demonstrated the effectiveness of this approach, providing new insights into arbitrarily oriented object detection. Considering that a significant portion of existing data is annotated using horizontal boxes, the implementation of methods such as Yang et al.'s approach [53] expanded the available data for arbitrarily oriented object detection. This unification of horizontal bounding box and rotating bounding box datasets in the domain of arbitrarily oriented object detection opens up new possibilities for research and applications in this area. Such a diverse range of methods allows researchers and practitioners to explore various techniques, fostering continued progress and innovation in arbitrarily oriented object detection.

3. The Proposed Method

In this section, we begin by presenting our novel representation of an oriented bounding box and conducted a comparative analysis with similar methods to emphasize the advantages of our approach. Following the introduction of our oriented bounding box representation, we integrated it into an anchor-free object detection framework and provide an overview of our complete object-detection method. Subsequently, based on the overview of our proposed object-detection method, we delve into the detailed explanation of the key modules that constitute our approach. These modules play crucial roles in achieving accurate and efficient arbitrarily oriented object detection. By providing a comprehensive explanation of these key components, we aimed to showcase the efficacy and innovation of our method in handling challenging scenarios of object detection with arbitrary orientations.

3.1. The Representation of Oriented Bounding Box

In vector-decomposition-based arbitrarily oriented object detection, we selected several methods similar to ours for analysis and comparison, which were BBAVectors [8], Longside [12], and Ellipse [11]. Our representation method can be seen as an effective combination of the above methods, but it is simpler and more effective than the original methods. The figures depicting BBAVectors, Longside, Ellipse, and our representation method are shown in Figure 1. In Figure 1a, based on BBAVectors [8], the oriented bounding box is divided into a horizontal bounding box and a rotational bounding box to handle corner cases, making the representation of an oriented bounding box more complex. Moreover, the parameters in BBAVectors are $(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{b}, \boldsymbol{l}, w, h, o)$, where each of $(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{b}, \boldsymbol{l})$ requires two parameters. Without considering the orientation parameter (o), predicting an oriented bounding box requires 10 parameters. In Figure 1b, Longside [12] introduces a novel idea by decomposing the oriented bounding box based on its long edge, but the number of predicted parameters in the Longside method $(x, y, l, s, w, h, v_x, v_y, o, d)$ is at least 10. Similar to BBAVectors, Longside also divides the oriented bounding box into horizontal and rotational components. In Figure 1c, the ellipse representation of an oriented bounding box is an interesting method that has achieved remarkable results in SAR ship detection datasets. The ellipse representation method (x, y, u, v, m, α) has demonstrated its potential in arbitrarily oriented object detection. However, when the oriented bounding box is a square, the focal vector of the ellipse will disappear. In the paper "AEDet" by Zhou et al. [11], the authors showed that the disappearance of the focal vector does not significantly affect the performance of AEDet, as evidenced by their experiments. In this paper, we analyzed the influence of the focal vector's disappearance from a formulaic perspective.

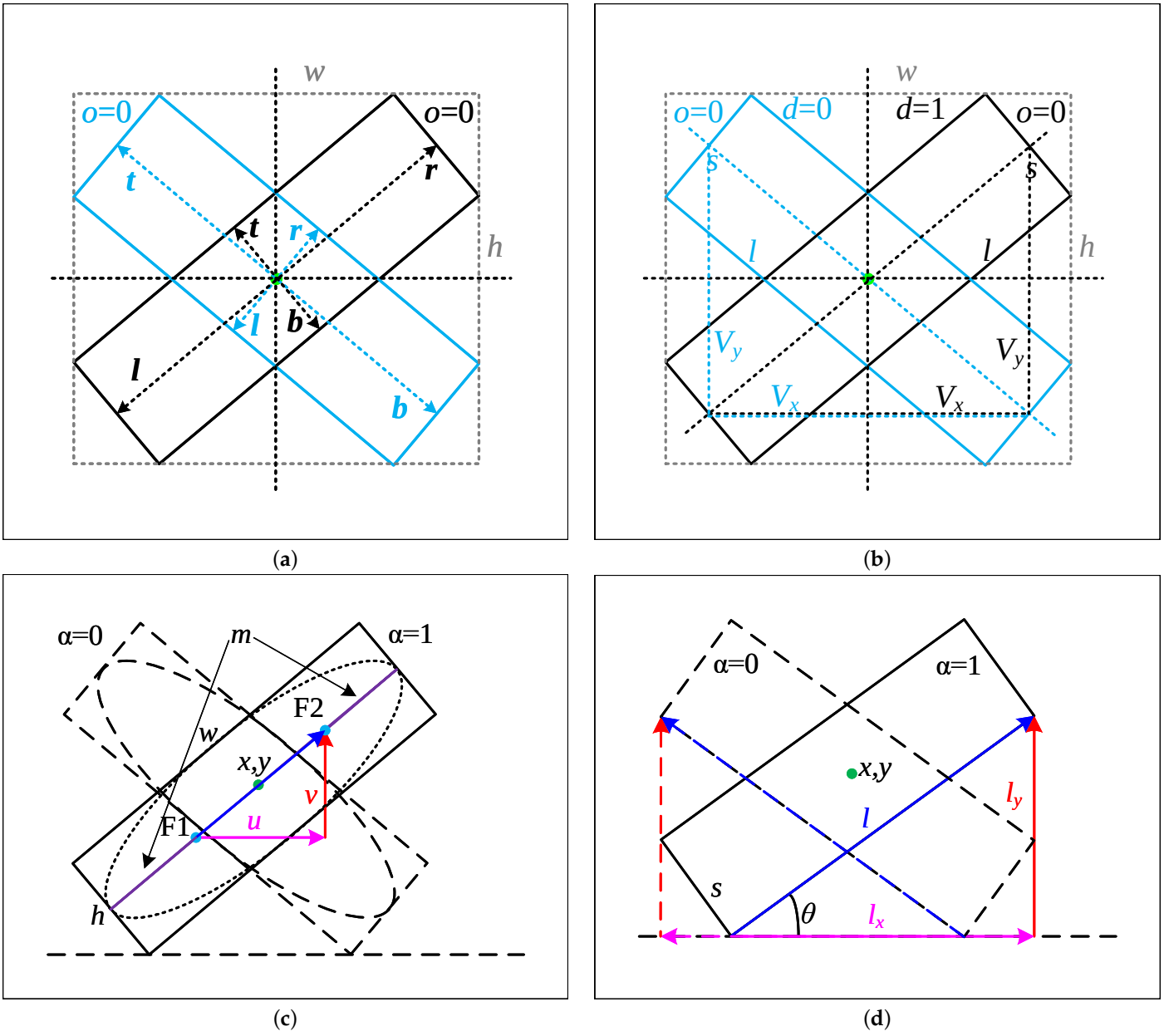


Figure 1. The different representations of the oriented bounding box. (**a**) BBAVectors: $(\boldsymbol{t}, \boldsymbol{r}, \boldsymbol{b}, \boldsymbol{l}, w, h, o)$; (**b**) Longside: $(x, y, l, s, w, h, v_x, v_y, o, d)$; (**c**) Ellipse: (x, y, u, v, m, α); (**d**) ours: $(x, y, l_x, l_y, s, \alpha)$.

When the oriented bounding box is a square, as illustrated in Figure 2, the dotted-line square has a rotation angle (θ) of arbitrary degrees from the solid-line square. The borders of the two squares intersect at the corners, forming eight identical triangles. Let us denote the area of each triangle as A. We can then compute the intersection-over-union (IoU) of the dotted-line square and the solid-line square by using the following formula.

$$IoU = \frac{L^2 - 4A}{L^2 + 4A} \tag{1}$$

where L is the side length of those squares. From this function, we know that the larger the A, the smaller the IoU. When A obtains the maximum value, IoU is the minimum. A is a function of the rotation angle (θ).

$$A = \frac{1}{2} \times w \times h = \frac{L^2}{4} \times \frac{tan(\theta/2)(1 - tan(\theta/2))}{1 + tan(\theta/2)} \tag{2}$$

where w and h are side length of those triangles, as shown in Figure 2. The range of θ is in [0, 90]; thus, the range of $tan(\theta/2)$ is in [0, 1]. Let us denote $tan(\theta/2)$ using x, then the above equation can be rewritten as:

$$A = \frac{L^2}{4} \times \frac{x(1-x)}{1+x} \tag{3}$$

The first-order derivative of A to x is:

$$A' = \frac{L^2}{4} \times \frac{1-2x-x^2}{(1+x)^2} \tag{4}$$

Let $A' = 0$; we can solve this equation to obtain $x = \sqrt{2} - 1$, which means that A reaches the maximum when $x = \sqrt{2} - 1$.

$$A_{max} = \frac{L^2}{4} \times \frac{3\sqrt{2}-4}{\sqrt{2}} \tag{5}$$

Substituting the value $A_{\max}$ into Equation (1), we can obtain the minimum value of IoU, which is equal to $(2-\sqrt{2})/(2\sqrt{2}-2)$ and approximately equal to 0.70711. This value is greater than 0.5, indicating that, when the ground truth is a square, the influence of the orientation is secondary, and the main influencing factor is the center distance between the predicted bounding box and the ground truth. Therefore, the absence of the focal vector does not affect the calculation of the $\mathrm{mAP}_{0.5}$ evaluation criterion.

