remote sensing

MDPI

Technical Note

Recalibrating Features and Regression for Oriented Object Detection

Weining Chen [1,2,3,†], Shicheng Miao [1,2,†], Guangxing Wang [1,2] and Gong Cheng [1,2,*]

[1] School of Automation, Northwestern Polytechnical University, Xi'an 710129, China
[2] Research & Development Institute of Northwestern Polytechnical University in Shenzhen, Shenzhen 518057, China
[3] Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China
* Correspondence: gcheng@nwpu.edu.cn
† These authors contributed equally to this work.

Abstract: The objects in remote sensing images are normally densely packed, arbitrarily oriented, and surrounded by complex backgrounds. Great efforts have been devoted to developing oriented object detection models to accommodate such data characteristics. We argue that an effective detection model hinges on three aspects: feature enhancement, feature decoupling for classification and localization, and an appropriate bounding box regression scheme. In this article, we instantiate the three aspects on top of the classical Faster R-CNN, with three novel components proposed. First, we propose a weighted fusion and refinement (WFR) module, which adaptively weighs multi-level features and leverages the attention mechanism to refine the fused features. Second, we decouple the RoI (region of interest) features for the subsequent classification and localization via a lightweight affine transformation-based feature decoupling (ATFD) module. Third, we propose a post-classification regression (PCR) module for generating the desired quadrilateral bounding boxes. Specifically, PCR predicts the precise vertex location on each side of a predicted horizontal box, by simply learning the following: (i) classify the discretized regression range of the vertex, and (ii) revise the vertex location with an offset. We conduct extensive experiments on the DOTA, DIOR-R, and HRSC2016 datasets to evaluate our method.

Keywords: oriented object detection; feature enhancement; feature decoupling; bounding box regression

Citation: Chen, W.; Miao, S.; Wang, G.; Cheng, G. Recalibrating Features and Regression for Oriented Object Detection. *Remote Sens.* **2023**, *15*, 2134. https://doi.org/10.3390/rs15082134

Academic Editors: Xu Tang, Yansheng Li, Lichao Mou, Xiangrong Zhang and Licheng Jiao

Received: 22 February 2023
Revised: 15 April 2023
Accepted: 15 April 2023
Published: 18 April 2023

1. Introduction

Object detection is fundamental, yet challenging. As the objects in remote sensing images are normally densely packed and arbitrarily oriented [1,2], oriented object detection has become a research hotspot over the past few years. It requires rotated or quadrilateral bounding boxes to compactly enclose arbitrarily oriented objects [2]. This line of research has achieved considerable success, driven by available large-scale remote sensing datasets [3–5], high-performance computation platforms, and recent advances in deep learning [6–10].

The current deep learning-based detection frameworks are well shaped. We focus on the two-stage frameworks [11,12]. In general, the classical pipeline (e.g., that of Faster R-CNN [6]) proceeds in two steps. First, a region proposal network (RPN) generates several object proposals based on the multi-level features extracted by a general backbone [13], possibly with a feature pyramid network (FPN) [7,14]. Second, a series of RoI (region of interest)-related operations (e.g., RoI Pooling and RoI Align) produce RoI features, which are sent to a single detection head to accomplish both classification and localization. There remains much room for improvement, including architectural details, strategic designs, etc. In relation to our work, the following are the most concerning issues: (i) the multi-level feature fusion of the FPN for feature enhancement [8,9,15], and (ii) the fact that the learning

mechanisms of the tasks of classification and localization are divergent but use the same RoI features, which can result in feature misalignment and conflict between the two tasks [1,16].

Features matter. In a deep learning model, deep semantic information and shadow detailed information play complementary roles in object detection. A classical FPN provides a top–down pathway that introduces high-level semantic information into shadow features. In order to achieve comprehensive feature fusion among different levels, some researchers have made valid modifications based on the FPN structure [8,9], yielding enhanced feature pyramids. It is worth noting that ability to stress local information is also necessary so as to tackle the information redundancy induced by feature fusion [17]. Apart from building holistic representations, we regard catering to the preference of subsequent tasks as part of feature engineering. Object detection ends with a combination of classification and regression (localization). The two tasks show different emphases on RoI features. Thus, obtaining task-specific RoI features deserves effort and falls exactly into the concept of feature decoupling [16,18,19].

To a large extent, the emphasis laid on object orientations tells apart conventional object detection from oriented object detection. There exist a large number of methods that use rotated bounding boxes to locate arbitrarily oriented objects in remote sensing images [11,20,21] by means of introducing an additional angle prediction branch to the conventional object detection framework (e.g., Faster R-CNN [6]). Irrespective of its simplicity in implementation, directly adding an angle prediction branch suffers from non-negligible drawbacks. In such detectors, a minor angle deviation could lead to a significant IoU (intersection over union) drop, and, thus, result in inaccurate localization. In other words, the final detection performance is highly reliant on the predicted angles. Unfortunately, the additional angle prediction branch itself is prone to causing training instability because of the problems of angular periodicity and bounding box boundary discontinuity [2,22]. In order to overcome the drawbacks of angle prediction, a few methods [23,24] employ quadrilateral bounding boxes to locate oriented objects. Specifically, Xu et al. [23] proposed gliding the four vertices of a horizontal bounding box to generate the desired quadrangle. Qian et al. [24] focused on directly predicting the locations of the four vertices of a quadrangle. In this article, we also focus on using quadrilateral bounding boxes to locate oriented objects.

With the above analyses in mind, we argue that an effective detection model hinges on three aspects: feature enhancement, feature decoupling for classification and localization, and an appropriate bounding box regression scheme. In this article, we select the classical Faster R-CNN [6] as the basic detector, and propose three novel components to strengthen it. First, a weighted fusion and refinement (WFR) module accounts for feature enhancement. It adaptively weighs the multi-level features produced by FPN and leverages the attention mechanism [25–28] to refine the fused features. Second, a lightweight affine transformation-based feature decoupling (ATFD) module produces decoupled features for subsequent classification and localization. Third, a post-classification regression (PCR) module facilitates regressing quadrilateral bounding boxes. Given a predicted horizontal box, we discretize the regression range (i.e., box height or width) of each vertex into several bins and apply a simple classifier to predict which bin a vertex should belong to, so that only the vertex offsets with regard to corresponding ground truths need to be regressed.

This article is extended from our conference version [29]. In contrast with our preliminary work, this long version includes the following contributions. We propose the WFR module, which contains two types of feature enhancement methods. The weighted fusion part supplements high-level deep features (low-level shadow features) with detailed information (semantic information). The attention-based feature refinement part is used to highlight discriminative regions. Overall, this feature enhancement process, along with feature decoupling and the bounding box regression scheme, shape our three-aspect considerations for building a strong oriented object detection model. We conduct more extensive experimental evaluations on three large-scale datasets. The results validate our considerations and the corresponding modules proposed.

2. Related Work

2.1. Feature Fusion

Deep learning models extract multi-level features. In general, high-level deep features convey rich semantic information, while low-level shadow ones possess abundant detailed information related to specific objects. Most advanced vision methods [7,30] set multi-level feature fusion as default to deal with the scale shift of objects. From another point of view, this suggests that both semantic information and detailed information are crucial for accurate object detection.

Feature Pyramid Network (FPN) [7] utilizes a top–down pathway to enrich the shadow features with high-level semantic information. Liu et al. [8] introduced a bottom–up pathway, based on the FPN, to supplement deep features with low-level detailed information. Kong et al. [17] combined the features across different spatial locations and scaled with lightweight global attention and local reconfigurations. Zhao et al. [31] obtained multi-level feature pyramids via alternating two U-shaped modules. Instead of hand-drafted architectural designs, Ghiasi et al. [9] obtained pyramidal architectures by applying the neural architecture search technique within a scalable search space.

We conduct a weighted feature fusion of the pyramidal features produced by FPN. The weights are data-relevant and adaptively learned for each pyramidal level. The fused features go through an attention-based feature refinement procedure to implicitly alleviate the information redundancy induced by feature fusion.

2.2. Feature Decoupling

The classification and localization tasks in object detection are commonly completed by a shared detection head in most advanced detectors. Normally, the classification task favors discriminative regions, while the localization task works on precisely locating objects with their boundaries. In light of this, some researchers have paid close attention to the feature misalignment and conflict arising from using the same features for the two different tasks [16,18,19].

