remote sensing

MDPI

Article

SA3Det: Detecting Rotated Objects via Pixel-Level Attention and Adaptive Labels Assignment

Wenyong Wang [1], Yuanzheng Cai [2,*], Zhiming Luo [3], Wei Liu [4], Tao Wang [2] and Zuoyong Li [2]

[1] The College of Computer and Big Data, Fuzhou University, Fuzhou 350108, China
[2] Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, School of Computer and Big Data, Minjiang University, Fuzhou 350121, China
[3] The Department of Artificial Intelligence, Xiamen University, Xiamen 361005, China
[4] The School of Software, East China Jiaotong University, Nanchang 330013, China
* Correspondence: yuanzheng_cai@mju.edu.cn

Abstract: Remote sensing of rotated objects often encounters numerous small and dense objects. To tackle small-object neglect and inaccurate angle predictions in elongated objects, we propose SA3Det, a novel method employing Pixel-Level Attention and Adaptive Labels Assignment. First, we introduce a self-attention module that learns dense pixel-level relations between features extracted by the backbone and neck, effectively preserving and exploring the spatial relationships of potential small objects. We then introduce an adaptive label assignment strategy that refines proposals by assigning labels based on loss, enhancing sample selection during training. Additionally, we designed an angle-sensitive module that enhances angle prediction by learning rotational feature maps and incorporating multi-angle features. These modules significantly enhance detection accuracy and yield high-quality region proposals. Our approach was validated by experiments on the DOTA and HRSC2016 datasets, demonstrating that SA3Det achieves mAPs of 76.31% and 89.4%, respectively.

Keywords: remote-sensing detection; self-attention; adaptive label assignment strategy; rotation object detection

Citation: Wang, W.; Cai, Y.; Luo, Z.; Liu, W.; Wang, T.; Li, Z. SA3Det: Detecting Rotated Objects via Pixel-Level Attention and Adaptive Labels Assignment. *Remote Sens.* **2024**, *16*, 2496. https://doi.org/10.3390/rs16132496

Academic Editors: Gonzalo Pajares Martinsanz, Andrzej Stateczny, Mingyang Zhang and Hao Li

Received: 13 May 2024
Revised: 20 June 2024
Accepted: 4 July 2024
Published: 8 July 2024

1. Introduction

Rotated bounding-box detectors for remote-sensing object detection have gained attention owing to the triumph of deep convolutional neural networks (CNN). Instead of simply drawing horizontal boxes around detected objects, it detects objects by generating directional bounding boxes. A considerable amount of research [1–3] has focused on improving the representation of directional bounding boxes for remote-sensing object detection. Transfers from axis-aligned detectors have been developed, such as RoI Transformer [4], Oriented RCNN [5], and SCRDet [6]. Moreover, frameworks without anchor frames, such as RepPoint [7] and DETR [8], have also been employed. Furthermore, researchers have proposed various loss functions like GWD [9], KLD [10], and KFIoU [11] to enhance the performance of these methods. Overall, significant progress has been made in improving the representation of directional bounding boxes for remote-sensing object detection through the development of specialized detection frameworks and optimized loss functions.

Despite the advances in remote-sensing technology, aerial images are primarily captured from a bird's-eye view, which can pose challenges in identifying tiny objects based on their appearance alone. Although many efforts have been made to tackle this issue, our analysis of mainstream remote-sensing datasets (DOTA [1], HRSC2016 [12]) has revealed two areas in the current frameworks that can be further optimized: (1) Object detectors in remote-sensing images often lack robust contextual understanding, which can lead to potential misclassifications and missed detections, resulting in decreased recall rates. For instance, as illustrated in Figure 1, the object is prone to being overlooked or incorrectly

detected. Therefore, we propose leveraging comprehensive contextual information to enhance the precision of object detection in remotely sensed imagery. (2) Decoupling rotation angles has been proven to enhance the performance of various object detection techniques. For example, recent studies, such as S^2A-Net [13], and PP-YOLOE-R [14], have achieved promising results by adopting this approach. However, existing methods for decoupling angles still rely on the same feature map, which limits the significance of angle decoupling to only avoid interference with localization regression. As depicted in Figure 1b, the estimated angle may lack precision.

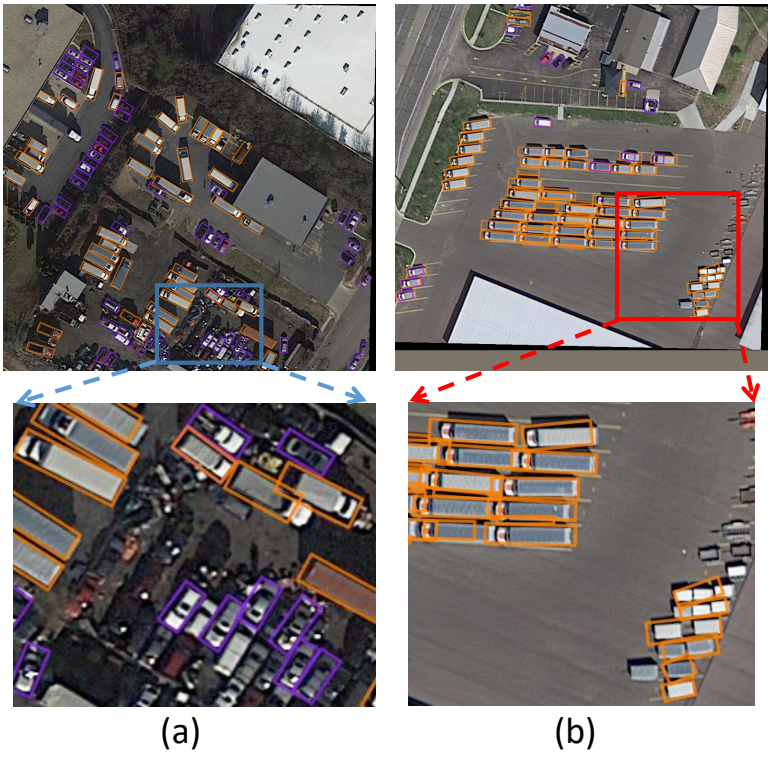

Figure 1. Visualization of some challenges encountered by existing detection models, such as mission detection, tiny object detection, and inaccurate bounding-box prediction. (**a**) The blue box in the image represents objects lost in dense small-object detection. (**b**) The red box in the image shows inaccurate detection angles and error classification.

To address the aforementioned situations, SA3Det is proposed to achieve accurate object detection in remote-sensing images. Specifically, a PSA (Pixel-level Self-Attention) module is elaborated, which efficiently detects objects and their extensive background information by guiding the attention weighting mechanism to assign higher weights to relevant elements. Furthermore, an ALA (Adaptive Label Assignment) strategy is presented that decouples the label assignment of the classification branch and the regression branch and utilizes the loss value to allocate labels. This strategy reduces the influence of the classification branch on the regression branch and enables adaptive label assignment. Additionally, an ASM (Angle-Sensitive Module) is designed to optimize angular representation and enable more precise angle prediction. It employs rotating filters to extract angle-sensitive features and introduces phase-shift encoding to overcome the problem of angle periodicity, therefore generating accurate angle information.

With enhancements to the PSA module, adaptive label assignment of ALA strategy, and new feature angle prediction of the ASM module, SA3Det achieves excellent detection performance on two popular datasets. Figure 2 illustrates the proposed framework, and the main contributions of this article are summarized as follows:

- Based on the self-attention module, we formulate a PSA module that leverages inter-pixel relationships to guide the feature map, which effectively preserves potential useful contextual details for small objects and improves the recall rate of object detection.
- Respecting the diverse learning difficulties of samples, we have implemented an ALA strategy to allocate labels based on loss values, therefore automatically selecting more valuable samples. Additionally, we have decoupled the labels of the two branches to improve the regression accuracy.
- We design an ASM module that generates an independent angle-sensitive feature map for more accurate bounding boxes regression while subtly addressing the issues caused by angle information.

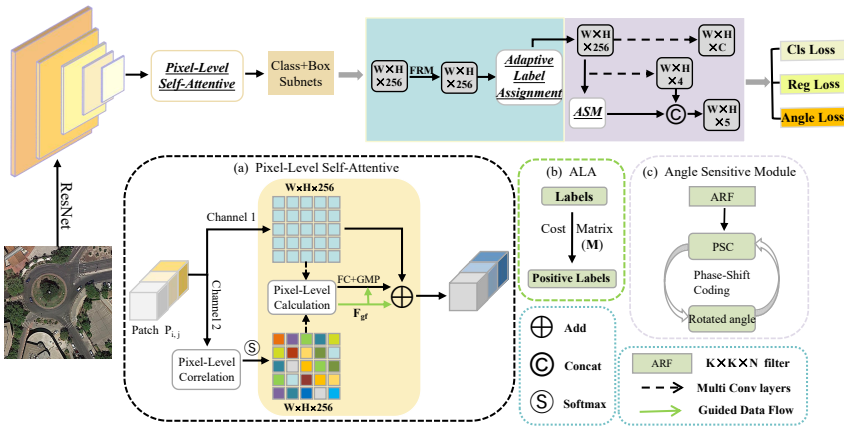

Figure 2. Architecture of the proposed SA3Det. SA3Det is composed of a backbone network, a Feature Pyramid Network [15], and a rotating object detection head. The PSA(a) acts as a bridge between FPN and the detection head. ALA(b) is responsible for using the cost matrix to divide positive and negative labels. Through ASM(c), we obtain an independent rotation feature map, which enables us to accurately determine the rotation angle.