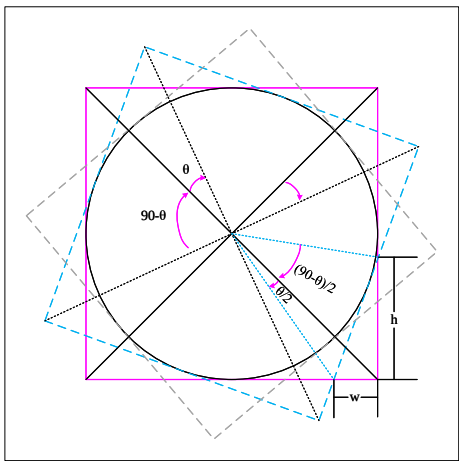


Figure 2. The same object can possess multiple square bounding boxes.

From Equation (1) to Equation (5), we can observe that, when the prediction of the center of the square object is accurate enough, it does not significantly impact the evaluation criteria of the object-detection method. However, it becomes challenging to assess the shape matching degree between the predicted box and the ground truth box due to the absence of angle information in the square-object-prediction process. The removal of the ellipse focal vector is one of the reasons why AEDet's performance improvements are limited on the complex DIOR-R [15] and DOTA [16] datasets. Inspired by the long-edge-decomposition method presented in [12], we redesigned the representation of the oriented bounding boxes, as depicted in Figure 1d. In our novel representation approach, we replaced the focal vector of the ellipse with a long edge vector. Subsequently, we predicted the two components of the long edge vector decomposition. This representation method can be considered as an enhancement over the ellipse representation utilized in AEDet [11]. In our specific representation, we predicted the following parameters $(x, y, l_x, l_y, s, \alpha)$ to determine a predicted oriented bounding box, where (x, y) represent the center of the oriented bounding box, (l_x, l_y) denote the two components of the long edge vector in the oriented bounding box, (s) refers to the short edge of the oriented bounding box, and (α) serves as an orientation

indicator. When $\alpha = 1$, this means that the orientation of the object belongs to the first and third quadrants; when $\alpha = 0$, this means that the orientation of the object belongs to the second and fourth quadrants. Additionally, for simplicity, when $\alpha = 1$, this also indicates that the orientation of the object coincides with the coordinate axis. As depicted in Figure 1d, this representation method successfully overcomes the problem of disappearing ellipse focal vectors. Moreover, owing to the advantages of the long-edge-decomposition method, the angle information of the square bounding box is preserved effectively. This improvement enhances the representation capabilities and performance of our model. Indeed, our predicted parameters $(x, y, l_x, l_y, s, \alpha)$ offer a simpler representation compared to the parameters used in BBAVectors [8] and Longside [12], namely $(\boldsymbol{t, r, b, l}, w, h, o)$ and $(x, y, l, s, w, h, v_x, v_y, o, d)$, respectively. Unlike BBAVectors [8] and Longside [12], we did not divide the oriented bounding box into horizontal and rotational bounding boxes, which further simplified our representation method. The most-crucial advantage of our method lies in the combination of representation techniques from Ellipse [11] and Longside [12]. This enabled our method to retain angle information in the prediction of square bounding boxes, thereby leading to improved object-detection performance.

3.2. The Overall Structure of Proposed Method

After introducing the vector representation of an oriented bounding box, we constructed VODet by incorporating it into the YOLOX [54] object-detection framework. The overview of the proposed VODet is illustrated in Figure 3. Additionally, we display the positions of the ATVEncode, VTADecode, and AdaCFPS modules, which will be introduced in the following subsections.

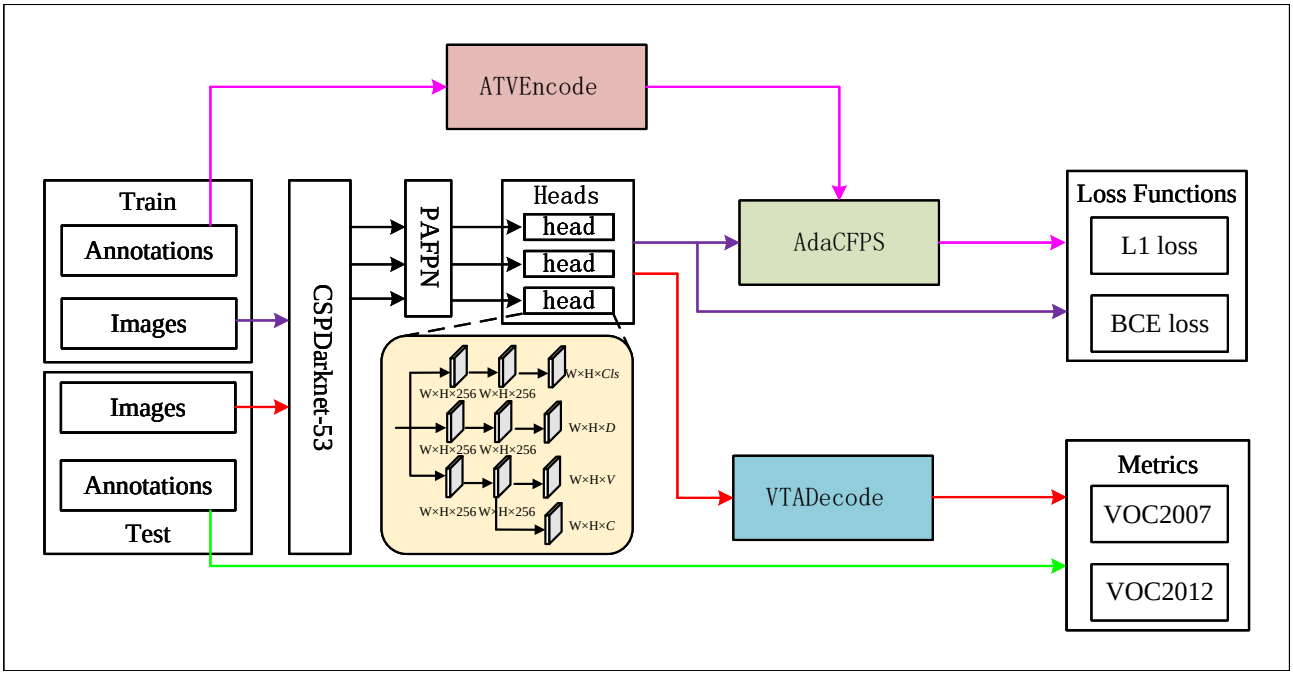


Figure 3. The structure of VODet.

In the structure of VODet, the vector representation of an oriented bounding box is utilized in each head. Each head is derived from the the PAFPN [55] network and consists of three branches. The first branch is responsible for class prediction (*Cls* represents the number of object categories). The second branch predicts the orientations of the oriented bounding box by treating orientation prediction as a classification task. Thus, the value D in the head can predict either α_1 or α_2. If the predicted α_1 is greater than or equal to α_2, then the orientation of the predicted bounding box belongs to the second and fourth quadrants. Otherwise, if the predicted α_1 is less than α_2, then the orientation of the predicted bounding

box belongs to the first and third quadrants. The third branch is utilized to predict the oriented bounding box information (x, y, l_x, l_y, s). Therefore, the dimensionality V of this branch is set to five. Additionally, on this branch, another branch is introduced to predict the confidence (C is 1) of the object detection.

We can observe the positions of three modules, namely ATVEncode, VTADecode, and AdaCFPS, in Figure 3. During the training phase, ATVEncode is used to convert the ground truth bounding box coordinates (x, y, w, h, θ) into $(x, y, l_x, l_y, s, \alpha)$. The AdaCFPS module dynamically selects positive samples based on the ground truth bounding boxes transformed via the ATVEncode module and the predicted bounding boxes. In the testing phase, to visualize and evaluate the predicted results, we employed the VTADecode module to convert the predicted oriented bounding box coordinates $(x, y, l_x, l_y, s, \alpha_1, \alpha_2)$ into (x, y, l, s, θ). VODet employs specific loss functions, including the L1 loss for bounding box regression (x, y, l_x, l_y, s) and the BCE loss for the predictions of categories and orientations. The evaluation metrics used are the mAPs of VOC2007 or VOC2012.