Jiang et al. [18] found that those features generating high classification scores usually produce coarse bounding boxes. To compensate, they introduced an extra branch to predict the IoUs between bounding boxes and corresponding ground truths as the localization scores, and used the products of classification scores and localization scores as the sorting criterion in the non-maximum suppression (NMS) post-processing procedure. Although the sorting criterion is relatively reliable, the two scores still originate from the same features. Wu et al. [19] proposed Double-Head R-CNN that utilizes two specific branches for classification and localization, separately. Double-Head R-CNN effectively disentangles the shared information and parameters of the two tasks. However, the problems of feature misalignment and conflict still exist because the RoI features fed into the two branches are generated from the same proposals. More recently, Song et al. [16] proposed generating disentangled proposals for the classification and localization tasks so as to decouple the two tasks from the spatial dimension.

In contrast to the above-mentioned methods, we alleviate the problems of feature misalignment and conflict by using lightweight affine transformation to derive transformation-invariant features for classification, and, thus, achieve decoupling of the RoI features.

2.3. Oriented Bounding Boxes

There have been many strategies proposed for obtaining the desired rotated or quadrilateral bounding boxes. Ma et al. [20] directly placed a large number of rotated anchors on each location of feature maps to generate proposals, incurring heavy computational burdens. Ding et al. [11] proposed generating rotated proposals from horizontal ones, avoiding the need for dense rotated anchors, but incurring expensive computational costs. There are also some methods using horizontal proposals to generate rotated or quadrilateral bounding boxes [21,32,33]. For instance, Yang et al. [21] introduced an angle-related parameter to the Faster R-CNN head and located arbitrarily oriented objects using rotated rectangles.

Their extended work [33] further investigated instance-level denoising to enhance the detection of small and cluttered objects. Xu et al. [23] and Qian et al. [24] utilized quadrangles to locate oriented objects. Specifically, Xu et al. [23] described an oriented object using the gliding offsets of four vertices, in addition to the conventional horizontal bounding box representation. Qian et al. [24] directly regressed the four vertices of a quadrangle and devised a modulated rotation loss to optimize the regression process, largely eliminating the problems of angular periodicity and bounding box boundary discontinuity [2,22]. Apart from the above strategies, a series of works [34,35] model the rotated objects as Gaussian distributions and build the regression loss with certain distributional distance measurements, e.g., Wasserstein distance and Kullback–Leibler Divergence.

We use quadrilateral bounding boxes to locate oriented objects and propose a post-classification regression module. Given a horizontal bounding box, we discretize the regression range (i.e., box height or width) of each vertex into several bins. A simple classifier is used to predict which bin a vertex should belong to, followed by the regression of the offsets with regard to corresponding ground truths.

3. Methodology

We describe our method in detail in this section. We have three-aspect considerations for building a strong oriented object detection model: feature enhancement, feature decoupling for classification and localization, and an appropriate bounding box regression scheme. Correspondingly, we propose a weighted fusion and refinement (WFR) module, a lightweight affine transformation-based feature decoupling (ATFD) module, and a post-classification regression (PCR) module. We implement them on top of the classical Faster R-CNN [6], as shown in Figure 1. Note that the detailed architectures of the backbone in the feature pyramidal network (e.g., Backbone-FPN in Figure 1) and the region proposal network (RPN) are omitted for simplicity.

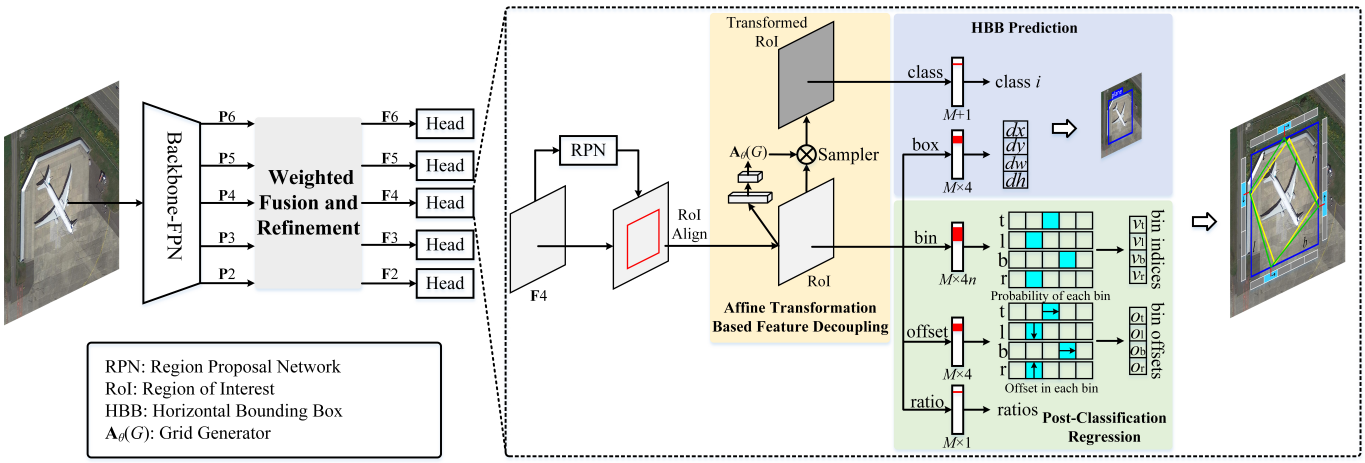


Figure 1. Illustration of our method. On top of Faster R-CNN, we introduce a weighted fusion and refinement (WFR) module, an affine transformation-based feature decoupling (ATFD) module, and a post-classification regression (PCR) module. WFR receives the pyramidal features $\mathbf{P}*$ and derives an enhanced feature pyramid $\mathbf{F}*$. After obtaining the RoI features, ATFD performs task-specific feature decoupling for the subsequent classification and regression. The regression is split into the prediction of regular HBB and that of orientation-relevant parameters responsible for converting the predicted HBB into quadrilateral bounding boxes. PCR facilitates the prediction of orientation-relevant parameters. M The number of classes is in the HBB prediction part. n The number of bins is in the post-classification regression module.

We briefly introduce the detection pipeline of our method below. See Figure 1 for an illustration. The backbone extracts multi-level features of the input images and an FPN enriches the semantic information of the shallow features. The output pyramidal

features produced by FPN, indicated as **P**2∼**P**6, are sent to our WFR module for feature enhancement. We denote the enhanced pyramidal features as **F**2∼**F**6, and collectively refer to the rest of the components as the head. In the head, a standard RPN utilizes **F**2∼**F**6 to generate horizontal proposals, separately. After that, RoI (region of interest) features are obtained through a series of RoI-related operations (e.g., RoI Pooling and RoI Align). The ATFD module is introduced to decouple the RoI features for subsequent classification and localization. The detection framework ends with a horizontal bounding box (HBB) prediction part and our PCR module. The HBB prediction part regresses horizontal bounding boxes and completes classification, based on the two sets of decoupled RoI features. Our PCR module predicts orientation-relevant parameters, with which we can easily convert the horizontal bounding boxes into quadrilateral bounding boxes.

3.1. Weighted Fusion and Refinement

The WFR module accounts for feature enhancement. It receives the pyramidal features produced by FPN and outputs the enhanced pyramidal features. We aim for task-relevant foregrounds to be emphasized and task-irrelevant backgrounds to be suppressed [36]. To this end, we conducted a weighted fusion and use the attention mechanism to refine the fused features, as shown in Figure 2.