2. Related Works

2.1. Rotated Object Detection

According to the detection strategy adopted, the current deep learning methods used for remote-sensing image object detection can be divided into anchor-based, anchor-free, and Transformer-based methods. Anchor-based methods can be further divided into single-stage regression methods and two-stage region proposal methods.

Single-stage methods, such as SSD [16] and YOLO series [14,17–20], Directly output the target position and category from the features extracted from the network, bypassing the region proposal stage. These methods provide high detection speeds suitable for real-time applications but typically exhibit lower accuracy and higher miss rates. Recently, Yang et al., proposed SCRDet++ [21], which introduces instance-level denoising technology to enhance the detection effect of small targets in aerial remote-sensing images.

The two-stage method represented by the RCNN series [4,5,22] is divided into two stages: first, extracting regions of interest (ROIs), and then classifying and regressing these regions. The effectiveness of these methods depends on the rotation anchor generation strategy to accurately cover the target's bounding box and azimuth. Methods such as RRPN [23], ReDet [24], and Oriented-RCNN [5] combine directional information by adding angle parameters to bounding-box representations and using sample loss methods for regression. In addition, LSKNet [25] proposes a detection method based on multi-scale feature pyramids, which achieves high-precision detection by combining features of different scales and is particularly suitable for multi-scale object detection tasks in remote-sensing images.

The anchor-free method was pioneered by CornerNet [26], which identifies key points in the upper left and lower right corners and assembles targets based on their similarity to

detect them. Accurate detection is achieved by regressing boundary perception vectors to capture rotated bounding boxes. DETR [27] introduces a Transformer-based method that integrates Transformer into object detection, combining CNN and Transformer components without the need for post-processing. Although DETR achieves high accuracy, it consumes many resources and converges slowly. In order to reduce the computational requirements of the DETR self-attention module, methods such as AO2-DETR [28] have been developed. By providing improved initial weights for target queries and reference points, the number of decoder layers has been significantly reduced, therefore maintaining acceptable accuracy while reducing computational complexity.

2.2. Attention Mechanism

Attention mechanisms used in computer vision can be categorized into three types: channel domain, spatial domain, and mixed domain. The main idea behind attention mechanisms is to assign different weights to different channels or spatial regions, allowing the network to focus on extracting more important information instead of treating all positions or channels equally during convolution/pooling operations. To achieve this, the channel attention SE block [29] utilizes global average pooling operations to assess the significance of various channels. Spatial attention modules like GENet [30] and SGE [31] enhance the network's ability to comprehend contextual information by incorporating spatial masks. CBAM [32] integrates both channel and spatial attention to leverage their respective advantages. Furthermore, Non-local Neural Networks [33] perform precise correlation modeling by calculating correlations between different positions in the spatial domain and learning channel weights in the channel domain.

Our approach shares similarities with Non-local NN [33], but there are two important distinctions. First, our proposed pixel-level mechanism explicitly models the correlation between each pixel position in the input image, allowing us to capture local details and texture information more effectively. Second, unlike Non-local, which aggregates information in the channel domain, we aggregate information in the spatial domain. This design is more intuitive and effective for remote-sensing tasks because channel selection cannot adequately capture the spatial variance of different targets in the image space.

2.3. Label Assignment

Label assignment in object detection can be categorized into two strategies: static and dynamic. In dynamic label assignment, the model's output is used to select positive and negative samples during training. On the other hand, static label assignment relies on predefined rules and ground conditions to determine positive and negative samples. A common practice is to allocate anchor points with an Intersection over Union (IoU) and ground truth greater than a certain threshold [15,34]. To enhance label assignment in object detectors, researchers have explored different matching methods. For example, G-Rep [35] uses normalized Gaussian distribution distance as an allocation indicator instead of IoU. ATSS [36] adapts anchor allocation based on statistics like the average and standard deviation of IoU values on a set of anchors from each ground truth. To adaptively separate anchors based on the model's learning state, PAA [37] proposes a probabilistic approach for label assignment. DAL [38] introduces a prior matching degree that considers spatial matching and feature alignment capabilities to dynamically select positive samples.

3. Methods

3.1. Overview of SA3Det

The overall network architecture is constructed based on the popular structures R^3Det [39], as depicted in Figure 2. SA3Det takes an image as input and predicts the positions of objects in the form of oriented bounding boxes (x, y, w, h, θ). Initially, ResNet and FPN are used as the backbone to extract compact multi-scale feature maps from the given image. The image features from the backbone are then enhanced by pixel-level

self-attention in each multi-scale feature map. This helps mitigate the problem of blurred object boundaries and missed detection.

Then, the feature maps are received by the classification and regression subnets. Subsequently, a feature reconstruction module (FRM) is employed to address the issue of regional feature misalignment, as described in [39]. The ALA module utilizes a cost matrix to dynamically select positive labels for classification and regression. Next, the ASM module is used to generate rotation sensing features independently, which are utilized to produce independent rotation angles. To ensure the accuracy of angle prediction during training, we incorporate an independent angle loss to provide constraints. For a detailed introduction to all three modules, please refer to Sections 3.2–3.4.

3.2. Pixel-Level Self-Attention

As depicted in Figure 3a, the fusion of features can introduce a considerable amount of noise, resulting in the loss of small target details. Several previous studies [33,40] have aimed to alleviate this issue. Therefore, a new insight is proposed to formulate a PSA module that utilizes global semantic contextual information to guide the learning of features, better preserving the relevant features of small objects while reducing noise and relatively enhancing object information. Since the feature map is continuous, non-object information cannot be eliminated. However, this allows for the retention of certain contextual information, therefore improving overall robustness.

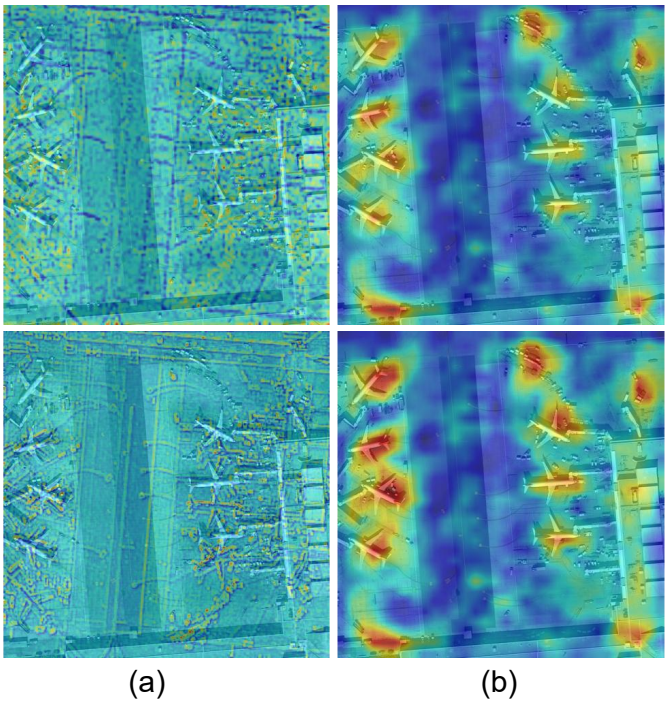

Figure 3. The visualization of image without/with PSA. The upper line indicates a feature map without PSA, while the lower line indicates a feature map with PSA. (**a**) Feature maps without/with PSA. (**b**) Attention weights without/with PSA. Red denotes higher attention values, and blue denotes lower values.

As illustrated in Figure 2, let $P_{i,j} \in \mathbb{R}^{h \times w \times b}$ represents the input module patch. In channel 1, a linear transformation is applied to acquire the feature representation, while in channel 2, the pixel-level correlation block is used to determine the attention weight of each pixel concerning the others. To begin, the patch is cropped, and a vector $X_{i,j} \in \mathbb{R}^b$ is obtained for each pixel. Subsequently, the vectors $X_{i,j}$ undergo MaxPooling and AvgPooling operations, and the resulting vectors are concatenated to create a new feature vector A_0. This step introduces position-sensitive feature representations. Next, A_0 is transposed to

yield A_1. Finally, based on these two vectors, the pixel-by-pixel attention weights can be generated as follows:

$$\begin{aligned} A_0 &= Conv_{3\times3}(MaP(X_{i,j}) + AvP(X_{i,j})), \\ A_1 &= (A_0)^T, \\ W_{pbp} &= A_0 \otimes A_1 \end{aligned} \tag{1}$$

where $W_{pbp} \in \mathbb{R}^{h\times w\times 256}$. A comprehensive illustration of this block is provided is shown in Figure 4 (left), where MaP represents MaxPooling and AvP represents AvgPooling. Through multi-layer convolution and pooling operations, pixel-level attention can capture spatial relationships and contextual information between pixels.