3.3. The Proposed ATVEncode Module

The paper introduces a novel representation $(x, y, l_x, l_y, s, \alpha)$ for oriented bounding boxes in the context of arbitrarily oriented object detection. Unlike the traditional format (x, y, w, h, θ), this new representation hides the angle (θ) in the decomposition of the long-side vector (l_x, l_y) instead of directly using the angle information. During the training phase, a conversion from the traditional format (x, y, w, h, θ) to the proposed format $(x, y, l_x, l_y, s, \alpha)$ is necessary for proper loss calculation. However, this transformation process involves determining the long side and the corresponding orientation (α), which can be error-prone and crucial for accurate results. To address this, the paper introduces the angle-to-vector encode (ATVEncode) algorithm, presented in Pytorch-style code in Algorithm 1. The ATVEncode algorithm demonstrates that all transformation processes involve matrix operations, enabling parallel computation. As a result, this significantly reduces the transformation time, leading to faster training times.

Algorithm 1 Angle-to-vector encode (Pytorch-style)

Input: $\mathcal{I}$, which is an $n \times p$ matrix; n means the number of ground truth bounding boxes; $p = 5$ means the representation vector with the angle, i.e., (x, y, w, h, θ)

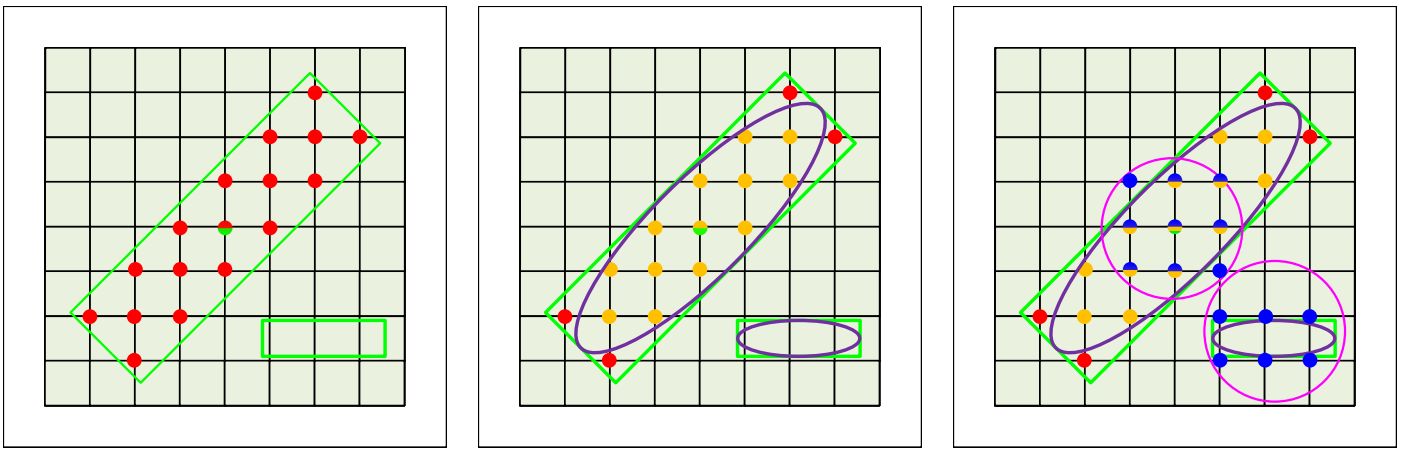
Output: $\mathcal{O}$, which is an $n \times q$ matrix; n means the number of bounding boxes; $q = 6$ means the representation vector with vector decomposition, i.e., $(x, y, l_x, l_y, s, \alpha)$

Clone the input $\mathcal{I}$ as $\mathcal{T}$
$\mathcal{T} = \mathcal{I}.clone()$
Find the long edge and short edge; put the long edge in the position of w; put the short edge in the position of h
$mask = \mathcal{T}[:, 2:3] > \mathcal{T}[:, 3:4]$
$\mathcal{T}[:, 2:3] = \mathcal{I}[:, 2:3] * mask + \mathcal{I}[:, 3:4] * (\sim mask)$
$\mathcal{T}[:, 3:4] = \mathcal{I}[:, 2:3] * (\sim mask) + \mathcal{I}[:, 3:4] * mask$
Adjust the angle according to the long edge and short edge
$\mathcal{T}[:, 4:5] = \mathcal{I}[:, 4:5] * mask + (\pi/2 + \mathcal{I}[:, 4:5] * (\sim mask))$
Generate the horizontal $(\mathcal{H})$ and vertical $(\mathcal{V})$ components of the long edge
$\mathcal{H} = \mathcal{T}[:, 2:3] * torch.cos(\mathcal{T}[:, 4:5])$
$\mathcal{V} = \mathcal{T}[:, 2:3] * torch.sin(\mathcal{T}[:, 4:5])$
Generate the orientation sign (α) of the oriented bounding box
$\alpha = \mathcal{H}[:, 0:1] * \mathcal{V}[:, 0:1] > 0$
Concatenate $x, y, l_x, l_y, s, \alpha$ in the second dimension as the final output
$x = \mathcal{T}[:, 0:1]$
$y = \mathcal{T}[:, 1:2]$
$l_x = torch.abs(\mathcal{H})$
$l_y = torch.abs(\mathcal{V})$
$s = \mathcal{T}[:, 3:4]$
$\mathcal{O} = torch.cat((x, y, l_x, l_y, s, \alpha), dim = -1)$

3.4. The Proposed AdaCFPS Module

The proposed VODet builds upon the YOLOX framework, which was originally designed for horizontal object detection, not oriented object detection. As a result, the positive-sample-selection method used in YOLOX cannot be directly applied to VODet for oriented object detection. Therefore, the paper introduces a new module called adaptive coarse-to-fine positive sample selection (AdaCFPS) to address this issue and enable the selection of positive samples for VODet.

In the AdaCFPS module, the primary objective is to select coarse positive samples for oriented object detection. The process involves two main steps: first, choosing points within the inscribed ellipse regions of the oriented ground truth bounding boxes and, second, selecting points within a circle region of each oriented ground truth bounding box. The decision to choose the points in the circle region as positive samples in the AdaCFPS module is motivated by the need to address the challenge of small object detection during the training process. Figure 4c illustrates the importance of including the circle region as part of the positive-sample-selection process. When the circle region lies entirely inside the oriented ground truth bounding box, the points within the inscribed ellipse region are considered positive samples (as shown in Figure 4c). In this situation, the existence of the circle region has little impact because there are already positive samples available within the oriented ground truth bounding box. However, the critical role of the circle region becomes evident when an object is small and there are no points inside the oriented ground truth bounding box. This situation would result in a lack of any positive samples for this specific oriented ground truth bounding box, which is unacceptable for small object detection during training. In such cases, the circle region ensures that the oriented ground truth bounding box has matching coarse positive samples as much as possible, as depicted in Figure 4c. By including the circle region, the AdaCFPS module improves the positive-sample-matching rate for small objects, addressing the challenge of detecting and accurately predicting small objects in the training dataset.



(**a**) (**b**) (**c**)

Figure 4. The steps of coarse-to-fine positive sample selection, where the green rectangle box means ground truth box, the ellipse and circle mean the selected coarse positive samples regions, the points mean the positive samples. (**a**) The original positive samples; (**b**) The selected coarse positive samples in ellipse region; (**c**) The coars positive samples in ellipse and circle regions.

After obtaining the final coarse positive samples, we employed the Kullback–Leibler divergence (KLD) loss [13] to assess the matching degree between the coarse positive samples and oriented ground truth bounding boxes. Subsequently, we selected the top-k samples and calculated their costs [54], taking into account both the classification loss and the KLD loss. Finally, we generated dynamic k samples based on the combined score of the matching degree and corresponding cost. In cases where multiple oriented ground truth bounding boxes have the same positive sample, we assigned the positive sample to the

oriented ground truth bounding box with the lowest cost. To facilitate a clear understanding of the positive-sample-selection process, we provide the detailed steps in Algorithm 2.

Algorithm 2 Adaptive coarse-to-fine positive sample selection (Pytorch style)

Input: $\mathcal{I}_g$, which is an $n \times t$ matrix; n means the number of ground truth bounding boxes; $t = 5$ means the long edge representation vector with the angle, i.e., (x, y, l, s, θ);
$\mathcal{M}$, which is a $w \times h$ feature map with stride $\mathcal{S}$; (x_m, y_m) are the coordinates of $\mathcal{M}$. For simplicity, $\mathcal{M}$ is reshaped to $n \times w * h \times 2$, where n means the number of ground truth bounding boxes; 2 means x_m andy_m;
$\mathcal{I}_p$, which is an $m \times r$ matrix; $m = w * h$ means the number of predicted bounding boxes; $r = 7$ means the predicted vector with vector decomposition, i.e., $(x, y, l_x, l_y, s, \alpha_1, \alpha_2)$;
$\mathcal{C}_g$, which is an $n \times 1$ matrix; n means the number of ground truth bounding boxes; "1" indicates the category to which the ground truth bounding box belongs;
$\mathcal{C}_p$, which is an $m \times c$ matrix; $m = w * h$ means the number of predicted bounding boxes; c indicates the number of object categories.