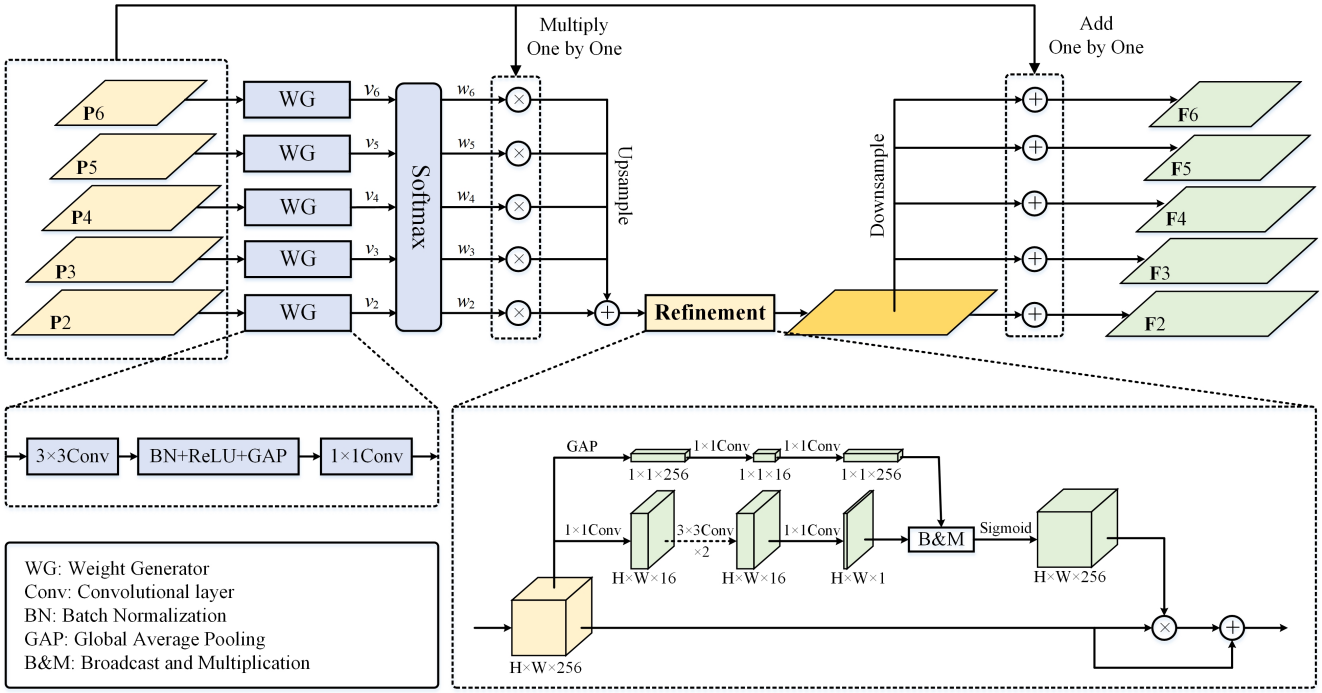


Figure 2. Illustration of our proposed WFR module.

We use a straightforward implementation to achieve adaptively weighted fusion, with the aim of supplementing the pyramidal features with useful information. Given **P**2∼**P**6, we put a separate weight generator at each level to measure the importance of the features at the corresponding level. Each weight generator is in the form of a two-layer convolutional neural network. The outputs are scalars, indicated as $v_2 \sim v_6$. The final weights $w_2 \sim w_6$ are obtained through a Softmax function as follows:

$$w_i = \frac{e^{v_i}}{\sum_{j=2}^{6} e^{v_j}}, i = 2, 3, \ldots, 6 \quad (1)$$

$w_2 \sim w_6$ are assigned to the features at corresponding levels. Then, the weighted features are uniformly upsampled to the size of **P**2, so as to be fused by element-wise summation.

We denote the fused features as $\mathbf{F} \in \mathbb{R}^{H \times W \times 256}$. The values H and W are height and width, respectively. The channel number of $\mathbf{F}$ is 256, the same as that of the pyramidal features produced by FPN.

We employ an attention-based refinement network mainly composed of convolutional layers. The refinement network receives $\mathbf{F}$ as input. There are two branches realizing channel and spatial attention in parallel. The channel attention branch contains global average pooling and two 1×1 convolutional layers to generate channel weights with a size of $1 \times 1 \times 256$. Meanwhile, the spatial attention branch contains two 1×1 convolutional layers and two 3×3 convolutional layers to generate spatial weights with a size of $H \times W \times 1$. The two forms of weights are broadcast to the same size of $\mathbf{F}$, i.e., $H \times W \times 256$, and fused by element-wise multiplication. Denoting the final attention weights as $\mathbf{W}_{\text{att}}$ and the refined features as $\hat{\mathbf{F}}$, we formulate the operating steps of the refinement network as follows:

$$\begin{aligned} \hat{\mathbf{F}} &= (\mathbf{W}_{\text{att}} \otimes \mathbf{F}) \oplus \mathbf{F} \\ &= \{\sigma[f_{1\times1}(f_{1\times1}(f_{\text{gap}}(\mathbf{F}))) \otimes f_{1\times1}(f_{3\times3}(f_{3\times3}(f_{1\times1}(\mathbf{F}))))] \otimes \mathbf{F}\} \oplus \mathbf{F} \end{aligned} \tag{2}$$

where $f_{1\times1}$, $f_{3\times3}$, and f_{gap} denote 1×1 convolution, 3×3 convolution, and global average pooling, respectively. σ denotes the sigmoid function.

After the above weighted fusion and refinement, the refined features $\hat{\mathbf{F}}$ are downsampled to accord with the original sizes of feature pyramids. Simple one-by-one element-wise summation operations yield the enhanced pyramidal features $\mathbf{F}2 \sim \mathbf{F}6$ to be sent to the head.

3.2. Affine Transformation-Based Feature Decoupling

The ATFD module aims at decoupling the RoI features $\mathbf{R} \in \mathbb{R}^{H' \times W' \times C}$ to tackle the feature misalignment and conflict resulting from the same features being used for the two different tasks. Our implementation is simple. In feature decoupling, we only fasten on classification and let the localization task directly operate on $\mathbf{R}$. Considering that the classification task prefers transformation-invariant features, we use affine transformation to derive the features for classification, indicated as $\hat{\mathbf{R}}$.

In general, the affine transformation is formulated as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A}_\theta \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{23} & \theta_{33} \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{3}$$

where (x_0, y_0) and (x, y) are the source coordinate and target coordinate, respectively. $\mathbf{A}_\theta \in \mathbb{R}^{2\times3}$ is the affine transformation matrix. $\boldsymbol{\theta}$ is a vector comprising the parameters of the affine transformation matrix, i.e., θ_{**}. Inspired by the Spatial Transformer Networks [37], we adopt a single fully-connected layer to predict $\boldsymbol{\theta}$, as shown in Figure 3.

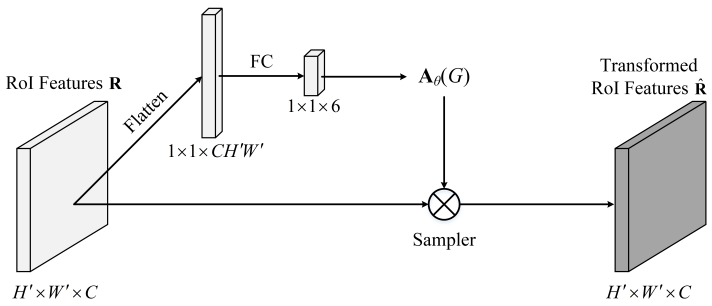


Figure 3. The architecture of the proposed ATFD module. FC denotes a fully-connected layer.

Our ATFD is lightweight. In the module, the RoI features $\mathbf{R}$ are first flattened into a vector with the size of $1 \times 1 \times CH'W'$. A single fully-connected layer (FC) maps the vector into $\boldsymbol{\theta}$. Then, a grid generator $\mathbf{A}_\theta(G)$ receives $\boldsymbol{\theta}$ to create a sampling grid, indicating which

elements of $\mathbf{R}$ should be sampled. Given $\mathbf{R}$ and $\mathbf{A}_\theta(G)$, a sampler produces the transformed RoI features $\hat{\mathbf{R}}$ used for classification. Note that a bilinear sampling operation is used in the sampler to tackle the misaligned mapping between the source and target coordinates. The process can be formulated as follows:

$$R_i^c = \sum_{j=1}^{H'} \sum_{k=1}^{W'} U_{jk}^c \max(0, 1 - \|x_i^s - k\|) \max(0, 1 - \|y_i^s - j\|), i = 1, 2, \ldots, H'W', c = 1, 2, \ldots, C \tag{4}$$

where R_i^c denotes the transformed RoI feature value for pixel i in channel c. U_{jk}^c denotes the original RoI feature value at location (j, k) in channel c. $\left(x_i^s, y_i^s\right)$ denotes the source coordinate of pixel i.

3.3. Post-Classification Regression

The PCR module converts the predicted horizontal bounding boxes into the desired quadrilateral bounding boxes, so as to compactly enclose the oriented objects in remote sensing images [2]. This is realized by precisely predicting the coordinates of vertices on each side of the predicted horizontal bounding boxes. Given a predicted horizontal box, we discretize the regression range (i.e., box height or width) of each vertex into several bins and apply a simple classifier to predict which bin a vertex should belong to, so that only the vertex offsets with regard to corresponding ground truths need to be regressed.