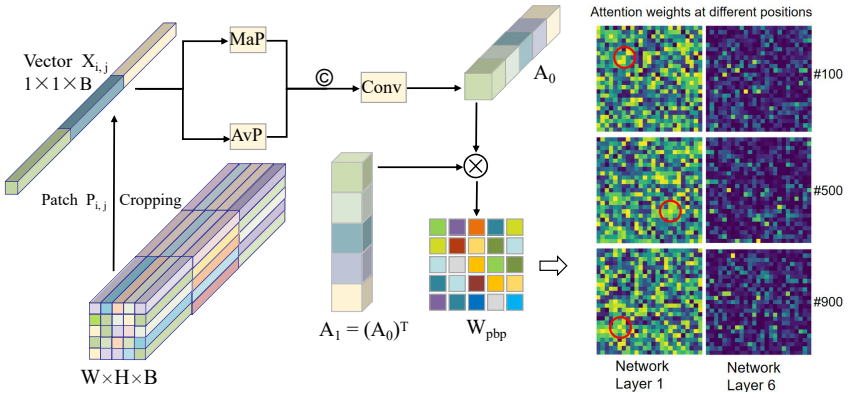

Figure 4. (**Left**) The detailed diagram of the pixel-level correlation block in PSA. It consists of two main parts: acquiring the new feature vector A_0 and obtaining the pixel correlation weight matrix W_{pbp}. Different colors represent different pixel weights. Symbol $\otimes$ denotes element-wise multiplication. (**Right**) Visualization of pixel self-attention weights at different positions and layers. The area where circles are drawn in layer 1 displays the effect of pixel self-attention, which enhances attention near key areas of objects in space.

The weight is then normalized using the SoftMax function, and the resulting matrix is added at the pixel-level by V to obtain a guided feature map F_{gf}. This process enhances object clues and facilitates guided feature map learning. The equation for this procedure is as follows:

$$F_{gf} = Conv_{3\times3}(Softmax(W_{pbp}) + V) \tag{2}$$

where V represents the feature map derived from channel 1. Then, we proceed to enhance it by multiplying it with V to obtain the enhanced attention map. Subsequently, we summarized the F_{gf}, V, and the enhanced attention map together. The equation for this step can be expressed as:

$$G_i = \sum_{i=0}^{N}(F_{gf}^i + V^i + W_{fc}^i * Gmp(V^i) * F_{gf}^i) \tag{3}$$

where W_{fc} is the weight of the fully connected layer and Gmp() is the global maximum pooling operation.

Figure 4 (Right) shows the attributes of two layers, with each matrix representing the attention weights of a single position. In layer 1, we observe that small-object details are highlighted by enhancing the weight values of key pixel positions, while local attention appears in adjacent pixels. This indicates that pixel-level attention mainly focuses on the

features of individual pixels, but it does not completely ignore contextual information. Please note that layer 6 displays a more global weight matrix without any specific interpretable local patterns. A visual comparison is presented in Figure 3, illustrating the impact of PSA on the sample features. It can be observed that using PSA results in obtaining more relevant and cleaner features.

3.3. Adaptive Label Assignment

We argue that when dealing with harder-to-learn samples, it becomes necessary to dynamically adjust the thresholds to make accurate predictions. In our baseline method trained on DOTA-v1.5, the last category (Container Crane) did not yield any predictions. However, when we attempted to train again by lowering the threshold of the baseline, the last category started appearing with predicted probabilities. Furthermore, papers such as [36,41] have demonstrated that the regression threshold does not necessarily have to be consistent with the classification threshold. It is reasonable to set a separate threshold for bounding-box regression. Some good attempts show that replacing IoU with the loss value is a more suitable approach for label assignment.

Motivated by these observations, we have designed an adaptive label assignment strategy, which adapts a clean and effective cost formula proposed in [42] to assignment labels. It decouples labels on the classification and regression branches, therefore reducing the interference of classification on regression and alleviating such situations. Given an image I, let there be Ω predictions based on the predefined anchors and $\mathcal{G}$ ground truth. For each candidate, the foreground probability $\hat{p}$ and the regressed bounding box $\mathcal{C}_g$ are output for each category. To this end, a cost matrix can be formulated as:

$$M_{i,\pi(i)} = \mathbb{1}\left[\pi(i) \in \mathcal{C}_g^i\right] \cdot \left(\hat{p}_{\pi(i)}(i)\right)^{1-\alpha} \cdot \left(\mathrm{IoU}(g_i, \mathcal{C}_g^{\pi(i)}(i))\right)^{\alpha} \tag{4}$$

where $M_{i,\pi(i)} \in [0,1]$ represents the matching quality of the $\pi(i)$-th prediction with respect to each ground truth, and $\mathcal{C}_g^i$ denotes the set of candidate predictions for i-th ground truth. $\alpha \in [0,1]$ balances the contribution between classification and regression labels, and the α defaults to 0.5. The relevant analysis of this parameter is presented in Section 4.4.

To stabilize the training process and accelerate model convergence, only candidates whose centers fall into the ground-truth boxes are considered to be potential foreground samples. During the allocation process, select the top K forecasts with the highest cost value from each FPN level. If the matching quality of the candidate exceeds the adaptive threshold calculated using ATSS (Adaptive Training Sample Selection), it is assigned as a foreground sample. Algorithm 1 describes how the proposed strategy works for an input image.

3.4. Angle-Sensitive Module

Many detection frameworks, such as S^2A-Net [13] and SCRDet [6], utilize a shared feature map to predict both the enclosing frame information and angle information of the rotated bounding box. However, in Figure 1b, it is observed that the localization of the bounding box is mostly accurate during object detection, while errors tend to occur in the prediction of the angle parameter. This observation suggests that the feature maps used for predicting the bounding-box localization may not be suitable for angle prediction. Therefore, an ASM module is proposed for independent regression of the angle parameters.

As shown in Figure 5, we first extract rotation-sensitive features using active rotating filters (ARF) [43]. The ARF is a directional filter with a size $k * k * N$, where N is the number of rotations performed during the convolution process (default value is 8). Each rotation produces a directional channel, resulting in a feature map with N channels. Here, k denotes

the kernel size. The output feature map Y consists of N orientation channels, each feature map Y can be calculated as follows:

$$Y^{(i)} = \sum_{n=0}^{N-1} G_{\theta_i}^{(n)} * X^{(n)}, \theta_i = i\frac{2\pi}{N}, i = 0, \ldots, N-1 \quad (5)$$

where G_{θ_i} is the clockwise θ_i-rotated version of G, and $G_{\theta_i}^{(n)}$ and $X^{(n)}$ are the n-th directional channel of G_{θ_i} and X, respectively. By applying ARF to the convolutional layer, we can obtain information with direction-sensitive features. Additionally, learning ARF requires much fewer training examples since the parameters between the N filters are shared.

Algorithm 1: Adaptive Label Assignment. (ALA)

Input:
$\mathcal{G}$ is a set of ground-truth boxes on the image
$\mathcal{L}$ is the number of feature pyramid levels
π_i is a set of prediction boxes from the i_{th} pyramid levels
Ω is a set of all prediction boxes
k is a quite robust hyperparameter with a default value of 9
Output:
$\mathcal{P}$ is a set of positive labels

for *each ground-truth* $g \in \mathcal{G}$ **do**
  build an empty set for candidate positive labels of the ground-truth g: $\mathcal{C}_g \leftarrow \varnothing$;
  for *each level* $i \in [1, \mathcal{L}]$ **do**
    $\mathcal{U}_i \leftarrow$ select k prediction boxes from π_i whose centers are closest to the center of ground-truth g based on L2 distance;
    $\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{U}_i$;
  compute IoU between g and $\mathcal{C}_g$: $\text{IoU}(g, \mathcal{C}_g)$;
  compute foreground probability $\hat{p}$: $\text{sigmoid}(cls\_scores[:, \text{gt}\_labels])$;
  compute a cost matrix $\mathcal{M}$: Equation (4);
  compute mean of $\mathcal{M}$: $m_g = \text{Mean}(\mathcal{M})$;
  compute standard deviation of $\mathcal{M}$: $v_g = \text{Std}(\mathcal{M})$;
  compute IoU threshold for ground-truth g: $t_g = m_g + v_g$;
  for *each candidate* $c \in \mathcal{C}_g$ **do**
    if $IoU(c, g) \geq t_g$ *and center of* c *in* g **then**
      $\mathcal{P} = \mathcal{P} \cup c$;
return $\mathcal{P}$;

To fully leverage the directional information in the feature map, we incorporated the directional encoding module PSC [44] for angle recognition. The PSC module efficiently computes cosine values using three distinct phase-shift codes, effectively addressing the boundary issue associated with angle generation. Specifically, due to the fact that rectangular objects encounter boundary problems every 180 degrees of rotation, and the phase period is 360 degrees, both periods must match to effectively solve the boundary problem, thus establishing a double-layer mapping relationship. Similarly, square objects exhibit 90-degree rotational symmetry and require a quadruple mapping relationship. Therefore, the different rectangles within the light-yellow area in Figure 5 represent two unique mapping relationships. The phase difference between the three different colors in a long rectangle is $\frac{2}{3}\pi$, so it just covers the range $[0, 2\pi]$.

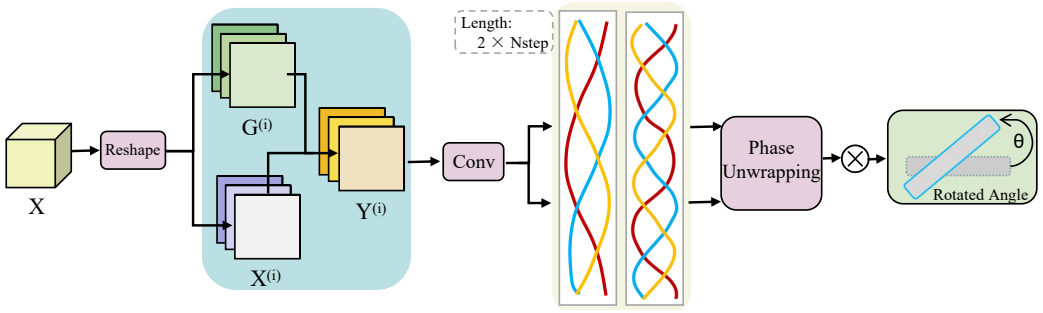

Figure 5. Illustration of the proposed ASM. The blue area represents the process of obtaining rotation sensing features. The light-yellow area indicates the use of a dual-frequency phase-shift encoder. Three different colored lines in a rectangle represent the cosine encoding of angle data from three different phases. The "Length" in the middle represents the length of the data encoding. N_{step} means the number of phase-shift steps. The final angle prediction is obtained through phase unwrapping at the end.