Output: $\mathcal{P}$, which is a $d \times r$ matrix; d means the number of predicted fine positive bounding boxes; $r = 7$ means the representation vector with the angle, i.e., $(x, y, l_x, l_y, s, \alpha_1, \alpha_2)$

First step: select the inscribed ellipse regions of ground truth bounding boxes as positive samples
$x_e = \mathcal{M}[:, :, 0] * \mathcal{S} - \mathcal{I}_g[:, 0]$
$y_e = \mathcal{M}[:, :, 1] * \mathcal{S} - \mathcal{I}_g[:, 1]$
$A = (torch.cos(\mathcal{I}_g[:, 4])/(\mathcal{I}_g[:, 2]/2))^2 + (torch.sin(\mathcal{I}_g[:, 4])/(\mathcal{I}_g[:, 3]/2))^2$
$B = 2 * torch.cos(\mathcal{I}_g[:, 4]) * torch.sin(\mathcal{I}_g[:, 4]) * (1/(\mathcal{I}_g[:, 2]/2)^2 - 1/(\mathcal{I}_g[:, 3]/2)^2)$
$C = (torch.sin(\mathcal{I}_g[:, 4])/(\mathcal{I}_g[:, 2]/2))^2 + (torch.cos(\mathcal{I}_g[:, 4])/(\mathcal{I}_g[:, 3]/2))^2$
$inbox = A * x_e^2 + B * x_e * y_e + C * y_e^2 \leqslant 1$
$inbox\_mask = inbox.sum(dim = 0) > 0$
Second step: select samples within a fixed radius as positive samples, where the radius is set to $\mathcal{S}$ * 1.5
$incenter = x_e^2 + y_e^2 \leqslant (\mathcal{S} * 1.5)^2$
$incenter\_mask = incenter.sum(dim = 0) > 0$
Third step: select samples within inscribed ellipse regions or circles as the final coarse positive samples ($\mathcal{T}_p$)
$inboxanchor = inbox\_mask | incenter\_mask$
$\mathcal{T}_p = \mathcal{I}_p[inboxanchor]$
$\mathcal{C}_p = \mathcal{C}_p[inboxanchor]$
Fourth step: calculate the matching degree according to the Kullback–Leibler divergence (KLD) loss (refer to [13])
$md = 1 - KLDLoss(\mathcal{I}_g, \mathcal{T}_p)$
Fifth step: obtain the fine positive samples according the SimOTA algorithm (refer to [54])
$cost = BCELoss(\mathcal{C}_p, \mathcal{C}_g) + 3 * KLDLoss(\mathcal{I}_g, \mathcal{T}_p)$
$finemask = SimOTA(cost, md, inboxanchor)$
$\mathcal{P} = \mathcal{I}_p[finemask]$

3.5. The Proposed VTADecode Module

In the testing or inference phase, it is necessary to visualize or evaluate the results. However, the predicted format $(x, y, l_x, l_y, s, \alpha_1, \alpha_2)$ is not convenient for this purpose. Therefore, we need to convert this format into another format that includes angle information, i.e., vector-to-angle transformation. The transformation process involves several steps to ensure the accurate orientations of the predicted bounding boxes based on the predicted orientation parameters (α_1, α_2). Additionally, we need to calculate the angle (θ) and long-side length (l) using the two components (l_x, l_y) of the long-side vector. It is crucial to correct the angle according to the orientation, as an incorrect angle could be obtained otherwise.

To achieve the desired format (x, y, l, s, θ), we present the detailed transformation steps in Algorithm 3.

Algorithm 3 Vector-to-angle decode (Pytorch style)

Input: $\mathcal{I}$, which is an $n \times p$ matrix; n means the number of predicted bounding boxes; $p = 7$ means the predicted vector with vector decomposition, i.e., $(x, y, l_x, l_y, s, \alpha_1, \alpha_2)$
Output: $\mathcal{O}$, which is an $n \times q$ matrix; n means the number of predicted bounding boxes; $q = 5$ means the representation vector with the angle, i.e., (x, y, l, s, θ)

Calculate the lengths of the long edge and short edge
$l = torch.sqrt(\mathcal{I}[:, 2:3]^2 + \mathcal{I}[:, 3:4]^2)$
$s = \mathcal{I}[:, 4:5]$
Obtain the orientation of the predicted bounding boxes
$\mathcal{I}[:, 5] = torch.max(\mathcal{I}[:, 5:7], dim = -1)[1]$
$mask = \mathcal{I}[:, 5:6] < 1$
Calculate the angle of the predicted bounding boxes
$\theta = torch.acos(\mathcal{I}[:, 2:3]/l)$
Correct the angle according to the orientation of the predicted bounding boxes
$\theta = \theta * (\sim mask) + (-\theta) * mask$
Concatenate x, y, l, s, θ in the second dimension as the final output
$x = \mathcal{I}[:, 0:1]$
$y = \mathcal{I}[:, 1:2]$
$\mathcal{O} = torch.cat((x, y, l, s, \theta), dim = -1)$

4. Experiments and Analysis

In this section, we conducted experiments to validate the effectiveness of the proposed VODet on three datasets: HRSC2016, DOTA, and DIOR-R. The experiments were conducted on a PC with the following specifications: Intel® Core™i7-6850K CPU @ 3.60 GHz × 12, 64 GB of memory, and two NVIDIA TITAN Xp GPUs with 12 GB of memory each. The operating system used was 64 bit Ubuntu 18.04.6 LTS. Firstly, we describe the settings used in our object-detection method, followed by the introduction of the three datasets used in our experiments. We conducted the experiments on these three datasets and analyzed the experimental results subsequently. Finally, we performed related ablation experiments and provide the model size and inference time of the proposed method.

4.1. Experiment Settings

In the proposed VODet method, we used different input sizes for each dataset: 800 × 800 for HRSC2016 and DIOR-R and 1024 × 1024 for DOTA. During the training phase, we set the learning rate per image to 0.00015625. The optimizer used was SGD with a weight decay of 5×10^{-4} and a momentum of 0.9. To increase the diversity of the training samples, we employed several data-augmentation techniques, including Mosaic, Mixup, HSV Transformation, Horizontal Flip, and Vertical Flip. The probabilities of using Mosaic, Mixup, and HSV Transformation were set to 1.0, while the probabilities of applying Horizontal Flip and Vertical Flip were 0.5. We conducted the training for a total of 36 epochs, and within the last 15 epochs, the data-augmentation methods were prohibited to ensure better convergence. To speed up the convergence of VODet, we initialized the model using a pre-trained model on the COCO dataset. During the test phase, we set the confidence threshold to 0.01 and the non-maximum suppression (NMS) threshold to 0.5 to post-process the detection results.

4.2. Experiment Datasets

4.2.1. HRSC2016

The HRSC2016 dataset is an arbitrarily oriented ship-detection dataset comprising 1061 images with sizes ranging from 300 × 300 to 1500 × 900. The dataset is divided into

three subsets: a training set (436 images), a validation set (181 images), and a testing set (444 images). During the training phase, both the training set and validation set were utilized for training the ship-detection model.

4.2.2. DOTA

The DOTA dataset contains a total of 2806 images and includes 15 object categories, namely: plane, baseball diamond, bridge, ground track field, small vehicle, large vehicle, ship, tennis court, basketball court, storage tank, soccer ball field, roundabout, harbor, swimming pool, helicopter. Since the original images in the DOTA dataset have large sizes ranging from 800 × 800 to 4000 × 4000, they are too large for direct training and testing. Therefore, the images were cropped into smaller patches of size 1024 × 1024 with a stride of 512. This cropping process allowed for better handling of the large images during training and testing. For the multi-scale training and testing, the original images were first resized at three scales: 0.5, 1.0, and 1.5. After resizing, the images were then cropped into patches of 1024 × 1024 with a stride of 512 at each scale. During the training phase, both the training dataset and validation dataset were used to train the model. After training, the model's performance was evaluated using the testing dataset. The experimental results were obtained through the DOTA evaluation server, which provides a standardized way to evaluate the performance of object-detection algorithms on the DOTA dataset.

4.2.3. DIOR-R

DIOR-R is an extended version of the DIOR dataset, which has been relabeled with oriented annotations. It consists of a total of 23,463 images and contains 192,472 instances across 20 object categories. The 20 object categories present in the DIOR-R dataset are: airplane, ground track field, overpass, airport, golf field, baseball field, basketball court, bridge, train station, chimney, dam, expressway service area, expressway toll station, harbor, ship, stadium, storage tank, tennis court, vehicle, windmill. The training and validation sets combined contain 11,725 images with a total of 68,073 instances. The testing set, on the other hand, includes 11,738 images with a total of 124,445 instances.