As shown in Figure 1, the PCR module consists of three branches, i.e., bin, offset, and ratio branches. The bin branch outputs a real-value vector for each side. We denote the indices with the largest values on the top, left, bottom, and right sides as $(v_\mathrm{t}, v_\mathrm{l}, v_\mathrm{b}, v_\mathrm{r})$. The offset branch outputs the predicted offsets between the center coordinates of the indexed bins and the coordinates of the exact vertices. We denote the four offsets as $(o_\mathrm{t}, o_\mathrm{l}, o_\mathrm{b}, o_\mathrm{r})$. The ratio branch outputs the area ratios of the predicted quadrilateral bounding boxes with regard to the predicted horizontal bounding boxes, indicated as *ratio* simply.

3.3.1. Training

The bin, offset, and ratio branches are all involved in the training phase. Below we elaborate on how to generate supervisory signals. Given an oriented ground truth box determined by the coordinates of four vertices $\left(x_1^*, y_1^*\right)$, $\left(x_2^*, y_2^*\right)$, $\left(x_3^*, y_3^*\right)$, $\left(x_4^*, y_4^*\right)$, its minimum enclosing rectangle can be generated according to the maximum values and the minimum values of its horizontal and vertical coordinates.

$$\begin{cases} x_\mathrm{min}^* = \min(x_1^*, x_2^*, x_3^*, x_4^*) \\ y_\mathrm{min}^* = \min(y_1^*, y_2^*, y_3^*, y_4^*) \\ x_\mathrm{max}^* = \max(x_1^*, x_2^*, x_3^*, x_4^*) \\ y_\mathrm{max}^* = \max(y_1^*, y_2^*, y_3^*, y_4^*) \end{cases} \tag{5}$$

$(x_\mathrm{min}^*, y_\mathrm{min}^*)$ and $(x_\mathrm{max}^*, y_\mathrm{max}^*)$ are the top-left and bottom-right vertices of the minimum enclosing rectangle, respectively.

We divide each side of the minimum enclosing rectangle into n bins. For each of the four sides, the center coordinate of the i-th bin can be obtained by:

$$\begin{cases} \boldsymbol{x}_\mathrm{t}^*(i) = x_\mathrm{min}^* + (0.5 + i) \times w_\mathrm{bin} \\ \boldsymbol{y}_\mathrm{l}^*(i) = y_\mathrm{min}^* + (0.5 + i) \times h_\mathrm{bin} \\ \boldsymbol{x}_\mathrm{b}^*(i) = x_\mathrm{max}^* - (0.5 + i) \times w_\mathrm{bin} \\ \boldsymbol{y}_\mathrm{r}^*(i) = y_\mathrm{max}^* - (0.5 + i) \times h_\mathrm{bin} \end{cases} \tag{6}$$

where w_bin and h_bin are the width and height of each bin, and $w_\mathrm{bin} = (x_\mathrm{max}^* - x_\mathrm{min}^*)/n$ and $h_\mathrm{bin} = (y_\mathrm{max}^* - y_\mathrm{min}^*)/n$. $\boldsymbol{x}_\mathrm{t}^*(i)$, $\boldsymbol{y}_\mathrm{l}^*(i)$, $\boldsymbol{x}_\mathrm{b}^*(i)$, and $\boldsymbol{y}_\mathrm{r}^*(i)$ are the center coordinate vectors of bins on the top, left, bottom, and right sides of the minimum enclosing rectangle, respectively. Assume that $\left(x_1^*, y_1^*\right)$, $\left(x_2^*, y_2^*\right)$, $\left(x_3^*, y_3^*\right)$, and $\left(x_4^*, y_4^*\right)$ are located on the top, left, bottom, and right edges of the minimum enclosing rectangle, respectively. The normal-

ized offsets between the center coordinates of bins and corresponding ground truths are obtained by:

$$\begin{cases} \boldsymbol{u}_{\mathrm{t}}^{*} = (x_{1}^{*} - \boldsymbol{x}_{\mathrm{t}}^{*})/w_{\mathrm{bin}} \\ \boldsymbol{u}_{\mathrm{l}}^{*} = (y_{2}^{*} - \boldsymbol{y}_{\mathrm{l}}^{*})/h_{\mathrm{bin}} \\ \boldsymbol{u}_{\mathrm{b}}^{*} = (x_{3}^{*} - \boldsymbol{x}_{\mathrm{b}}^{*})/w_{\mathrm{bin}} \\ \boldsymbol{u}_{\mathrm{r}}^{*} = (y_{4}^{*} - \boldsymbol{y}_{\mathrm{r}}^{*})/h_{\mathrm{bin}} \end{cases} \tag{7}$$

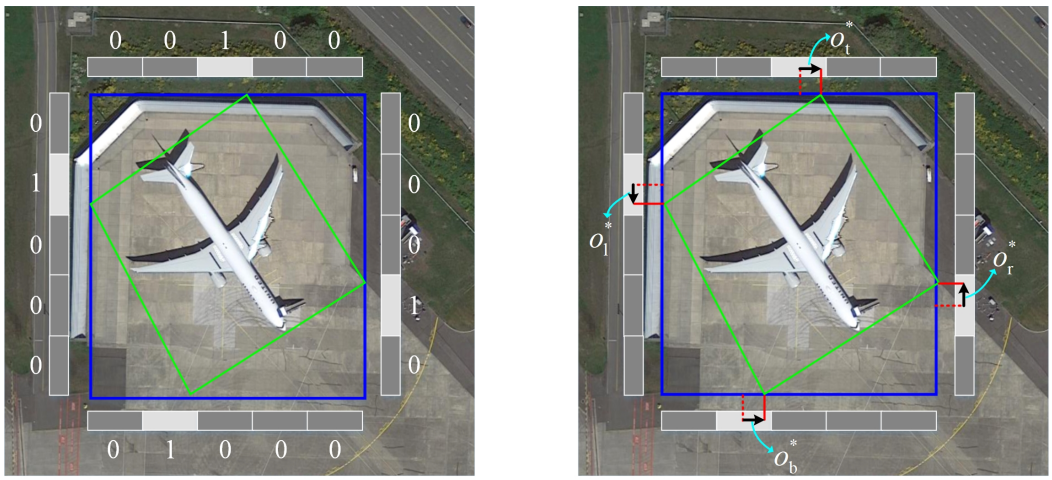
where $\boldsymbol{u}_{\mathrm{t}}^{*}$, $\boldsymbol{u}_{\mathrm{l}}^{*}$, $\boldsymbol{u}_{\mathrm{b}}^{*}$, and $\boldsymbol{u}_{\mathrm{r}}^{*}$ denote the normalized offset vectors on the four sides of the minimum enclosing rectangle. Denote the ground truth bin indices as $(v_{\mathrm{t}}^{*}, v_{\mathrm{l}}^{*}, v_{\mathrm{b}}^{*}, v_{\mathrm{r}}^{*})$. The normalized offsets within corresponding bins can be easily obtained by indexing the offset vectors as follows:

$$\begin{cases} o_{\mathrm{t}}^{*} = \boldsymbol{u}_{\mathrm{t}}^{*}(v_{\mathrm{t}}^{*}) \\ o_{\mathrm{l}}^{*} = \boldsymbol{u}_{\mathrm{l}}^{*}(v_{\mathrm{l}}^{*}) \\ o_{\mathrm{b}}^{*} = \boldsymbol{u}_{\mathrm{b}}^{*}(v_{\mathrm{b}}^{*}) \\ o_{\mathrm{r}}^{*} = \boldsymbol{u}_{\mathrm{r}}^{*}(v_{\mathrm{r}}^{*}) \end{cases} \tag{8}$$

The one-hot vectors converted from $(v_{\mathrm{t}}^{*}, v_{\mathrm{l}}^{*}, v_{\mathrm{b}}^{*}, v_{\mathrm{r}}^{*})$ serve as the bin labels for classification in the bin branch. The normalized offsets $(o_{\mathrm{t}}^{*}, o_{\mathrm{l}}^{*}, o_{\mathrm{b}}^{*}, o_{\mathrm{r}}^{*})$ are used as the training targets for regression in the offset branch, respectively. Figure 4 gives an intuitive illustration of the bin labels and the offset targets. The training target of the ratio branch $ratio^{*}$ is easily derived by:

$$ratio^{*} = \frac{A_{\mathrm{gt}}}{A_{\mathrm{hbr}}} \tag{9}$$

where A_{gt} and A_{hbr} denote the area of the ground truth box (i.e., the green box in Figure 4) and that of the minimum enclosing rectangle (i.e., the blue box in Figure 4), respectively.