Subsequently, the network predicts three phase-shifted cosine values and reverts them to the angle formula using the following equation, where N_{step} defaults to 3:

$$\mathcal{F}_\theta = -\arctan \frac{\sum_{n=1}^{N_{\text{step}}} x_n \sin(\frac{2n\pi}{N_{\text{step}}})}{\sum_{n=1}^{N_{\text{step}}} x_n \cos(\frac{2n\pi}{N_{\text{step}}})} \tag{6}$$

In practice, the arctan in the formula is implemented by the arctan 2 function which limits its output to the range $(-\pi, \pi]$. x_n represents the cosine function value of different phases. It is worth noting that the angle in the formula $\mathcal{F}_\theta$ should be twice or four times the actual angle of the object.

This approach enables us to extract orientation-sensitive information from each channel of the feature map, which contains multiple orientation channels. After combining this information with the previous localization information, it is fed into the sub-network to regress the enclosing box.

3.5. Training Loss Function

In the training phase, in order to improve efficiency, we adopted a loss function similar to the one used in [39]. The difference is that, to match our independent angle prediction, we added an angle loss to constrain it. Specifically, we use Snap Loss [45] to solve the periodic problem of angular consistency during the rotation process. The angle loss can be expressed as:

$$\begin{aligned} L_{angle} &= \ell\left(\theta_{\text{pred}}, \theta_{\text{target}}\right), \\ X_{pred} &= 2 \times sigmoid(X_{feat}) - 1, \\ \theta_{\text{pred}} = \mathcal{F}_\theta(X_{pred}), &\theta_{\text{target}} = \mathcal{F}_\theta(X_{target}) \end{aligned} \tag{7}$$

where X_{feat} is the output feature of the convolution layer, X_{pred} is the predicted encoded data in the range $[-1, 1]$, and X_{target} is the ground-truth phase-shifting patterns encoded from the orientation angle of annotation boxes. The formulations of $\ell()$ can be expressed as:

$$\ell\left(\theta_{\text{pred}}, \theta_{\text{target}}\right) = \min_{k \in Z}\left(smooth_{L1}\left(\theta_{\text{pred}}, k\pi + \theta_{\text{target}}\right)\right) \tag{8}$$

After obtaining the loss function for this portion, we define the total loss as the sum of the three previously mentioned loss functions, which can be formulated as:

$$L_{total} = \omega_1 L_{cls} + \omega_2 L_{box} + \omega_3 L_{angle} \quad (9)$$

where ω_1, ω_2, and ω_3 are weights of each sub-loss function and set to 1, 0.5, and 0.2 by default.

4. Experimental Results and Analysis

4.1. Experimental Setup

We conduct experiments on three rotated object detection datasets.

DOTA [1] is one of the largest datasets used for object-oriented detection in aerial images, with two versions: DOTA-v1.0 and DOTA-v1.5. DOTA-v1.0 contains 2806 aerial images with a size range of 800 × 800 to 4000 × 4000, including 188,282 instances in 15 common categories: Plane (PL), Baseball diamond (BD), Bridge (BR), Ground track field (GTF), Small vehicle (SV), Large vehicle (LV), Ship (SH), Tennis court(TC), Basketball court (BC), Storage tank (ST), Soccer-ball field (SBF), Roundabout (RA), Harbor (HA), Swimming pool(SP), and Helicopter (HC).

DOTA-v1.5 is released with a new category, Container Crane (CC). DOTA-v1.5 contains 402,089 instances. Compared to DOTA-v1.0, DOTA-v1.5 is more challenging but remains stable in the training phase.

We use both training and validation sets for training, and the test set for testing. According to the settings in the previous method [46], We cropped the original image to 1024 × 1024 blocks in step 824. Random horizontal flipping is adopted to avoid over-fitting during training, and no other tricks are utilized. For fair comparisons with other methods, we adopt data augmentation at three scales 0.5, 1.0, 1.5. The performance of the test set is evaluated on the official DOTA evaluation server.

HRSC2016 [12] only contains one category "ship". The image size ranges from 300 × 300 to 1500 × 900. The HRSC2016 dataset contains 1061 images in total (436 for training, 181 for validation, and 444 for testing). We use both training and validation sets for training and the test set for testing. All images are resized to (800, 512) without changing the aspect ratio. Random horizontal flipping is applied during training.

Implementation details. The experiments are based on the MMRotate [46] toolbox, using libraries such as PyTorch 1.12.1, CUDA 10.2, and Python3.8. All experiments are carried out on NVIDIA RTX 2080Ti GPU cards (NVIDIA, Santa Clara, CA, USA).

In all experiments, We adopt ResNet50 and FPN (i.e., P3 to P7) as the backbone network for a fair comparison with other methods. We train all models in 12 epochs for DOTA and 36 epochs for HRSC2016. SGD optimizer is adopted with an initial learning rate of 0.01, and the learning rate is divided by 10 at each decay step. The momentum and weight decay are 0.9 and 0.0001, respectively. We use random flipping as the only data augmentation method which is also the original setting of the official MMDetection code when performing the comparison of the experiments.

4.2. Comparison to State-of-the-Art

We compare SA3Det against some state-of-the-art methods (the selected comparison method comprehensively covers popular methods, including single-stage method, two-stage method, and anchor-free method) in oriented datasets. The results are shown in Tables 1–3. The backbone used in the experiments is as follows: R-50, 101, 152 denotes ResNet-50, ResNet-101, ResNet-152, and H-104 refers to a 104-layer hourglass network.

Table 1. Comparison with state-of-the-art methods on DOTA. R101(152)-FPN stands for ResNet-101(152) with FPN and H104 stands for Hourglass-104. * Indicates multi-scale training and testing. † represents the actual experimental results of the framework.

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
Anchor-free																	
Deformable Detr [47]	R50-FPN	77.5	69.2	66.7	49.6	55.4	53.4	88.8	50.3	76.0	68.9	62.9	64.7	65.5	57.2	44.0	63.4
O2-DETR * [48]	R50-FPN	86.01	75.92	46.02	66.65	79.70	79.93	89.17	90.44	81.19	76.00	56.91	62.45	64.22	65.80	58.96	72.15
Rotated RepPoints [7]	R50-FPN	83.36	63.71	36.27	51.58	71.06	50.35	72.42	90.10	70.22	81.98	47.46	59.50	50.65	55.51	3.07	59.15
DRN * [49]	H-104	88.91	80.22	43.52	63.35	73.48	70.69	84.94	90.14	83.85	84.11	50.12	58.41	67.62	68.60	52.50	70.70
AO2-DETR [28]	R50-FPN	87.7	73.06	45.28	64.79	75.68	71.12	83.12	90.12	72.99	83.20	52.17	62.25	59.74	68.33	61.34	70.06
CFA † [50]	R50-FPN	90.20	85.36	61.92	75.17	73.61	80.13	88.53	90.10	79.09	85.04	78.93	68.86	65.83	85.31	77.24	72.91
Two-stage																	
RoI-Transformer * [4]	R101-FPN	88.64	78.52	43.44	75.92	68.81	73.68	83.59	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	69.56
SCRDet * [6]	R101-FPN	89.98	80.65	52.09	68.36	68.36	60.32	72.41	90.85	87.94	86.86	65.02	66.68	66.25	68.24	65.21	72.61
CSL * [51]	R152-FPN	90.25	85.53	54.64	75.31	70.44	73.51	77.62	90.84	86.15	86.69	69.60	68.04	73.83	71.10	68.93	76.17
ReDet [24]	ReR50-ReFPN	88.79	82.64	53.97	74.00	78.13	84.06	88.04	90.89	87.78	85.75	61.76	60.39	75.96	68.07	63.59	76.25
Gliding Vertex * [52]	R101-FPN	89.64	85.00	52.26	77.34	73.01	73.14	86.82	90.74	79.02	86.81	59.55	70.91	72.94	70.86	57.32	75.02
LSKNet [25]	R50-FPN	90.7	82.9	61.7	74.3	71.6	81.2	80.5	90.9	83.1	75.1	70.7	79.8	67.3	72.0	87.2	72.30
One-stage																	
DAL [38]	R101-FPN	88.61	79.69	46.27	70.37	65.89	76.10	78.53	90.84	79.98	78.41	58.71	62.02	69.23	71.32	60.65	71.78
S^2A-Net [13]	R50-FPN	89.11	82.84	48.37	71.11	78.11	78.39	87.25	90.83	84.90	85.64	60.36	62.60	65.26	69.13	57.94	74.12
S^2A-Net † [13]	R50-FPN	88.52	77.33	51.99	72.70	76.86	73.49	85.48	90.90	81.24	83.27	55.86	66.08	63.79	67.20	52.12	72.46
R^3Det † [39]	R50-FPN	88.65	73.92	43.83	69.10	77.05	72.56	82.39	90.88	76.98	84.02	55.66	66.92	59.98	65.10	47.69	70.32
PIoU [53]	DLA-34	80.9	69.7	24.1	60.2	38.3	64.4	64.8	90.9	77.2	70.4	46.5	37.1	57.1	61.9	64.0	60.5
O^2-DNet * [54]	H-104	89.31	82.14	47.33	61.21	71.32	74.03	78.62	90.76	82.23	81.36	60.93	60.17	58.21	66.98	61.03	71.04
SCRDet++ [21]	R152-FPN	89.20	83.36	50.92	68.17	71.61	80.23	78.53	90.83	86.09	84.04	65.93	60.86	68.83	71.31	66.24	74.41
YOLOv5m † [19]	R50-FPN	90.7	86.3	62.8	84.1	73.6	85.2	82.5	90.5	87.1	77.1	75.7	65.86	75.8	80.3	86.2	72.66
Our																	
SA3Det	R50-FPN	88.25	81.80	47.24	70.41	77.93	75.09	86.03	90.88	83.27	84.34	61.56	61.63	65.61	69.08	55.24	73.22
SA3Det	R101-FPN	88.96	81.15	49.49	74.59	79.84	80.38	86.88	90.88	78.46	85.22	61.98	70.32	67.95	70.70	53.22	74.67
SA3Det	R152-FPN	89.26	84.04	51.38	73.31	80.24	81.64	87.46	90.88	85.91	85.87	63.01	70.34	72.61	71.75	57.00	76.31
SA3Det *	R101-FPN	88.71	85.16	55.83	79.62	80.07	82.79	88.32	90.88	85.68	87.53	67.80	71.87	76.01	77.89	67.95	79.07
SA3Det *	R152-FPN	88.89	82.99	56.09	79.44	80.66	83.32	88.38	90.85	85.93	87.58	66.81	73.79	75.53	78.70	69.52	79.23