4.3. Experiments on HRSC2016

We conducted experiments on the HRSC2016 dataset, and the results are presented in Table 1. The evaluation was based on the mean average precision (mAP) metrics for both the VOC2007 and VOC2012 datasets. The results clearly demonstrated the remarkable detection performance of the proposed VODet method. It achieved an mAP of 90.23% on VOC2007 and 96.25% on VOC2012, surpassing the performance of most algorithms participating in the comparison. In terms of the VOC2007 mAP, VODet performed only 0.22 lower than AEDet and 0.17 lower than Oriented R-CNN. For the VOC2012 mAP, VODet scored 0.65 lower than AEDet and 0.25 lower than Oriented R-CNN. These small differences highlight that VODet achieved competitive results when compared with other excellent algorithms. Overall, the impressive performance of VODet in object detection on remote sensing images underscores its potential and effectiveness in handling challenging detection tasks.

In order to visually demonstrate the detection capabilities of VODet, we present several images with predicted bounding boxes in Figure 5, where the yellow ellipse means false detection or missed detection or inaccurate orientation prediction. Through the visualization of VODet's detection results, it became evident that the model can effectively handle objects with high aspect ratios. However, when the object was too similar to the surrounding background, as shown in the second row and second column image in Figure 5, the orientation of the object may be inaccurate due to interference from the background information. Additionally, missed detections will occur, one of the reasons being that similar objects do not appear in the training dataset, another reason being that the objects are too small to detect. In the last two images of Figure 5, the missed detections occurred due to the two few similar training samples according to the analysis of HRSC2016.

Anyway, the effectiveness of VODet was demonstrated by the successful visualization and the results in Table 1.

Table 1. The results of different methods on the HRSC2016 dataset.

Methods	Backbone	Size	mAP (VOC2007)	mAP (VOC2012)
Two-stage method				
Rotated RPN [56]	ResNet-101	800 × 800	79.08	85.64
R2CNN [57]	ResNet-101	800 × 800	73.07	79.73
RoI Transformer [28]	ResNet-101	512 × 800	86.20	-
Gliding Vertex [4]	ResNet-101	512 × 800	88.20	-
CenterMap-Net [58]	ResNet-50	-	-	92.80
Oriented R-CNN [52]	ResNet-50	-	90.40	96.50
FPN-CSL [1]	ResNet-101	-	89.62	96.10
one-stage method				
R3Det [30]	ResNet-101	800 × 800	89.26	96.01
DAL [2]	ResNet-101	800 × 800	89.77	-
S^2ANet [29]	ResNet-101	512 × 800	90.17	95.01
RRD [59]	VGG16	384 × 384	84.30	-
RetinaNet-O [60]	ResNet-101	800 × 800	89.18	95.21
PIoU [31]	DLA-34	512 × 512	89.20	-
DRN [61]	Hourglass-34	768 × 768	-	92.70
CenterNet-O [62]	DLA-34	-	87.89	-
AEDet [11]	CSPDarknet-53	800 × 800	90.45	96.90
ours				
VODet	CSPDarknet-53	800 × 800	90.23	96.25

Figure 5. The visualization of the detection results on the HRSC2016 dataset, the yellow ellipse means false detection or missed detection or inaccurate orientation prediction.

4.4. Experiments on DOTA

To further validate the effectiveness of the proposed VODet method, we conducted additional experiments on the widely used DOTAv1.0 dataset. The experimental results under different data-processing scenarios are summarized in Table 2. In the case where both the training and testing data were generated using a single-scale (1.0) cropping, VODet achieved a promising result of 75.7%. This already surpassed the performance of most other arbitrarily oriented algorithms participating in the comparison, notably outperforming AEDet with single-scale cropping, which achieved a result of 72.8%. To explore the benefits

of multiple-scale cropping, we conducted experiments with VODet using training and testing data generated at scales of 0.5, 1.0, and 1.5 (denoted as VODet‡). The results showed that VODet‡ obtained even better detection results compared to VODet's single-scale cropping, achieving an outstanding result of 77.8%. This outperformed the VODet's performance by 2.1%, and it stood as the best result among the algorithms participating in the comparison. The remarkable performance of both VODet and VODet‡ demonstrated the superiority of the proposed arbitrarily oriented object-detection method, further validating its effectiveness and potential.

Table 2. The experimental results of different methods on the DOTAv1.0 dataset. ‡ denotes the training and testing of multiple-scale cropping.

Methods	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
two-stage method																	
CenterMap-Net [58]	ResNet-50	89.0	80.6	49.4	62.0	78.0	74.2	83.7	89.4	78.0	83.5	47.6	65.9	63.7	67.1	61.6	71.6
SCRDet [63]	ResNet-101	90.0	80.7	52.1	68.4	68.4	60.3	72.4	90.9	87.9	86.9	65.0	66.7	66.3	68.2	65.2	72.6
RoI Transformer [28]	ResNet-101	88.6	78.5	43.4	75.9	68.8	73.7	83.6	90.7	77.3	81.5	58.4	53.5	62.8	58.9	47.7	69.6
FPN-CSL [1]	ResNet-152	90.3	85.5	54.6	75.3	70.4	73.5	77.6	90.8	86.2	86.7	69.6	68.0	73.8	71.1	68.9	76.2
Gliding Vertex [4]	ResNet-101	89.6	85.0	52.3	77.3	73.0	73.1	86.8	90.7	79.0	86.8	59.6	70.9	72.9	70.9	57.3	75.0
FR-Est [3]	ResNet-101	89.6	81.2	50.4	70.2	73.5	78.0	86.4	90.8	84.1	83.6	60.6	66.6	70.6	66.7	60.6	74.2
DODet [64]	ResNet-50	89.3	84.3	51.4	71.0	79.0	82.9	88.2	90.9	86.9	84.9	62.7	67.6	75.5	72.2	45.5	75.5
Oriented R-CNN [52]	ResNet-50	89.5	82.1	54.8	70.9	78.9	83.0	88.2	90.9	87.5	84.7	64.0	67.7	74.9	68.8	52.3	75.9
AOPG [15]	ResNet-50	89.3	83.5	52.5	70.0	73.5	82.3	88.0	90.9	87.6	84.7	60.0	66.1	74.2	68.3	57.8	75.2
one-stage method																	
RetinaNet-O [60]	ResNet-50	88.7	77.6	41.8	58.2	74.6	71.6	79.1	90.3	82.2	74.3	54.8	60.6	62.6	69.6	60.6	68.4
S^2ANet [29]	ResNet-50	89.1	82.8	48.4	71.1	78.1	78.4	87.3	90.8	84.9	85.6	60.4	62.6	65.3	69.1	57.9	74.1
R3Det [30]	ResNet-101	88.8	83.1	50.9	67.3	76.2	80.4	86.7	90.8	84.7	83.2	62.0	61.4	66.9	70.6	53.9	73.8
DAL [2]	ResNet-50	88.7	76.6	45.1	66.8	67.0	76.8	79.7	90.8	79.5	78.5	57.7	62.3	69.1	73.1	60.1	71.4
AEDet [11]	CSPDarknet-53	87.5	77.6	51.7	68.2	78.0	80.5	86.5	90.3	80.7	75.4	54.6	59.6	73.4	73.8	53.8	72.8
DRN [61]	Hourglass-104	88.9	80.2	43.5	63.4	73.5	70.7	84.9	90.1	83.9	84.1	50.1	58.4	67.6	68.6	52.5	70.7
RSDet [6]	ResNet-101	89.8	82.9	48.6	65.2	69.5	70.1	70.2	90.5	85.6	83.4	62.5	63.9	65.6	67.2	68.0	72.2
ours																	
VODet	CSPDarknet-53	86.3	80.0	52.4	67.9	79.3	83.9	87.9	90.8	87.6	85.6	63.3	61.2	75.8	78.9	54.5	75.7
VODet ‡	CSPDarknet-53	88.8	83.6	53.2	78.7	79.9	84.1	88.5	90.8	88.1	86.2	64.4	67.7	76.9	79.7	57.8	77.8

PL: plane, BD: baseball diamond, BR: bridge, GTF: ground track field, SV: small vehicle, LV: large vehicle, SH: ship, TC: tennis court, BC: basketball court, ST: storage tank, SBF: soccer ball field, RA: roundabout, HA: harbor, SP: swimming pool, HC: helicopter.