(a) Bin Labels **(b)** Offset Targets

Figure 4. Bin labels and offset targets used in our PCR module. The boxes in green are ground truths and the boxes in blue are their minimum enclosing rectangles.

The loss function of the PCR module $\mathcal{L}_{\mathrm{PCR}}$ is given as follows:

$$\begin{aligned} \mathcal{L}_{\mathrm{PCR}} = & \frac{1}{N_{\mathrm{pos}}} \sum_{i} \mathcal{L}_{\mathrm{bin}}^{i}(\boldsymbol{v}, \boldsymbol{v}^{*}) \\ & + \frac{1}{N_{\mathrm{pos}}} \sum_{i} \mathcal{L}_{\mathrm{offset}}^{i}(o, o^{*}) \\ & + \frac{1}{N_{\mathrm{pos}}} \sum_{i} \mathcal{L}_{\mathrm{ratio}}^{i}(ratio, ratio^{*}) \end{aligned} \tag{10}$$

where N_{pos} is the number of positive samples. Note that $\boldsymbol{v}$ and $\boldsymbol{v}^*$ are vectors generated from bin indices. $\mathcal{L}^i_{\text{bin}}$, $\mathcal{L}^i_{\text{offset}}$, and $\mathcal{L}^i_{\text{ratio}}$ denote the bin, offset and ratio losses of the i-th training sample. $\mathcal{L}^i_{\text{bin}}$ is the standard cross entropy loss, supervising the classification process of bins. $\mathcal{L}^i_{\text{offset}}$ and $\mathcal{L}^i_{\text{ratio}}$ are in the form of smooth L1 loss, supervising the regression processes in the offset and ratio branches.

3.3.2. Inference

The operations in inference are almost the same as those in training. Let us denote the coordinates of the left-top and bottom-right vertices for a predicted horizontal bounding box as $(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$. The center coordinates of the bins on four sides, indicated as $(\boldsymbol{x}_{\text{t}}, \boldsymbol{y}_{\text{l}}, \boldsymbol{x}_{\text{b}}, \boldsymbol{y}_{\text{r}})$, can be calculated by Equation (6). With the predicted bin index $(v_{\text{t}}, v_{\text{l}}, v_{\text{b}}, v_{\text{r}})$ and the predicted offset $(o_{\text{t}}, o_{\text{l}}, o_{\text{b}}, o_{\text{r}})$, the coordinate parameters of a converted quadrilateral bounding box are obtained as follows:

$$\begin{cases} x_1 = \boldsymbol{x}_{\text{t}}(v_{\text{t}}) + o_{\text{t}} \times w_{\text{bin}} \\ y_1 = y_{\min} \\ x_2 = x_{\min} \\ y_2 = \boldsymbol{y}_{\text{l}}(v_{\text{l}}) + o_{\text{l}} \times h_{\text{bin}} \\ x_3 = \boldsymbol{x}_{\text{b}}(v_{\text{b}}) + o_{\text{b}} \times w_{\text{bin}} \\ y_3 = y_{\max} \\ x_4 = x_{\max} \\ y_4 = \boldsymbol{y}_{\text{r}}(v_{\text{r}}) + o_{\text{r}} \times h_{\text{bin}} \end{cases} \tag{11}$$

(x_1, y_1), (x_2, y_2), (x_3, y_3), and (x_4, y_4) are the coordinates of interest that determine a quadrilateral bounding box used to enclose an oriented object. In particular, if a predicted ratio surpasses a preset threshold (e.g., 0.8), we directly use the predicted horizontal bounding box as the final output bounding box. This substitution only appears in the inference phase.

4. Experiments and Results

We use the term EDA to describe our method in this section for simplicity, given that we have three-aspect considerations: feature Enhancement, feature Decoupling for classification and localization, and an Appropriate bounding box regression scheme.

4.1. Datasets and Experimental Settings

We used ResNet50 [13] with FPN (abbreviated as R50-FPN) as the default backbone for feature extraction. The hyperparameters of our EDA were kept the same as those of Faster R-CNN. All of our experiments were performed on a single NVIDIA GeForce GTX 1080Ti, with a batch size of 2. The model training was optimized by stochastic gradient descent (SGD). The initial learning rate, momentum, and weight decay rate were set to 0.005, 0.9, and 0.0001, respectively. The source code was made available at https://github.com/ShichengMiao16/EDA (accessed on 4 March 2023).

We conducted rigorous evaluations on the DOTA [3], DIOR-R [4], and HRSC2016 [5] datasets. DOTA contains 188,282 instances of 15 classes. DIOR-R has 192518 instances divided into 20 classes. In particular, HRSC2016 applies to single-class ship detection. For the DOTA and DIOR-R datasets, all the models were trained for 12 epochs and the learning rate was divided by 10 after epochs 8 and 11. For HRSC2016, the number of training epochs was 36 and the learning rate was divided by 10 after epochs 24 and 33. We cropped the images in DOTA into small patches with a size of 1024 × 1024. When it came to multi-scale training and testing, the images were resized at three scales (0.5, 1.0, 1.5) and then cropped into 1024 × 1024 patches. The image size of DIOR-R (800 × 800) remained unchanged. As for the HRSC2016 dataset, all the images were resized to a range of (800, 1333) without changing image aspect ratios. For each of the three datasets, the original training set and validation set were combined for training, while the testing set was left for testing. In particular, the reported results on DOTA were obtained from the official evaluation server.

The evaluation metrics were the Average Precision (AP) per class and the mean Average Precision (mAP) of all classes.

4.2. Ablation Studies

We first conducted some ablation studies on DOTA to validate the key components and settings of our EDA. We selected Faster R-CNN-O [6] as the baseline, which relies on an angle prediction branch in addition to Faster R-CNN to realize oriented object detection.

Table 1 evaluates the number of bins (denoted as n) in PCR. Here, we only added the PCR module to the baseline. It can be observed that dividing each side into 5 bins delivered the highest mAP, but 4 bins obtained the best balance regarding detection accuracy and model complexity. Too few bins would not suppress the regression range effectively, while too many bins would make the final performance overly dependent on the classification results. In the rest of the experiments, we fixed the number of bins to 4.

Table 1. Effect of the number of bins in PCR.

n	**1**	**2**	**3**	**4**	**5**	**6**	**7**
mAP	71.9	71.4	71.6	72.4	72.5	71.7	72.1