Results on DOTA. Table 1 shows a comparison of our SA3Det with the recently state-of-the-art detectors on the DOTA-v1.0 dataset with respect to oriented bounding-box detection. Among these methods, Redet and CSL are implemented by adding angle prediction channels in the bounding-box regression branch of the classical computer vision algorithms Faster-RCNN [34] and RetinaNet [55], respectively. Other methods are especially proposed to detect rotating objects in remote-sensing images. SCRDet++ [21] introduces the idea of denoising to object detection. Instance-level denoising on the feature map is performed to enhance the detection of small and cluttered objects. DAL [38] is a dynamic anchor learning method that uses a new matching mechanism to evaluate anchors and assign them more efficient labels. S^2A-Net [13] uses a new alignment convolution, which can adaptively align convolution features according to anchors. CFA [50] proposes a convex hull representation method that can more accurately locate the range of objects while reducing feature aliasing to some extent. LSKNet [25] dynamically adjusts the receptive field of targets through a series of Depth wise convolution kernels and spatial selection mechanisms, allowing the model to adapt to target detection in different backgrounds. YOLOv5m [19] is a model in the YOLOv5 series, and a rotation detection version of this model has already appeared in the field of remote sensing.

Unlike comparison methods, our method proposes a new pixel-level attention mechanism and independent angle regression branches to enhance the network's regression and directional feature extraction, therefore improving the detection ability of rotating objects. For the accuracy measured by mAP, we achieved 76.31% mAP with single-scale data and 79.23% mAP with multi-scale data. Specifically, SA3Det outperforms RoI-Transformer 5.11% (74.67% vs. 69.56%), better than R^3Det 2.9% (73.22% vs. 70.32%), SCRDet 2.06% (74.67% vs. 72.61%), O^2-DNet 3.63% (74.67% vs. 71.04%), CFA 0.31% (73.22% vs. 72.91%), LSKNet 0.92% (73.22% vs. 72.30%), YOLOv5m 0.56% (73.22% vs. 72.66%), which is a great improvement. It is worth noting that our results have a good lead in the detection of GTF, RA, and SP. The directionality of these classes of objects is obvious, indicating that our detector has a strong ability in direction detection.

We further conduct the experiments by setting the backbone of all models to ResNet50 to investigate the effects of the backbone. From Table 1, It can be observed that our SA3Det achieves the best result in comparison to all anchor-free methods. SA3Det achieves 73.22% mAP, about 2.52% mAP higher than the second-best method DRN*. Compared with the anchor-based methods, our method is better than most single-stage methods and two-stage methods, even though many of them use ResNet101, which contributes to a stronger backbone. The results show that our model performs slightly worse than S^2A-Net by 0.9%. Although our method does not achieve the best performance, the proposed method has some apparent advantages over the anchor-based methods. When detecting objects with dense distribution and large-scale differences, our SA3Det generates fewer error angles and a lower probability of missed detections. Partial visualization results are shown in Figure 6.

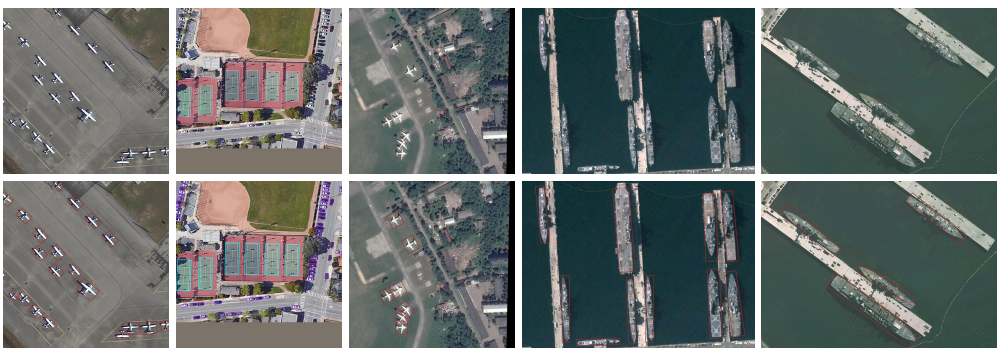

Figure 6. Visualization of SA3Det detection. The first three images and the last two images, respectively, show the images used for DOTA and HRSC detection. The second row shows the corresponding detection results obtained by SA3Det.

Results on DOTA-v1.5. Compared to DOTA-v1.0, DOTA-v1.5 contains more tiny objects. We summarize the results for DOTA-v1.5 in Table 2. Compared with state-of-the-art methods, SA3Det achieves 67.18% mAP with single-scale data and 76.02% mAP with multi-scale data, outperforming Mask RCNN [56], AO2-DETR [28], and HTC [57]. The experiments verify that our proposed SA3Det can achieve superior performance in small-object detection.

Results on HRSC2016. For the HRSC2016 dataset, some of them have large aspect ratios and various orientations. In Table 3, it can be seen that our SA3Det achieves good performance. Among these methods, R2CNN [58] and RRPN [23] are proposed in the field of computer vision to detect slanted text with angles. Other methods are proposed to detect rotated objects in RSIs. It is worth noting that we used the PASCAL VOC 2007 metric to calculate the mAP of the detection results (as we did not find the dataset for the 2012 metric), and the mAP of other methods compared was also calculated under this metric. Specifically, SA3Det achieved 88.5% and 89.4% mAP using R101 and R152, respectively, under VOC 2007. Partial visualization results are shown in Figure 6. From it, it can be seen that although some ships have the characteristics of large-scale differences and dense arrangement, SA3Det can always provide appropriate OBB (Oriented Box Boundary) to tightly surround ships in any direction. Even in different environments such as ports, coasts, and seas, this method can still perform high-quality detection.

Table 2. Comparison with state-of-the-art methods on DOTA-v1.5. R50-FPN stands for ResNet-50 with FPN, and H104 stands for Hourglass-104. * Indicates multi-scale training and testing.

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	CC	mAP
Mask RCNN [56]	R50-FPN	76.84	73.51	49.90	57.80	51.31	71.34	79.75	90.46	74.21	66.07	46.21	70.61	63.07	64.46	57.81	9.42	62.67
HTC [57]	R50-FPN	77.80	73.67	51.40	63.99	51.54	73.31	80.31	90.48	75.12	67.34	48.51	70.63	64.84	64.48	55.87	5.15	63.40
AO2-DETR [28]	R50-FPN	79.55	78.14	42.41	61.23	55.34	74.50	79.57	90.64	74.76	77.58	53.56	66.91	58.56	73.11	69.64	24.71	66.26
AO2-DETR * [28]	R50-FPN	87.13	85.43	65.87	74.69	77.46	84.13	86.19	90.23	81.14	86.56	56.04	70.48	75.47	78.30	72.66	42.62	75.89
ReDet * [24]	ReR50-ReFPN	88.51	86.45	61.23	81.20	67.60	83.65	90.00	90.86	84.30	75.33	71.49	72.06	78.32	74.73	76.10	46.98	76.80
Point RCNN * [59]	ReR50-ReFPN	83.40	86.59	60.76	80.25	79.92	83.37	90.04	90.86	87.45	84.50	72.79	77.32	78.29	77.48	78.92	47.97	78.74
Our																		
SA3Det	R50-FPN	78.07	84.33	48.04	70.30	55.80	75.52	80.54	90.86	78.04	74.67	50.63	69.96	68.12	65.63	60.02	15.59	67.18
SA3Det *	R50-FPN	85.48	86.39	59.82	76.30	69.13	81.49	89.15	90.86	83.05	84.28	65.21	74.43	78.92	75.33	69.90	40.57	76.02

Table 3. Accuracy and speed on HRSC2016. And 07 (12) means using the 2007 (2012) evaluation metric.