We also present the visualization of the VODet detection results in Figure 6, where the yellow ellipse means missed detection or false detection. It was evident that VODet performed exceptionally well with small and dense oriented objects, such as swimming pools and small vehicles. Moreover, VODet accurately detected densely distributed oriented objects, such as large vehicles and ships. The impressive detection visualization in harbors further confirmed VODet's capability to handle high aspect ratios effectively. However, VODet also made mistakes sometimes, as shown in the last two images of Figure 6: the shadowed small vehicles were not detected, and the false harbor was detected. These situations occurred because the proposed VODet does not provide an extra strategy to distinguish the object from the background, which led to the occurrence of the missed detection and false detection. Some effective strategies [65,66] can be applied to VODet in future research for better detection performance.

Figure 6. The visualization of the detection results on the DOTA dataset, the yellow ellipse means false detection or missed detection or inaccurate orientation prediction.

4.5. Experiments on DIOR-R

DIOR-R is another widely used arbitrarily oriented object-detection dataset. In our experiments, we compared VODet with several other algorithms, including Faster R-CNN-O [67], RetinaNet-O [60], Gliding Vertex [4], RoI Transformer [28], AOPG [15], DODet [64], QPDet [68], and AEDet [11]. The experimental results are shown in Table 3. Among all the participating algorithms, VODet achieved the best performance on DIOR-R, obtaining an impressive result of 67.66%, which is 1.52% higher than AEDet. These results demonstrated that VODet maintained remarkable detection performance even on more-complex datasets, highlighting its superiority once again. Furthermore, on both the DOTA and DIOR-R datasets, VODet outperformed AEDet, indicating that the object representation of VODet was more efficient than that of AEDet, even when using the same CSPDarknet-53 backbone.

We present the visualization of VODet's detection results in Figure 7, which further validated its ability to detect small, densely arranged, and high-aspect-ratio objects. As depicted in Figure 7, where the yellow ellipse means missed detection or false detection, some failed detections are shown in the last row of Figure 7. In theory, the vehicles should be on the road or in the parking lot, and the ship should be in the harbor or at sea. However, VODet could not establish the connection between the object and the background, which led to the false detections in the first three images in the last row of Figure 7. The false detection in the last image in the last row of Figure 7 was because some objects of different categories had similar characteristics. Overall, although VODet achieved remarkable detection performance, the ability to distinguish object from the background needs to be further improved.

Table 3. The experimental results of different methods on the DIOR-R dataset.

Methods	Faster R-CNN-O [67]	RetinaNet-O [60]	Gliding Vertex [4]	RoI Transformer [28]	AOPG [15]	DODet [64]	QPDet [68]	AEDet [11]	VODet
Backbone	ResNet-50	ResNet-50	ResNet-50	ResNet-50	ResNet-50	ResNet-50	ResNet-50	CSPDarknet-53	CSPDarknet-53
APL	62.79	61.49	65.35	63.34	62.39	63.40	63.22	81.06	85.74
APO	26.80	28.52	28.87	37.88	37.79	43.35	41.39	48.09	53.15
BF	71.72	73.57	74.96	71.78	71.62	72.11	71.97	77.35	76.25
BC	80.91	81.17	81.33	87.53	87.63	81.32	88.55	89.66	89.41
BR	34.20	23.98	33.88	40.68	40.90	43.12	41.23	43.46	35.49
CH	72.57	72.54	74.31	72.60	72.47	72.59	72.63	76.42	72.47
DAM	18.95	19.94	19.58	26.86	31.08	33.32	28.82	27.46	31.30
ETS	66.45	72.39	70.72	78.71	65.42	78.77	78.90	71.83	74.31
ESA	65.75	58.20	64.70	68.09	77.99	70.84	69.00	79.60	81.82
GF	66.63	69.25	72.30	68.96	73.20	74.15	70.07	59.06	72.51
GTF	79.24	79.54	78.68	82.74	81.94	75.47	83.01	76.51	80.65
HA	34.95	32.14	37.22	47.71	42.32	48.00	47.83	45.40	46.26
OP	48.79	44.87	49.64	55.61	54.45	59.31	55.54	56.91	50.27
SH	81.14	77.71	80.22	81.21	81.17	85.41	81.23	88.50	89.15
STA	64.34	67.57	69.26	78.23	72.69	74.04	72.15	70.33	62.49
STO	71.21	61.09	61.13	70.26	71.31	71.56	62.66	68.55	73.25
TC	81.44	81.46	81.49	81.61	81.49	81.52	89.05	90.23	90.28
TS	47.31	47.33	44.76	54.86	60.04	55.47	58.09	48.80	58.62
VE	50.46	38.01	47.71	43.27	52.38	51.86	43.38	59.00	58.92
WM	65.21	60.24	65.04	65.52	69.99	66.40	65.36	64.69	70.34
mAP	59.54	57.55	60.06	63.87	64.41	65.10	64.20	66.14	67.66

APL: airplane, APO: airport, BF: baseball field, BC: basketball court, BR: bridge, CH: chimney, ETS: expressway toll station, ESA: expressway service area, DAM: dam, GF: golf field, GTF: ground track field, HA: harbor, OP: overpass, SH: ship, STA: stadium, STO: storage tank, TC: tennis court, TS: train station, VE: vehicle, WM: windmill.

Figure 7. The visualization of detection results on the DIOR-R dataset, the yellow ellipse means false detection or missed detection or inaccurate orientation prediction.

4.6. Ablation Experiments

In this section, we conducted experiments on DOTAv1.0 to explore the effects of multiscale training, multiple-scale cropping, and focal loss in the proposed VODet. The experimental results are shown in Table 4, where "MS_Train" refers to multiscale training, "SC_Crop" indicates single-scale cropping ("yes" for single-scale cropping and "no" for multiple-scale cropping), and "focal loss" denotes the use of focal loss as the classification loss. In the second and third rows of Table 4, we observed that multiscale training was beneficial for improving the detection performance when using single-scale cropping and focal loss simultaneously. Specifically, with multiscale training, $AP_{0.5:0.95}$, AP_{50}, and AP_{75} showed improvements of 1.16%, 0.94%, and 1.37%, respectively. After confirming the effectiveness of multiscale training, we conducted subsequent experiments under the premise of using multiscale training. In the third and fourth rows, when using single-scale cropping, we observed that the use of focal loss in VODet can degrade the detection performance. Similarly, the results in the fifth and sixth rows also demonstrated that focal loss led to worse detection performance when using multiple-scale cropping. These findings were consistent with YOLOv3 [69], which also experienced performance degradation when using focal loss for object detection. In the third and fifth rows, we noticed significant improvements in the $AP_{0.5:0.95}$, AP_{50}, and AP_{75} of 3.73%, 2.72%, and 4.68%, respectively, with the use of multiple-scale cropping. In the fourth and sixth rows, the corresponding improvements were 2.91%, 2.08%, and 5.34%, respectively. These results indicated that multiple-scale cropping can greatly enhance the detection performance compared to single-scale cropping. Furthermore, the substantial improvement in AP_{75} suggests that multiple-scale cropping can equip VODet with a more-accurate object location ability.

Table 4. Ablation experiments on the DOTA dataset.

MS_Train	SC_Crop	Focal Loss	$AP_{0.5:0.95}$	AP_{50}	AP_{75}
×	✓	✓	44.34	72.90	46.84
✓	✓	✓	45.50	73.84	48.21
✓	✓	×	46.74	75.68	49.07
✓	×	✓	49.23	76.56	52.89
✓	×	×	49.65	77.76	54.41

MS_Train means that the input size of VODet can be changed in the range of [imgsz $-$ 5 $\times$ 32, imgsz + 5 $\times$ 32], in which the imgsz is the preset size of VODet, 1024 $\times$ 1024 for the DOTA dataset. SC_Crop represents whether to use multiple-scale cropping. ✓ denotes cropping after resizing the original images at a scale of 1.0. × denotes cropping after resizing the original images at scales of (0.5, 1.0, 1.5).

4.7. The Comparison to Related Algorithms

To compare the vector-decomposition-based approach used in our proposed method, VODet, there are other vector-decomposition-based arbitrarily oriented object-detection methods available in the literature. To facilitate a comprehensive comparison of the experimental results among these related algorithms, we present the performance results of BBAVectors [8], ProjBB [9], RIE [10], AEDet [11], and our VODet in Table 5. It is evident from the table that our VODet achieved the best results when compared to the other related algorithms. Remarkably, even under the constraint of single-scale cropping, VODet outperformed the other algorithms. This comparison highlighted the superiority of VODet over its counterparts and demonstrated its significant potential for arbitrarily oriented object detection in remote sensing images.

Table 5. The comparison of related algorithms on the DOTA dataset, ‡ denotes the training and testing of multiple-scale cropping.