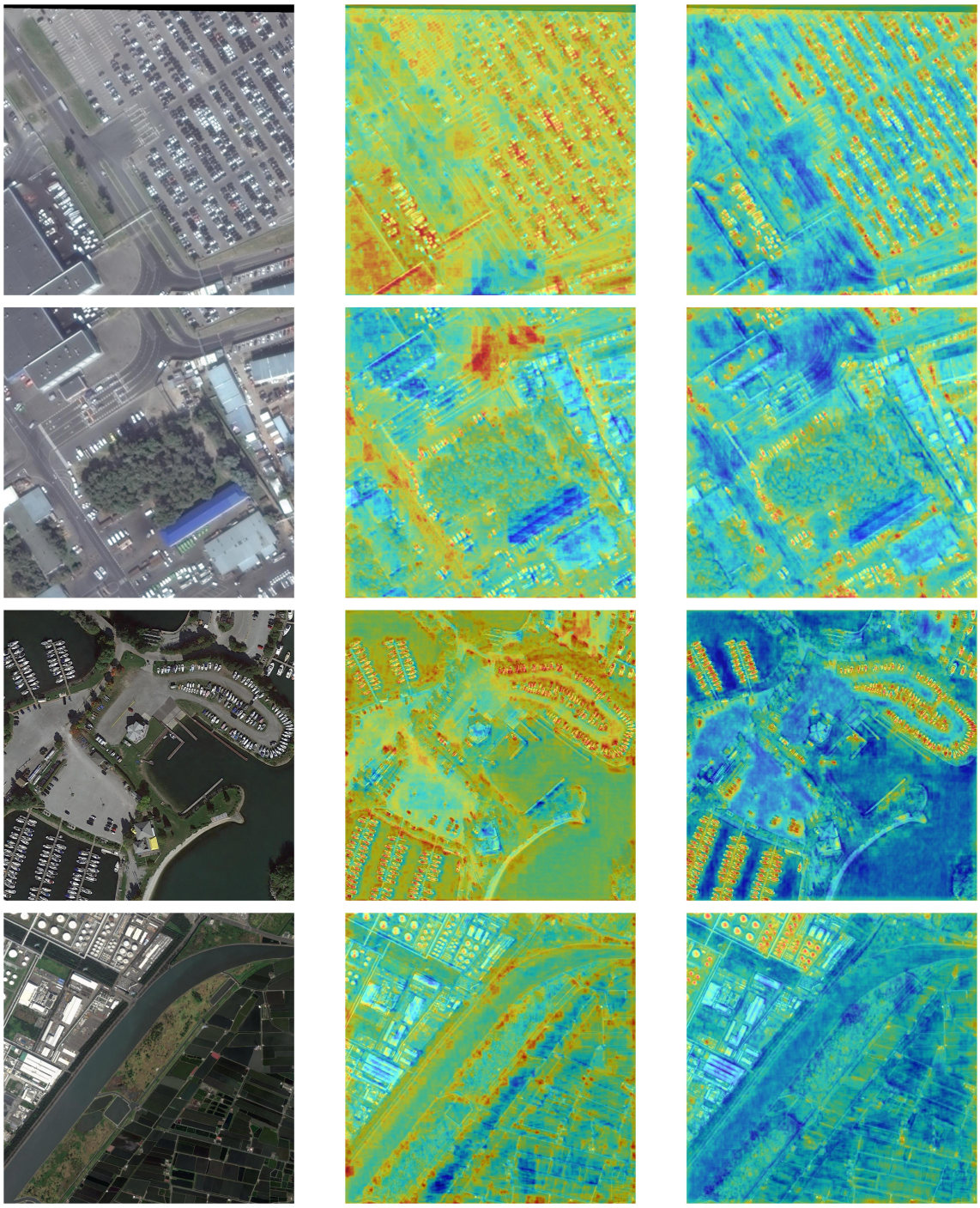
Table 2 shows the results for ablating the effectiveness of the three key modules, i.e., WFR, ATFD, and PCR. Compared with the baseline, singly applying WFR, ATFD, and PCR gave rise to 0.9%, 1.1%, and 1.0% improvements in mAP, respectively. This demonstrated that the three modules, as well as their underlying considerations, were almost equally important in regard to performance improvement. In addition, the model benefited from each pairwise combination. Pairwise combinations (i.e., WFR&ATFD, WFR&PCR, and ATFD&PCR) improved the baseline by 2.2%, 1.9%, and 1.7% in mAP, separately. All three types of pairwise combinations were superior to the separate uses. it can be seen that WFR and ATFD promoted each other as two effective tools of feature engineering for the final decision-making (both classification and localization). PCR facilitated localization by means of explicitly compressing the regression range and showed high compatibility with the above feature engineering. The combination of the three modules reported the best result (74.1% mAP), delivering noticeable performance gains over both the separate uses and the pairwise combinations. In particular, it led to 2.7% mAP gains over the baseline. In light of these results, we concluded that the three components combined with our three-aspect considerations were really effective.

Table 2. Effect of the WFR, ATFD, and PCR modules.

Method	**WFR**	**ATFD**	**PCR**	**mAP**
Faster R-CNN-O [6]				71.4
w. WFR	✓			72.3
w. ATFD		✓		72.5
w. PCR			✓	72.4
w. WFR&ATFD	✓	✓		73.6
w. WFR&PCR	✓		✓	73.3
w. ATFD&PCR		✓	✓	73.1
EDA	✓	✓	✓	74.1

Quantitatively, applying WFR led to an increase of about one point. In order to intuitively compare the feature enhancement effects, we visualized the features produced by FPN and our WFR at the first pyramidal level in Figure 5. It is obvious that the emphasized parts of our model were more concrete and accurate. The shallow features produced by FPN contained a lot of background noise for scenes with densely packed

small objects (e.g., small vehicles SV, ships SH, and storage tanks ST), as well as those with complex backgrounds (e.g., city streets and river lines). Our WFR module succeeded in feature enhancement, with task-relevant foregrounds emphasized and task-irrelevant backgrounds suppressed.



(a) Input Images **(b)** FPN Features **(c)** WFR Features

Figure 5. The features produced by FPN and our WFR at the first pyramidal level.

4.3. Comparisons with Other Methods

We conducted quantitative comparisons between our EDA and several representative detectors on the DOTA, DIOR-R, and HRSC2016 datasets. The selected comparison methods included one-stage detectors (RetinaNet-O [38], ATSS-O [39], FCOS-O [40], DAL [41],

R3Det [42], and RSDet [42]), as well as two-stage detectors (Fast R-CNN-O [6], RRPN [20], R2CNN [43], RoI Transformer [11], Gliding Vertex [23], and SCRDet[21]). In addition, the qualitative results obtained by our EDA on the three datasets were given.

Tables 3–5 list the quantitative comparisons in terms of mAP on DOTA, DIOR-R, and HRSC2016, respectively. We also list the AP for each class on the two multi-class datasets, DOTA and DIOR-R. The best results are marked in bold. Our EDA performed well on all three datasets. In particular, our EDA was clearly superior to other methods on DOTA and HRSC2016. Equipped with the backbones of ResNet50-FPN and ResNet101-FPN (i.e., ResNet101 with FPN), our EDA achieved mAPs of 74.1% and 74.3% on DOTA. Our EDA with ResNet50-FPN as the backbone largely surpassed some well-known methods equipped with ResNet101-FPN, e.g., RoI Transformer [11] and Gliding Vertex [23]. The use of multi-scale training and testing raised the mAP by about 3%, bringing the highest mAP to 77.4 % (with ResNet101-FPN as the backbone). Using ResNet50-FPN, our EDA reported 89.13% mAP on the single-class dataset HRSC2016. On the challenging DIOR-R dataset, our EDA also beat all the other methods. Gliding Vertex [23] obtained a competitive result (60.0% mAP) similar to ours (60.2% mAP). However, it lost much to our EDA on the other two datasets. Moreover, our results for separate classes (in terms of AP) on DOTA and DIOR-R were at the top level overall, without biasing towards specific easy classes. An exception would be the airport class (APO), for which an AP of just 22.6 % was reported. Though, our EDA surpassed the baseline Fast R-CNN-O [6] by 6.4 points. These results demonstrated that the classical Faster R-CNN, strengthened by feature enhancement, feature decoupling, and an appropriate bounding box regression scheme, is strong and robust in oriented object detection. Our instantiated modules generalized well. Figures 6–8 provide some visualizations on DOTA, DIOR-R, and HRSC2016, respectively. From the visualized detection results, we can observe that our EDA was able to accurately classify and locate oriented objects from various scenes, including those with densely packed small objects.

Table 3. Quantitative comparisons on DOTA. ‡ denotes multi-scale training and testing.

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
One-stage																	
RetinaNet-O [38]	R50-FPN	86.5	77.5	42.8	64.9	71.1	58.5	73.5	90.7	81.0	66.7	52.4	62.2	60.8	64.8	40.8	66.3
ATSS-O [39]	R50-FPN	89.2	73.6	48.4	64.9	74.3	77.4	79.4	**90.9**	81.5	84.9	62.7	63.8	64.7	66.5	42.6	71.0
FCOS-O [40]	R50-FPN	89.1	79.5	48.5	59.7	79.3	78.0	85.9	**90.9**	81.4	83.9	57.9	63.4	64.3	68.5	49.8	72.0
DAL [41]	R50-FPN	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	71.5
R3Det [42]	R101-FPN	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
RSDet [42]	R101-FPN	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
Two-stage																	
Faster R-CNN-O [6]	R50-FPN	88.8	80.9	44.5	68.8	78.0	65.2	82.6	90.3	82.4	83.7	58.1	61.7	56.4	69.0	60.8	71.4
RRPN [20]	R101	88.5	71.2	31.7	59.3	51.9	56.2	57.3	90.8	72.8	67.4	56.7	52.8	53.1	51.9	53.6	61.0
R2CNN [43]	R101	80.9	65.8	35.3	67.4	59.9	50.9	55.8	90.7	66.9	72.4	55.1	52.2	55.1	53.4	48.2	60.7
RoI Transformer [11]	R101-FPN	88.6	78.5	43.3	75.9	68.8	73.7	83.6	90.7	77.3	81.5	58.4	53.5	62.8	58.9	47.7	69.5
Gliding Vertex [23]	R50-FPN	88.5	82.2	51.7	68.2	77.7	72.7	86.1	90.7	85.0	85.4	57.7	66.7	64.9	66.7	48.3	72.8
SCRDet [21]	R101-FPN	**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
Ours																	
EDA	R50-FPN	89.2	83.5	51.6	69.3	77.6	74.9	**86.3**	**90.9**	85.6	85.9	59.5	64.8	68.1	66.4	57.3	74.1
EDA	R101-FPN	89.2	83.6	52.6	75.0	78.3	74.9	86.2	**90.9**	84.6	84.6	62.2	65.8	72.7	65.7	48.5	74.3
EDA ‡	R50-FPN	89.7	**85.4**	**56.6**	78.0	**79.7**	76.8	85.0	**90.9**	86.5	**88.0**	66.2	68.3	72.6	68.2	64.6	77.1
EDA ‡	R101-FPN	89.9	85.3	56.4	**78.3**	78.2	75.2	85.5	90.8	86.0	86.6	**66.8**	**69.8**	**76.9**	69.6	66.4	**77.4**