Method	Backbone	Image Size	mAP (07)	mAP (12)	Speed
R^2CNN [58]	R101-FPN	800 × 800	73.07	79.73	5 fps
RoI-Transformer [4]	R101-FPN	512 × 800	86.20	-	6 fps
Gliding Vertex [52]	R101-FPN	-	88.20	-	-
DRN [49]	H-104	-	-	92.70	-
R^3Det [39]	R101-FPN	800 × 800	86.9	-	-
CSL [51]	R152-FPN	-	89.62	-	-
RRPN [23]	R101-FPN	800 × 800	79.08	85.64	1.5 fps
RRD [43]	VGG16	384 × 384	84.3	-	-
CenterMap-Net [60]	R50-FPN	-	-	92.8	-
Our					
SA3Det	R50-FPN	800 × 800	85.6	-	-
SA3Det	R101-FPN	800 × 800	88.5	-	-
SA3Det	R152-FPN	800 × 800	89.4	-	-

4.3. Ablation Studies

In this section, we conduct a series of experiments on the testing set to validate the effectiveness of our method. To further understand the effectiveness of our proposed method, we further explore and validate the contributions of different modules of the proposed SA3Det framework, i.e., the PSA module, the ALA module, and the ASM module. We conducted ablation experiments on the DOTA and HRSC2016 datasets, and the results are shown in Tables 4 and 5, respectively.

As shown in Table 4, in most categories, adding any module can improve the accuracy of detection, and the combination of the three modules is the best, with a mAP of 73.22%. This indicates that PSA retains more detailed features of small targets, the ALA module adaptively divides positive and negative sample labels, and ASM independently predicts angles. All three methods are effective. On the HRSC dataset, as shown in Table 5, our module has also improved accuracy. Figure 7 is a specific visualization of our three innovative methods, showing the problems we encountered in the baseline and the results we achieved after solving the corresponding problems.

In the ablation study of different losses, we classified the losses using Focal Loss and focused on the regression losses of the bounding box in the baseline. We compared several commonly used losses, as shown in Table 6. Specifically, our loss achieves a 1.39% gain in mAP relative to the KFIoU, 2.05% gain relative to the KLD, 4.97% gain relative to the GWD, 2.9% gain relative to the Smooth-L1, and 3.67% gain relative to the L1 loss. Our loss achieves a significant improvement in performance, demonstrating the effectiveness of angle constraints.

Table 4. The ablation of the modules presented in SA3Det was studied in this experiment using DOTA-v1.0. The striking results show the best performance. Both baseline and SA3Det use ResNet50 as the backbone. ✓indicates that the module is included in the model.

	PSA	ALA	ASM	PL	BD	BR	GTF	SV	LV	SBF	mAP
Baseline				88.65	73.92	43.83	69.10	77.05	72.56	55.66	70.32
SA3Det	✓			89.20	77.30	44.81	68.89	77.26	73.96	54.49	71.24 (+0.92)
SA3Det		✓		88.01	80.62	44.30	68.19	77.40	74.65	56.93	71.07 (+0.75)
SA3Det	✓	✓		89.04	77.77	44.81	69.03	77.47	76.06	53.78	71.72 (+1.40)
SA3Det	✓		✓	87.96	79.64	46.75	72.82	77.47	75.52	59.99	72.31 (+1.99)
SA3Det	✓	✓	✓	88.25	81.80	47.24	70.41	77.93	75.09	61.56	73.22 (+2.90)

Table 5. Ablation study of the module presented in SA3Det. This experiment was performed using HRSC2016. The striking results show the best performance. Both baseline and SA3Det use ResNet152 as the backbone. ✓indicates that the module is included in the model.

	PSA	ALA	ASM	RECALL	mAP
Baseline				90.2	86.9
SA3Det	✓			93.0	87.4 (+0.5)
SA3Det		✓		92.5	88.0 (+1.1)
SA3Det	✓	✓		92.9	88.4 (+1.5)
SA3Det	✓		✓	93.6	88.6 (+1.6)
SA3Det	✓	✓	✓	97.2	89.4 (+2.5)

Table 6. Comparison of the properties and performance of different regression losses. Baseline is R^3Det. The striking results show the best performance. Both baseline and SA3Det use ResNet50 as the backbone. ✓indicates that the method is included in the model.

	Focal Loss [15]	L1 Loss [49]	Smooth-L1 [34]	GWD [9]	KLD [10]	KFIoU [11]	Our Loss	mAP
Baseline	✓		✓					70.32
SA3Det	✓	✓						69.55
SA3Det	✓			✓				68.25
SA3Det	✓				✓			71.17
SA3Det	✓					✓		71.83
SA3Det	✓						✓	73.22

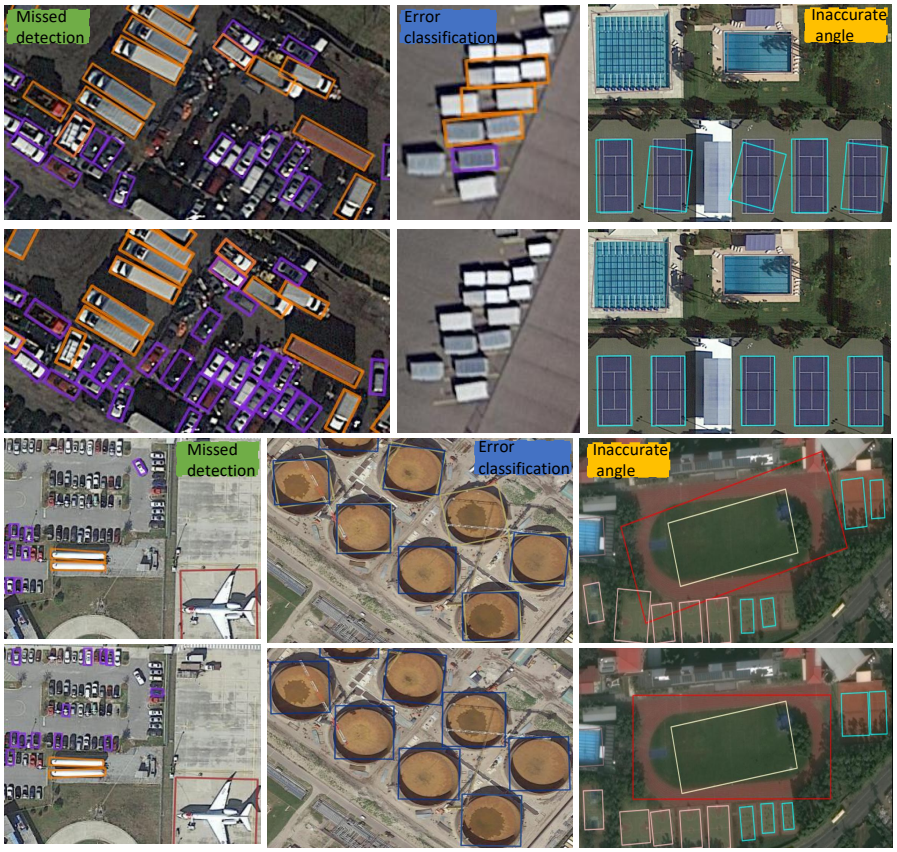

Figure 7. Visualizationof three innovative methods. The green label represents the problems encountered without the PSA module, the blue label represents the problems encountered without the ALA module, and the yellow label represents the problems encountered without the ASM module.

4.4. Parameter Analysis

Method parameter. R^3Det has high efficiency as an independent detector, but adding any module to it will introduce more computation, which may affect its efficiency. Therefore, We compared the models based on parameter counts (Params), inference speed (speed), floating-point operations per second (FLOPs), and mAP. The evaluated algorithms include RoI-Transformer, AO2-DETR, Yolov5m, S2ANet, Baseline (R3Det), and SA3Det, each evaluated using a standardized image size of 1024x1024 pixels and trained over a period of 12. All of these were evaluated under consistent conditions.

As shown in Table 7. Although SA3Det has poorer speed and Flops compared to RoI-Transformer, AO2-DETR, and S2ANet, our parameter count is lower and detection accuracy is higher. The R^3Det achieved 70.32% mAP across 37.08 M parameters, indicating that the baseline is reliable. After adding three modules, SA3Det achieved a parameter count of 37.27 M, a speed of 65.7 ms, and 232.92 GFLOPs, achieving a mAP of 73.22, indicating a good trade-off between computational efficiency and detection accuracy. In addition, Yolov5m achieved similar performance to ours with fewer model parameters, but our model produces better results in small-object detection, as shown in Section 4.5. These results indicate that our method can achieve competitive performance and a better balance of speed–accuracy, meeting the engineering needs of the real world.

The effect of ALA's parameter α. Here, we delve into the influence of the hyperparameter α within the ALA, as delineated in Table 8. When α is set to 0.5, SA3Det achieves a peak mean Average Precision (mAP) of 73.22%, indicating a notable performance enhancement. However, surpassing this threshold leads to a degradation in our method's performance. Our rationale is rooted in the prevalence of diminutive targets in remote-sensing imagery, often characterized by low Intersection over Union (IoU) values, posing challenges for precise target localization by the model. Excessive emphasis on IoU, re-

sulting from disproportionate weight allocation, exacerbates the detection constraints for small targets, therefore impinging on the efficacy of remote-sensing target detection. Thus, informed by this observation, we designate the default value of α as 0.5 to strike a harmonious balance between preserving detection rates for small targets and optimizing overall method performance.