Methods	BBAVectors [8]	ProjBB [9]	RIE [10]	AEDet [11]	VODet	VODet ‡
Backbone	ResNet-101	ResNet-101	HRGANet-W48	CSPDarknet-53	CSPDarknet-53	CSPDarknet-53
PL	88.35	88.96	89.23	87.46	86.34	88.83
BD	79.96	79.32	84.86	77.64	79.95	83.58
BR	50.69	53.98	55.69	51.71	52.43	53.17
GTF	62.18	70.21	70.32	68.21	67.90	78.70
SV	78.43	60.67	75.76	77.99	79.32	79.88
LV	78.98	76.20	80.68	80.53	83.85	84.15
SH	87.94	89.71	86.14	86.53	87.87	88.55
TC	90.85	90.22	90.26	90.33	90.85	90.82
BC	83.58	78.94	80.17	80.75	87.57	88.12
ST	84.35	76.82	81.34	75.44	85.57	86.18
SBF	54.13	60.49	59.36	54.63	63.26	62.38
RA	60.24	63.62	63.24	59.64	61.19	67.65
HA	65.22	73.12	74.12	73.35	75.75	76.91
SP	64.28	71.43	70.87	73.76	78.87	79.69
HC	55.70	61.96	60.36	53.82	54.53	57.81
mAP	72.32	73.03	74.83	72.79	75.68	77.76

4.8. Model Parameters and Inference Time

Because the proposed VODet is similar to AEDet, in this section, we compared the model parameters and inference time between VODet and AEDet to highlight the speed and precision advantages of VODet. The model parameters and inference time are listed in Table 6. Since VODet and AEDet have the same number of prediction parameters, their model parameters were identical, specifically 27.28 M parameters on the DOTA dataset with 15 categories and 27.29 M parameters on the DIOR-R dataset with 20 categories. Regarding the inference time, the bounding box transformation in VODet is fully parallel, which resulted in a faster inference time compared to AEDet with incomplete bounding box parallel transformation. As shown in Table 6, VODet was almost 3.5× faster than AEDet on the DOTA and DIOR-R datasets. Moreover, VODet exhibited better detection performance than AEDet. In conclusion, the proposed VODet achieved improvements in speed and precision compared to AEDet, making it highly advantageous for arbitrarily oriented object detection.

Table 6. The model parameters and inference time of different methods on the DOTA and DIOR-R datasets.

Datasets	Methods	AP_{50}	Model Parameters (M)	Inference Time (ms)
DOTA	AEDet	72.79	27.28	87.44
	VODet	77.76	27.28	25.89
DIOR-R	AEDet	66.14	27.29	69.86
	VODet	67.66	27.29	18.86

5. Discussion

The proposed VODet achieved remarkable detection results on HRSC2016, DOTA, and DIOR-R. However, it is worth noting that the current approach only utilizes vector decomposition to describe bounding boxes and employs a simple coarse-to-fine positive–negative -sample-selection method. Therefore, there are still other strategies that can be explored to further enhance the detection accuracy of the proposed VODet. In our VODet, we utilized CSPDarknet-53 as the backbone to extract features. Although CSPDarknet-53 is effective, there are more-powerful backbones available. We believe that replacing CSPDarknet-53 with a stronger backbone can lead to significant improvements in detection performance. Another area for improvement lies in the matching of positive and negative samples. Adopting a more-reasonable and -effective positive- and negative-sample-matching method can enhance the overall performance of VODet. Additionally, the bounding box regression loss currently employed was the L1 loss, which optimizes each prediction vector individually. Exploring joint optimization losses, such as the IoU-based loss, based on the representation method in this paper, can potentially enhance the training process of VODet.

In our future work, we plan to focus on further improving the detection performance of VODet by addressing the above-mentioned aspects and incorporating related strategies described earlier.

6. Conclusions

In this paper, we analyzed the advantages and disadvantages of vector-decomposition-based arbitrarily oriented object-detection methods. Based on this analysis, we proposed a new, simple, and efficient arbitrarily oriented object detector called VODet, built upon YOLOX. In VODet, we treated the long side of the oriented bounding box as a vector and decomposed it into horizontal and vertical vectors. This allowed us to encode the angle information of the oriented bounding box within the two decomposed vectors. To enable the transformation from angle-based representation to vector-decomposition-based representation, we introduced the ATVEncode and VTADecode modules, which significantly reduced the data-processing time. Furthermore, we presented a straightforward positive–negative-sample-matching method, which dynamically matched the positive and negative samples from coarse to fine. Through a thorough analysis of vector-decomposition-based arbitrarily oriented object-detection methods and the efficient implementation of VODet, we achieved impressive mAPs of 90.23%, 77.76%, and 67.66% on HRSC2016, DOTA, and DIOR-R, respectively. These results demonstrated the effectiveness of VODet through extensive experiments. Additionally, we conducted a comparison among various vector-decomposition-based arbitrarily oriented object-detection methods, and our proposed VODet outperformed other related algorithms in terms of both efficiency and accuracy. Its simple design made VODet fast, accurate, and a potential candidate for arbitrarily oriented object detection.

Author Contributions: K.Z., H.W. and M.Z. provided the ideas; K.Z. and Y.D. implemented the algorithm; J.T. and S.Z. processed the needed datasets. K.Z. and M.Z. wrote the paper; M.Z. and H.W. revised the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 12003018), the Fundamental Research Funds for the Central Universities (No. XJS191305), and the China Postdoctoral Science Foundation (No. 2018M633471).

Data Availability Statement: Public datasets were used in this study; no new data were created nor analyzed. Data sharing is not applicable to this article.

Acknowledgments: Thanks to the authors of HRSC2016, DOTA, and DIOR-R for providing the nice datasets and the authors of YOLOX for contributing an excellent object-detection method.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, X.; Yan, J. Arbitrarily oriented object detection with circular smooth label. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 677–694.
2. Ming, Q.; Zhou, Z.; Miao, L.; Zhang, H.; Li, L. Dynamic anchor learning for arbitrarily oriented object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 2355–2363.
3. Fu, K.; Chang, Z.; Zhang, Y.; Sun, X. Point-based estimator for arbitrarily oriented object detection in aerial images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4370–4387. [CrossRef]
4. Xu, Y.; Fu, M.; Wang, Q.; Wang, Y.; Chen, K.; Xia, G.S.; Bai, X. Gliding vertex on the horizontal bounding box for multi-oriented object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 1452–1459. [CrossRef]
5. Yang, X.; Yan, J.; Ming, Q.; Wang, W.; Zhang, X.; Tian, Q. Rethinking rotated object detection with gaussian wasserstein distance loss. In Proceedings of the International Conference on Machine Learning, Online, 18–24 July 2021; pp. 11830–11841.
6. Qian, W.; Yang, X.; Peng, S.; Yan, J.; Guo, Y. Learning modulated loss for rotated object detection. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 2458–2466.
7. Yang, X.; Hou, L.; Zhou, Y.; Wang, W.; Yan, J. Dense label encoding for boundary discontinuity free rotation detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 15819–15829.