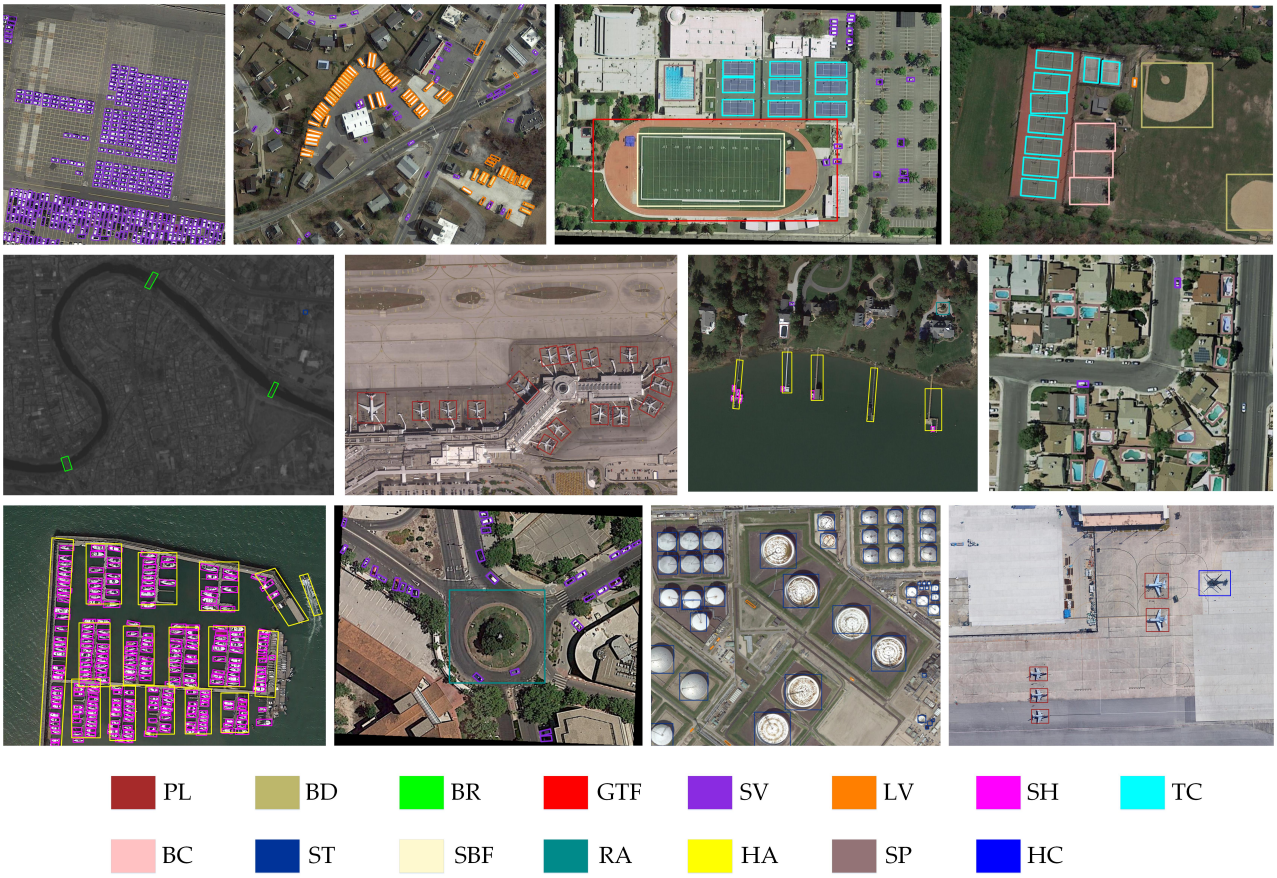


Figure 6. The visualized detection results on DOTA.

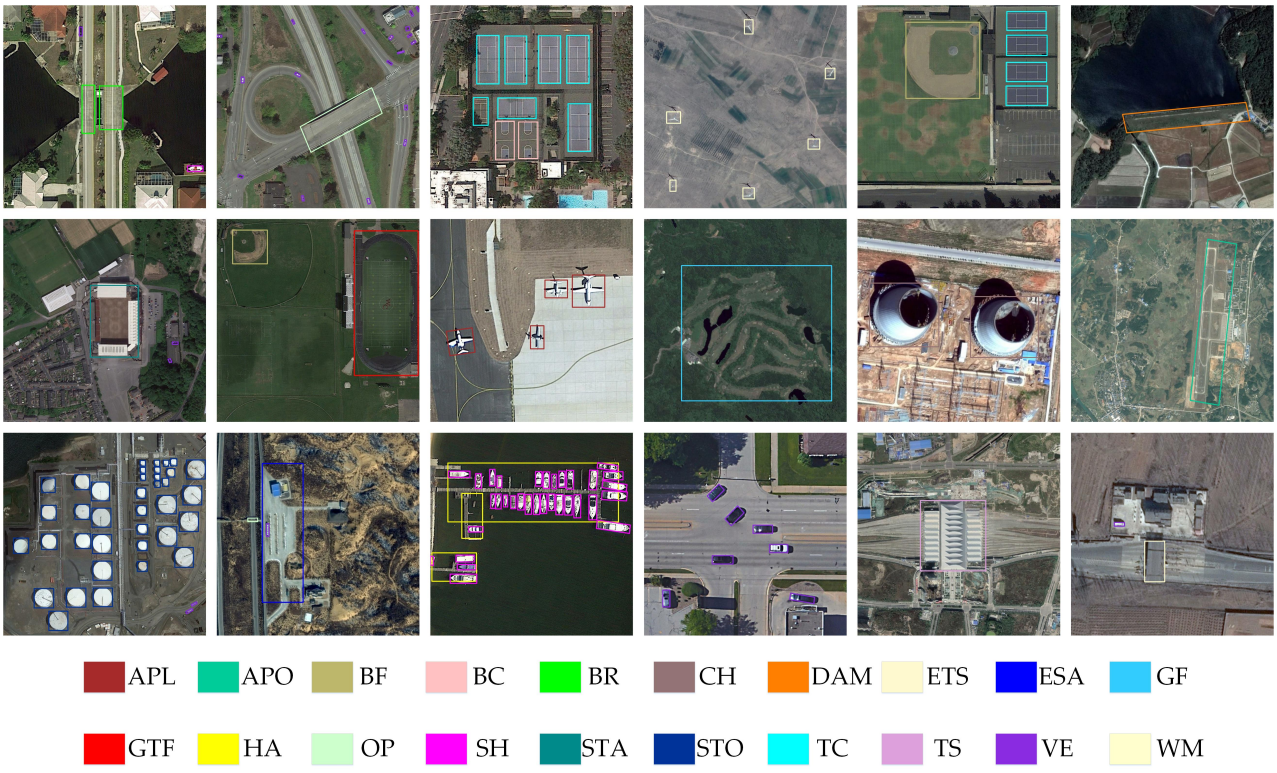


Figure 7. The visualized detection results on DIOR-R.

Table 4. Quantitative comparisons on DIOR-R.