Table 7. Research on SA3Det model parameters. This experiment used DOTA-1.0. Param represents the number of parameters for the entire network. Speed is the inference speed of each image. Flops is a floating-point operation per second.

Method	Size	Epochs	Param (M)	Speed (ms)	Flops	mAP
RoI-Transformer [4]	1024	12	55.13 M	61.5	225.29	69.56
AO2-DETR [28]	1024	12	46.95 M	63.45	236.8	70.06
Yolov5m [19]	1024	12	22.65 M	47.5	97.52	72.66
S2ANet [13]	1024	12	38.6 M	55.7	197.62	72.46
Baseline [39]	1024	12	37.08 M	60.5	232.67	70.32
SA3Det	1024	12	37.27 M	65.7	232.92	73.22

Table 8. The effect of the parameter α in the ALA.

Setting	α	mAP
I	0.1	70.93
II	0.3	72.23
III	0.5	73.22
IV	0.7	71.90
V	0.9	71.48

4.5. Qualitative Analysis

From Figure 8, it can be seen that the accuracy curve of SA3Det is more stable than Yolov5s overall, especially with a more obvious upward trend in the middle and later stages of training. For example, in remote-sensing subcategories such as small vehicles, ships, and bridges, the curve of SA3Det shows a smoother upward trend. The reason for this situation is that the pixel self-attention mechanism can make the model pay more attention to key areas in the image and suppress interference from irrelevant areas. This mechanism can more accurately extract useful features, improve the model's feature expression ability, and thus achieve more refined feature extraction and preservation of small-object details.

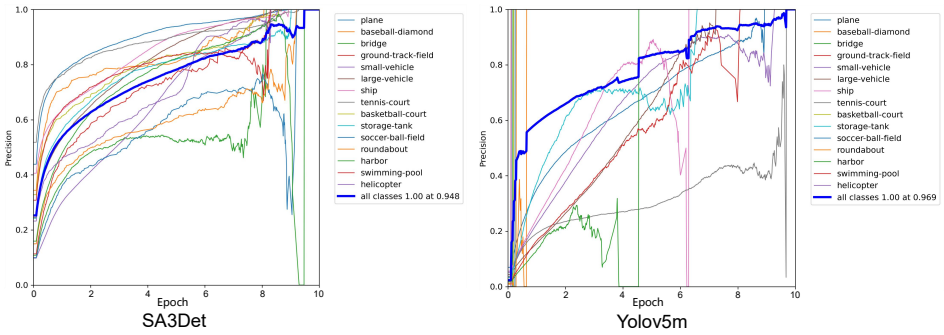

Figure 8. Qualitative comparison of detection results on DOTA using SA3Det and YOLOv5m. We only show the top 10 epochs with significant effects in the figure.

Figure 9 shows the qualitative analysis results of representative samples. It can be seen that the results of our method are very close to actual ground conditions and can accurately detect vehicles of different sizes. Other comparison methods only perform well on relatively large vehicles, while small vehicles may have missed detections. The comparison method has varying degrees of error detection and omission detection, while our method produces

identical basic facts. There may be significant differences in proportion between different types of objects. It displays the presence of objects of different sizes in the same image. All comparison methods will lose small objects. In contrast, we can perceive small objects well and detect them all. In addition, we can also see from the graph that our method is more accurate in terms of detection angle compared to other detection methods.

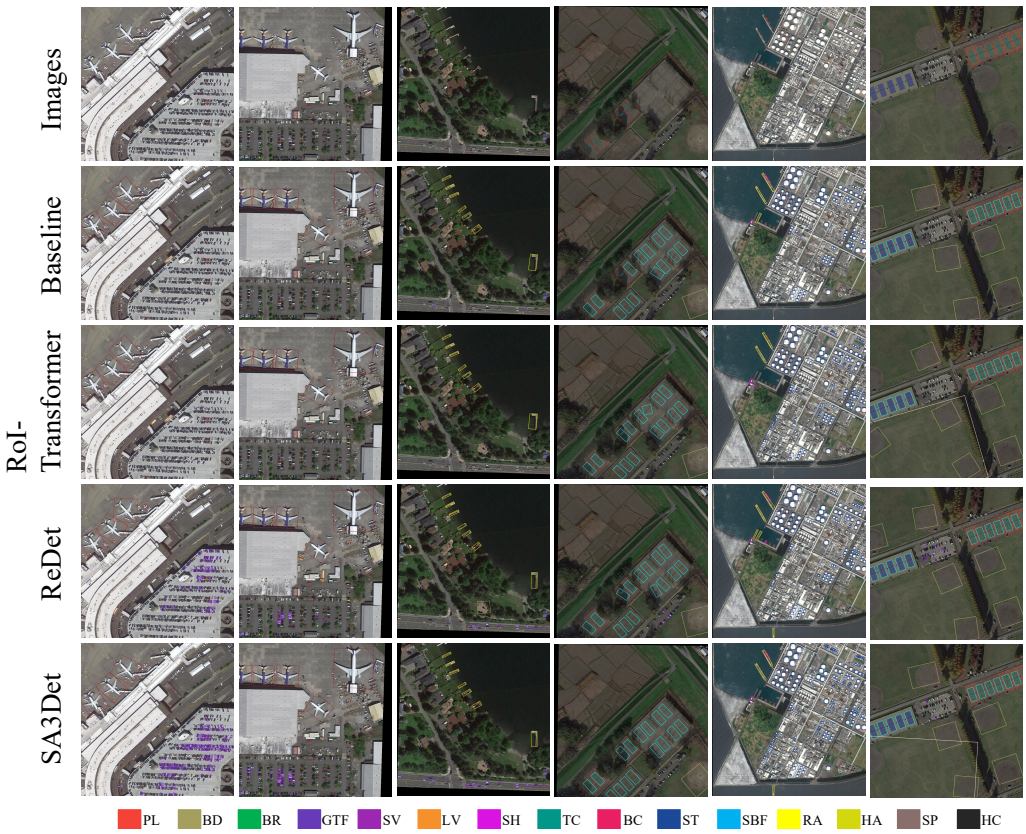

Figure 9. Qualitative comparison between proposed SA3Det and baseline, RoI-Transformer, and ReDet on DOTA. The original image, baseline, RoI-Transformer, ReDet, and our proposed SA3Det are presented from left to right in each column, respectively. The color description section below represents the detection objects corresponding to the detection boxes of different colors in the image.

5. Discussion

The proposed method demonstrates the potential of pixel-level self-attention and label optimization, offering a novel approach to designing remote-sensing object detection models. The detection of small objects poses a common challenge in object detection, not exclusive to remote-sensing imagery, mainly due to the limited pixel-level information provided by small objects, making robust feature extraction challenging. Conventionally, approaches such as image pyramids or multi-scale feature fusion are employed to address this issue, yet they often result in information loss when handling small objects. While these methods may enhance the accuracy of small-object detection to some extent, their performance tends to be suboptimal in complex backgrounds. Our research findings indicate that pixel-level self-attention plays a pivotal role in preserving and exploring fine-grained feature correlations crucially linked to the spatial relationships of potential small objects. In particular, in complex background remote-sensing images, this approach exhibits the capability to learn high-quality foreground information.

In object detection, a crucial challenge lies in balancing the learning of positive and negative samples by the network. It is commonly acknowledged that utilizing hard example mining techniques can be effective but often demands substantial computational resources and time. In contrast, adaptive label assignment (ALA) strategies allocate labels based on loss costs to refine suggestions, enabling the automatic selection and mining of more

valuable samples during training. Results demonstrate that ALA can efficiently and accurately select high-quality positive samples. While such approaches are not uncommon, by introducing loss costs as a benchmark, they essentially require minimal computational resources and time and can dynamically adjust the importance of samples during training, thus enhancing both training efficiency and detection performance of the model.

For remote-sensing detection tasks, dense objects pose challenges for bounding-box generation. Particularly for objects with high aspect ratios, subtle angular biases can lead to significant detection errors. Previous approaches have employed rotation-equivariant detectors to extract rotation-invariant features, typically in the feature extraction stage, but often result in increased computational complexity and insufficient rotational invariance capability. In addressing this issue, a primary approach involves utilizing angle encoding for improvement. Hence, our proposed angle-sensitive module implicitly learns feature maps representing rotations, enhancing the ability to predict angles by considering features predicted from multiple angles. This method effectively tackles the detection of objects with high aspect ratios, enhancing both detection accuracy and robustness, particularly demonstrating outstanding performance in remote-sensing imagery.

Limitations and Future Work. In the current landscape of deep learning for remote-sensing object detection, the application of techniques like pixel-level self-attention mechanisms has indeed achieved notable successes. However, it is imperative to acknowledge certain limitations. First, while pixel-level self-attention mechanisms excel in extracting fine-grained feature correlations, their high computational complexity constrains their applicability when dealing with large-sized, high-resolution images. This not only escalates the demand for computational resources but also hinders their widespread adoption in practical applications. Second, although angle-sensitive modules enhance a model's robustness to angular variations, they are restricted by predefined angle ranges, making them inadequate for addressing extreme angular changes. Consequently, future research directions should encompass optimizing computational efficiency, enhancing model stability and generalization capabilities, and exploring more flexible adaptive mechanisms and angle ranges to propel further advancements in deep learning for remote-sensing object detection. This entails but is not limited to the investigation of more efficient algorithms, dynamic parameter adjustments, and the exploration of more adaptable angle-sensitive module designs.