8. Yi, J.; Wu, P.; Liu, B.; Huang, Q.; Qu, H.; Metaxas, D. Oriented object detection in aerial images with box boundary-aware vectors. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Online, 5–9 January 2021; pp. 2150–2159.
9. Wu, Q.; Xiang, W.; Tang, R.; Zhu, J. Bounding Box Projection for Regression Uncertainty in Oriented Object Detection. *IEEE Access* **2021**, *9*, 58768–58779. [CrossRef]
10. He, X.; Ma, S.; He, L.; Ru, L.; Wang, C. Learning Rotated Inscribed Ellipse for Oriented Object Detection in Remote Sensing Images. *Remote Sens.* **2021**, *13*, 3622. [CrossRef]
11. Zhou, K.; Zhang, M.; Zhao, H.; Tang, R.; Lin, S.; Cheng, X.; Wang, H. Arbitrarily oriented Ellipse Detector for Ship Detection in Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 7151–7162. [CrossRef]
12. Jiang, X.; Xie, H.; Chen, J.; Zhang, J.; Wang, G.; Xie, K. Arbitrary-Oriented Ship Detection Method Based on Long-Edge Decomposition Rotated Bounding Box Encoding in SAR Images. *Remote Sens.* **2023**, *15*, 673. [CrossRef]
13. Yang, X.; Yang, X.; Yang, J.; Ming, Q.; Wang, W.; Tian, Q.; Yan, J. Learning high-precision bounding box for rotated object detection via kullback-leibler divergence. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 18381–18394.
14. Liu, Z.; Wang, H.; Weng, L.; Yang, Y. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1074–1078. [CrossRef]
15. Cheng, G.; Wang, J.; Li, K.; Xie, X.; Lang, C.; Yao, Y.; Han, J. Anchor-free oriented proposal generator for object detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [CrossRef]
16. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3974–3983.
17. Yang, L.; Chen, Y.; Song, S.; Li, F.; Huang, G. Deep Siamese networks based change detection with remote sensing images. *Remote Sens.* **2021**, *13*, 3394. [CrossRef]
18. Zhu, Q.; Guo, X.; Deng, W.; Shi, S.; Guan, Q.; Zhong, Y.; Zhang, L.; Li, D. Land-use/land-cover change detection based on a Siamese global learning framework for high spatial resolution remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* **2022**, *184*, 63–78. [CrossRef]
19. Zhang, C.; Wang, L.; Cheng, S.; Li, Y. SwinSUNet: Pure transformer network for remote sensing image change detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13. [CrossRef]
20. Fu, K.; Chang, Z.; Zhang, Y.; Xu, G.; Zhang, K.; Sun, X. Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2020**, *161*, 294–308. [CrossRef]
21. Rabbi, J.; Ray, N.; Schubert, M.; Chowdhury, S.; Chao, D. Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network. *Remote Sens.* **2020**, *12*, 1432. [CrossRef]
22. Zhou, K.; Zhang, M.; Lin, S.; Zhang, R.; Wang, H. Single-stage object detector with local binary pattern for remote sensing images. *Int. J. Remote Sens.* **2023**, *44*, 4137–4162. [CrossRef]
23. Cheng, X.; Zhang, M.; Lin, S.; Zhou, K.; Zhao, S.; Wang, H. Two-Stream Isolation Forest Based on Deep Features for Hyperspectral Anomaly Detection. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [CrossRef]
24. Wang, M.; Wang, Q.; Hong, D.; Roy, S.K.; Chanussot, J. Learning tensor low-rank representation for hyperspectral anomaly detection. *IEEE Trans. Cybern.* **2022**, *53*, 679–691. [CrossRef]
25. Lin, S.; Zhang, M.; Cheng, X.; Wang, L.; Xu, M.; Wang, H. Hyperspectral anomaly detection via dual dictionaries construction guided by two-stage complementary decision. *Remote Sens.* **2022**, *14*, 1784. [CrossRef]
26. Tang, T.; Zhou, S.; Deng, Z.; Lei, L.; Zou, H. Arbitrarily oriented vehicle detection in aerial imagery with single convolutional neural networks. *Remote Sens.* **2017**, *9*, 1170. [CrossRef]
27. Liu, Z.; Hu, J.; Weng, L.; Yang, Y. Rotated region based CNN for ship detection. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 900–904.
28. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning RoI transformer for oriented object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2849–2858.
29. Han, J.; Ding, J.; Li, J.; Xia, G.S. Align deep features for oriented object detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–11. [CrossRef]
30. Yang, X.; Yan, J.; Feng, Z.; He, T. R3det: Refined single-stage detector with feature refinement for rotating object. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 3163–3171.
31. Chen, Z.; Chen, K.; Lin, W.; See, J.; Yu, H.; Ke, Y.; Yang, C. Piou loss: Towards accurate oriented object detection in complex environments. In Proceedings of the European Conference on Computer Vision, Online, 23–28 August 2020; pp. 195–211.
32. Yang, X.; Yan, J. On the arbitrarily oriented object detection: Classification based approaches revisited. *Int. J. Comput. Vis.* **2022**, *130*, 1340–1365. [CrossRef]
33. Han, J.; Ding, J.; Xue, N.; Xia, G.S. Redet: A rotation-equivariant detector for aerial object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 2786–2795.
34. Ming, Q.; Miao, L.; Zhou, Z.; Dong, Y. CFC-Net: A critical feature capturing network for arbitrarily oriented object detection in remote-sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14. [CrossRef]
35. Yang, X.; Yan, J.; Liao, W.; Yang, X.; Tang, J.; He, T. Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 2384–2399. [CrossRef] [PubMed]

36. Lu, D.; Li, D.; Li, Y.; Wang, S. OSKDet: Orientation-sensitive keypoint localization for rotated object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1182–1192.
37. Li, W.; Chen, Y.; Hu, K.; Zhu, J. Oriented reppoints for aerial object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 1829–1838.
38. Guo, Z.; Liu, C.; Zhang, X.; Jiao, J.; Ji, X.; Ye, Q. Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 8792–8801.
39. Dai, P.; Yao, S.; Li, Z.; Zhang, S.; Cao, X. ACE: Anchor-free corner evolution for real-time arbitrarily-oriented object detection. *IEEE Trans. Image Process.* **2022**, *31*, 4076–4089. [CrossRef] [PubMed]
40. Sun, Y.; Sun, X.; Wang, Z.; Fu, K. Oriented ship detection based on strong scattering points network in large-scale SAR images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–18. [CrossRef]
41. Fu, K.; Fu, J.; Wang, Z.; Sun, X. Scattering-keypoint-guided network for oriented ship detection in high-resolution and large-scale SAR images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 11162–11178. [CrossRef]
42. Cui, Z.; Leng, J.; Liu, Y.; Zhang, T.; Quan, P.; Zhao, W. SKNet: Detecting rotated ships as keypoints in optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 8826–8840. [CrossRef]
43. Chen, X.; Ma, L.; Du, Q. Oriented object detection by searching corner points in remote sensing imagery. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
44. Zhou, J.; Zhang, R.; Zhao, W.; Shen, S.; Wang, N. APS-Net: An Adaptive Point Set Network for Optical Remote-Sensing Object Detection. *IEEE Geosci. Remote Sens. Lett.* **2022**, *20*, 1–5. [CrossRef]
45. Zhang, F.; Wang, X.; Zhou, S.; Wang, Y.; Hou, Y. Arbitrarily oriented ship detection through center-head point extraction. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–14.
46. Zhou, Q.; Yu, C. Point rcnn: An angle-free framework for rotated object detection. *Remote Sens.* **2022**, *14*, 2605. [CrossRef]
47. Wang, J.; Yang, L.; Li, F. Predicting arbitrarily oriented objects as points in remote sensing images. *Remote Sens.* **2021**, *13*, 3731. [CrossRef]
48. Wei, H.; Zhang, Y.; Chang, Z.; Li, H.; Wang, H.; Sun, X. Oriented objects as pairs of middle lines. *ISPRS J. Photogramm. Remote Sens.* **2020**, *169*, 268–279. [CrossRef]
49. He, Y.; Gao, F.; Wang, J.; Hussain, A.; Yang, E.; Zhou, H. Learning polar encodings for arbitrarily oriented ship detection in SAR images. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2021**, *14*, 3846–3859. [CrossRef]
50. Zhou, L.; Wei, H.; Li, H.; Zhao, W.; Zhang, Y.; Zhang, Y. Arbitrarily oriented object detection in remote sensing images based on polar coordinates. *IEEE Access* **2020**, *8*, 223373–223384. [CrossRef]
51. Zhao, P.; Qu, Z.; Bu, Y.; Tan, W.; Guan, Q. Polardet: A fast, more precise detector for rotated target in aerial images. *Int. J. Remote Sens.* **2021**, *42*, 5831–5861. [CrossRef]
52. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3520–3529.
53. Yang, X.; Zhang, G.; Li, W.; Wang, X.; Zhou, Y.; Yan, J. H2RBox: Horizonal Box Annotation is All You Need for Oriented Object Detection. *arXiv* **2022**, arXiv:2210.06742.
54. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
55. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
56. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrarily oriented scene text detection via rotation proposals. *IEEE Trans. Multimedia* **2018**, *20*, 3111–3122. [CrossRef]
57. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. R2CNN: Rotational region CNN for orientation robust scene text detection. *arXiv* **2017**, arXiv:1706.09579.
58. Wang, J.; Yang, W.; Li, H.C.; Zhang, H.; Xia, G.S. Learning center probability map for detecting objects in aerial images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4307–4323. [CrossRef]
59. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.S.; Bai, X. Rotation-sensitive regression for oriented scene text detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5909–5918.
60. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
61. Pan, X.; Ren, Y.; Sheng, K.; Dong, W.; Yuan, H.; Guo, X.; Ma, C.; Xu, C. Dynamic refinement network for oriented and densely packed object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11207–11216.
62. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
63. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Sun, X.; Fu, K. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8232–8241.
64. Cheng, G.; Yao, Y.; Li, S.; Li, K.; Xie, X.; Wang, J.; Yao, X.; Han, J. Dual-aligned oriented detector. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [CrossRef]

65. Yang, L.; Zheng, Z.; Wang, J.; Song, S.; Huang, G.; Li, F. An Adaptive Object Detection System based on Early-exit Neural Networks. *IEEE Trans. Cogn. Dev. Syst.* **2023**. [CrossRef]
66. Li, K.; Cheng, G.; Bu, S.; You, X. Rotation-insensitive and context-augmented object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 2337–2348. [CrossRef]
67. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28.
68. Yao, Y.; Cheng, G.; Wang, G.; Li, S.; Zhou, P.; Xie, X.; Han, J. On Improving Bounding Box Representations for Oriented Object Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *61*, 1–11. [CrossRef]
69. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.