Method	Backbone	APL	APO	BF	BC	BR	CH	DAM	ETS	ESA	GF	GTF	HA	OP	SH	STA	STO	TC	TS	VE	WM	mAP
RetinaNet-O [38]	R50-FPN	58.8	13.3	68.1	81.3	11.3	72.3	13.0	46.1	57.7	68.6	75.0	29.7	30.7	74.6	63.5	56.9	81.2	40.3	36.7	59.0	51.9
ATSS-O [39]	R50-FPN	60.2	**34.8**	**73.6**	79.3	32.2	70.9	**28.7**	59.1	75.3	69.5	78.2	34.1	51.6	77.8	70.3	59.6	79.4	50.6	39.5	62.0	59.3
FCOS-O [40]	R50-FPN	50.2	30.9	66.3	80.0	23.1	70.6	23.6	48.3	66.9	67.3	66.0	33.6	44.4	71.0	64.7	59.1	78.8	37.5	32.5	55.1	53.5
Faster R-CNN-O [6]	R50-FPN	62.5	16.2	71.7	81.3	24.2	72.5	15.2	63.6	62.6	71.6	82.1	38.0	37.2	80.4	67.0	**62.5**	**81.5**	52.5	41.4	**65.1**	57.4
Gliding Vertex [23]	R50-FPN	**62.9**	26.3	71.7	81.2	33.7	**72.6**	19.0	65.2	71.7	70.6	81.0	**41.4**	49.8	**81.0**	66.3	62.4	**81.5**	**54.3**	**42.8**	64.0	60.0
EDA (ours)	R50-FPN	62.5	22.6	71.8	81.4	34.1	72.3	18.8	66.7	75.3	70.9	**82.7**	39.2	48.9	80.9	70.6	62.4	81.4	52.9	**42.8**	64.7	60.2
EDA (ours)	R101-FPN	62.7	22.7	72.0	**81.5**	**38.2**	72.3	17.8	**68.1**	**76.7**	**73.1**	82.4	41.1	**53.1**	80.9	**72.9**	62.4	**81.5**	**54.3**	42.6	**65.1**	**61.1**

Figure 8. The visualized detection results on HRSC2016.

Table 5. Quantitative comparisons on HRSC2016.

Method	R2CNN [43]	RRPN [20]	Gliding Vertex [23]	RoI Transformer [11]	RSDet [42]	EDA (Ours)
mAP	73.07	79.08	88.20	86.20	86.50	**89.13**

5. Conclusions

We studied oriented object detection with considerations taken into feature enhancement, feature decoupling for classification and localization, and the bounding box regression scheme. We showed that the corresponding instantiated modules performed well in improving detection performance. The effectiveness of our method was validated on three well-recognized datasets for oriented object detection.

Our three-aspect considerations are general but decisive in making a strong oriented object detection method, wherein data-relevant pyramidal features, task-specific decoupled features, and a simplified regression scheme are advocated. We notice that most recent studies in oriented object detection place emphasis on exactly one or more of these to arm their detectors. We have shown that the three aspects are of equal importance for performance improvement. In this article, we limited our scope to strengthening a classical two-stage detector, i.e., Faster R-CNN. Our instantiated modules on top of it are simple in implementation but effective in performance. However, there is much room left for architectural and strategic improvements. A favorable direction beyond this easy-to-follow work is to achieve more upgraded instantiations on more efficient detectors.

Author Contributions: Conceptualization, W.C., S.M. and G.W.; methodology, W.C. and S.M.; software, S.M.; validation, W.C. and S.M.; formal analysis, W.C. and S.M.; investigation, S.M.; resources, G.C.; data curation, W.C. and S.M.; writing—original draft preparation, W.C. and S.M.; writing—review and editing, G.W. and G.C.; visualization, S.M.; supervision, G.C.; project administration, G.C.; funding acquisition, G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021B1515020072.

Data Availability Statement: The data used in this study are publicly available.

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

References

1. 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*, 21649983. [CrossRef]
2. 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.* **2023**, *61*, 22477046. [CrossRef]
3. 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 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
4. 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*, 21818767. [CrossRef]
5. 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]
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 1137–1149.
7. Lin, T.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
8. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
9. Ghiasi, G.; Lin, T.; Le, Q. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7029–7038.
10. Xie, X.; Cheng, G.; Li, Q.; Miao, S.; Li, K.; Han, J. Fewer is More: Efficient Object Detection in Large Aerial Images. *Sci. China Inf. Sci.* **2023**, 8–15. [CrossRef]
11. 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 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 2844–2853.
12. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for Object Detection. In Proceedings of the 18th IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 3500–3509.
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
14. Li, C.; Cheng, G.; Wang, G.; Zhou, P.; Han, J. Instance-Aware Distillation for Efficient Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 22595118. [CrossRef]
15. Cheng, G.; Lang, C.; Han, J. Holistic Prototype Activation for Few-Shot Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 4650–4666. [CrossRef] [PubMed]
16. Song, G.; Liu, Y.; Wang, X. Revisiting the sibling head in object detector. In Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 11563–11572.
17. Kong, T.; Sun, F.; Huang, W.; Liu, H. Deep Feature Pyramid Reconfiguration for Object Detection. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 172–188.
18. Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; Jiang, Y. Acquisition of Localization Confidence for Accurate Object Detection. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 816–832.
19. Wu, Y.; Chen, Y.; Yuan, L.; Liu, Z.; Wang, L.; Li, H.; Fu, Y. Rethinking classification and localization for object detection. In Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 10186–10195.
20. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H.; Zheng, Y.; Xue, X. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE Trans. Multimed.* **2018**, *20*, 3111–3122. [CrossRef]
21. 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 17th IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8231–8240.
22. Yang, X.; Yan, J. Arbitrary-Oriented Object Detection with Circular Smooth Label. In Proceedings of the 16th European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 677–694.
23. 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.* **2021**, *43*, 1452–1459. [CrossRef] [PubMed]

24. Qian, W.; Yang, X.; Peng, S.; Yan, J.; Guo, Y. Learning Modulated Loss for Rotated Object Detection. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 2458–2466.
25. Zhang, T.; Zhang, X.; Zhu, P.; Tang, X.; Li, C.; Jiao, L.; Zhou, H. Semantic Attention and Scale Complementary Network for Instance Segmentation in Remote Sensing Images. *IEEE Trans. Cybern.* **2022**, *52*, 10999–11013. [CrossRef] [PubMed]
26. Xiong, S.; Tan, Y.; Li, Y.; Wen, C.; Yan, P. Subtask Attention Based Object Detection in Remote Sensing Images. *Remote Sens.* **2021**, *13*, 1925. [CrossRef]
27. Cheng, G.; Lai, P.; Gao, D.; Han, J. Class Attention Network for Image Recognition. *Sci. China Inf. Sci.* **2023**, *66*, 132105. [CrossRef]
28. Cheng, G.; Wang, G.; Han, J. ISNet: Towards Improving Separability for Remote Sensing Image Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 21762431. [CrossRef]
29. Miao, S.; Cheng, G.; Li, Q.; Pei, L. Precise Vertex Regression and Feature Decoupling for Oriented Object Detection. In Proceedings of the 2022 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Kuala Lumpur, Malaysia, 17–22 July 2022; pp. 3111–3114.
30. Heidler, K.; Mou, L.; Baumhoer, C.; Dietz, A.; Zhu, X.X. HED-UNet: Combined Segmentation and Edge Detection for Monitoring the Antarctic Coastline. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
31. Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Chen, Y.; Cai, L.; Ling, H. M2Det: A Single-Shot Object Detector Based on Multi-Level Feature Pyramid Network. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 9259–9266.
32. 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. In Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NIPS), Virtual, 6–14 December 2021; pp. 18381–18394.
33. 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.* **2023**, *45*, 2384–2399. [CrossRef] [PubMed]
34. 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 38th International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; pp. 11830–11841.
35. Yang, X.; Zhang, G.; Yang, X.; Zhou, Y.; Wang, W.; Tang, J.; He, T.; Yan, J. Detecting Rotated Objects as Gaussian Distributions and Its 3-D Generalization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 4335–4354. [CrossRef] [PubMed]
36. Zhang, T.; Zhang, X.; Zhu, P.; Chen, P.; Tang, X.; Li, C.; Jiao, L. Foreground Refinement Network for Rotated Object Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5610013. [CrossRef]
37. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 2017–2025.
38. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the 16th IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007.
39. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 9759–9768.
40. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the 17th IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9626–9635.
41. Ming, Q.; Zhou, Z.; Miao, L.; Zhang, H.; Li, L. Dynamic Anchor Learning for Arbitrary-Oriented Object Detection. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 2355–2363.
42. Yang, X.; Yan, J.; Feng, Z.; He, T. R3det: Refined single-stage detector with feature refinement for rotating object. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; pp. 3163–3171.
43. 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.