6. Conclusions

In this paper, we propose a new remote-sensing object detection network (SA3Det) to improve the detection accuracy of multi-scale targets in remote-sensing images. The proposed SA3Det consists of three new modules: PSA, ALA, and ASM. The former uses pixel-level attention and guided feature maps to provide critical information, better preservation of details, and improved accuracy for small target detection. The ALA strategy decouples labels and assigns labels based on loss, which enables the automatic selection of more valuable samples during the training phase and improves robustness. ASM generates independent rotation-sensitive features to be used to generate more accurate angles. At the same time, angle loss is added to the loss to constrain the independent angles. Through comprehensive experimentation, SA3Det has demonstrated significant performance enhancements over existing methodologies on both the DOTA and HRSC2016 datasets, marking a notable advancement in the field.

Author Contributions: Conceptualization, W.W. and Y.C.; methodology, W.W.; software, T.W.; validation, W.W., Y.C. and Z.L. (Zhiming Luo); formal analysis, Z.L. (Zuoyong Li); investigation, W.W.; resources, W.W.; data curation, W.W.; writing—original draft preparation, W.W. and Y.C; writing—review and editing, Z.L. (Zhiming Luo) and W.L.; visualization, W.W.; supervision, Y.C. and Z.L. (Zuoyong Li); project administration, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: High-level Talent Research Start-up Fund Project of Fujian University of Traditional Chinese Medicine (No. X2020005-Talent)

Data Availability Statement: The three datasets used in this study are (1) and (2) open-source datasets from the Remote-Sensing National Heavy Industry Laboratory of Wuhan University, Can be found in https://captain-whu.github.io/DOTA/dataset.html (accessed on 28 November 2017), and (3) an open-source dataset from Northwestern Polytechnical University, available at https://sites.google.com/site/hrsc2016/ (accessed on 26 June 2016).

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

References

1. 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 International Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
2. Zhang, G.; Lu, S.; Zhang, W. CAD-Net: A Context-Aware Detection Network for Objects in Remote Sensing Imagery. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 10015–10024. [CrossRef]
3. 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]
4. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning RoI Transformer for Detecting Oriented Objects in Aerial Images. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
5. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
6. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Xian, S.; Fu, K. SCRDet: Towards More Robust Detection for Small, Cluttered and Rotated Objects. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
7. Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. RepPoints: Point Set Representation for Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
8. Liu, S.; Ren, T.; Chen, J.; Zeng, Z.; Zhang, H.; Li, F.; Li, H.; Huang, J.; Su, H.; Zhu, J.; et al. Detection Transformer with Stable Matching. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023.
9. 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 Proc. of International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022.
10. 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 Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
11. Yang, X.; Zhou, Y.; Zhang, G.; Yang, J.; Wang, W.; Yan, J.; Zhang, X.; Tian, Q. The KFIoU Loss for Rotated Object Detection. In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
12. Liu, Z.; Yuan, L.; Weng, L.; Yang, Y. A high resolution optical satellite image dataset for ship recognition and some new baselines. In Proceedings of the International Conference on Pattern Recognition Applications and Methods, Porto, Portugal, 24–26 February 2017; SciTePress: Setúbal, Portugal, 2017; Volume 2, pp. 324–331.
13. Han, J.; Ding, J.; Li, J.; Xia, G.S. Align Deep Features for Oriented Object Detection. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 5602511. [CrossRef]
14. Wang, X.; Wang, G.; Dang, Q.; Liu, Y.; Hu, X.; Yu, D. PP-YOLOE-R: An Efficient Anchor-Free Rotated Object Detector. *arXiv* **2022**, arXiv:2211.02386.
15. 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, Salt Lake City, UT, USA, 18–22 June 2018.
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
17. Luo, H.; Gao, F.; Lin, H.; Ma, S.; Poor, H.V. YOLO: An Efficient Terahertz Band Integrated Sensing and Communications Scheme with Beam Squint. *IEEE Trans. Wirel. Commun.* **2023**. [CrossRef]
18. Oguine, K.J.; Oguine, O.C.; Bisallah, H.I. YOLO v3: Visual and Real-Time Object Detection Model for Smart Surveillance Systems(3s). In Proceedings of the 2022 5th Information Technology for Education and Development (ITED), Abuja, Nigeria, 1–3 November 2022.
19. Masum, M.I.; Sarwat, A.; Riggs, H.; Boymelgreen, A.; Dey, P. YOLOv5 vs. YOLOv8 in Marine Fisheries: Balancing Class Detection and Instance Count. *arXiv* **2024**, arXiv:2405.02312.
20. Khare, O.M.; Gandhi, S.; Rahalkar, A.M.; Mane, S. YOLOv8-Based Visual Detection of Road Hazards: Potholes, Sewer Covers, and Manholes. In Proceedings of the 2023 IEEE Pune Section International Conference (PuneCon), Pune, India, 14–16 December 2023.
21. 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]

22. Cheng, G.; Li, Q.; Wang, G.; Xie, X.; Min, L.; Han, J. SFRNet: Fine-Grained Oriented Object Recognition via Separate Feature Refinement. *IEEE Trans. Geosci. Remote. Sens.* **2023**, *61*, 5610510. [CrossRef]
23. Nabati, R.; Qi, H. RRPN: Radar Region Proposal Network for Object Detection in Autonomous Vehicles. In Proceedings of the IEEE International Conference on Image Processing, Taipei, Taiwan, 22–25 September 2019.
24. Han, J.; Ding, J.; Xue, N.; Xia, G.S. ReDet: A Rotation-equivariant Detector for Aerial Object Detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2785–2794.
25. Li, Y.; Li, X.; Dai, Y.; Hou, Q.; Liu, L.; Liu, Y.; Cheng, M.M.; Yang, J. LSKNet: A Foundation Lightweight Backbone for Remote Sensing. In Proceedings of the IEEE International Conference on Computer Vision, Paris, France, 2–6 October 2023.
26. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *Int. J. Comput. Vis.* **2018**, *128*, 642–656. [CrossRef]
27. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
28. Dai, L.; Liu, H.; Tang, H.; Wu, Z.; Song, P. AO2-DETR: Arbitrary-Oriented Object Detection Transformer. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 2342–2356. [CrossRef]
29. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
30. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
31. Li, X.; Hu, X.; Yang, J. Spatial Group-wise Enhance: Improving Semantic Feature Learning in Convolutional Networks. *arXiv* **2019**, arXiv:1905.09646.
32. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
33. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-Local Neural Networks. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef]
35. Hou, L.; Lu, K.; Yang, X.; Li, Y.; Xue, J. G-Rep: Gaussian Representation for Arbitrary-Oriented Object Detection. *Remote Sens.* **2023**, *15*, 757. [CrossRef]
36. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
37. Kim, K.; Lee, H.S. Probabilistic Anchor Assignment with IoU Prediction for Object Detection. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
38. Ming, Q.; Zhou, Z.; Miao, L.; Zhang, H.; Li, L. Dynamic Anchor Learning for Arbitrary-Oriented Object Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
39. 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, Virtually, 2–9 February 2021; pp. 3163–3171.
40. Chen, T.; Li, R.; Fu, J.; Jiang, D. Tucker Bilinear Attention Network for Multi-scale Remote Sensing Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Paris, France, 2–6 October 2023.
41. Chen, Z.; Yang, C.; Li, Q.; Zhao, F.; Zha, Z.J.; Wu, F. Disentangle Your Dense Object Detector. In Proceedings of the ACM-MM, Virtual, 20–24 October 2021.
42. Wang, J.; Song, L.; Li, Z.; Sun, H.; Sun, J.; Zheng, N. End-to-End Object Detection with Fully Convolutional Network. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
43. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.S.; Bai, X. Rotation-Sensitive Regression for Oriented Scene Text Detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
44. Yu, Y.; Da, F. Phase-Shifting Coder: Predicting Accurate Orientation in Oriented Object Detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023.
45. Yu, Y.; Yang, X.; Li, Q.; Zhou, Y.; Zhang, G.; Da, F.; Yan, J. H2RBox-v2: Incorporating Symmetry for Boosting Horizontal Box Supervised Oriented Object Detection. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023.
46. Zhou, Y.; Yang, X.; Zhang, G.; Wang, J.; Liu, Y.; Hou, L.; Jiang, X.; Liu, X.; Yan, J.; Lyu, C.; et al. MMRotate: A Rotated Object Detection Benchmark using PyTorch. In Proceedings of the 30th ACM International Conference on Multimedia, Lisbon, Portugal, 10–14 October 2022.
47. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020.
48. Ma, T.; Mao, M.; Zheng, H.; Gao, P.; Wang, X.; Han, S.; Ding, E.; Zhang, B.; Doermann, D. Oriented Object Detection with Transformer. *arXiv* **2021**, arXiv:2106.03146.

49. Guo, Y.; Chen, J.; Wang, J.; Chen, Q.; Cao, J.; Deng, Z.; Xu, Y.; Tan, M. Closed-loop Matters: Dual Regression Networks for Single Image Super-Resolution. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
50. 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 International Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021.
51. Yang, X.; Yan, J. On the Arbitrary-Oriented Object Detection: Classification based Approaches Revisited. *Int. J. Comput. Vis.* **2022**, *130*, 1340–1365. [CrossRef]
52. 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]
53. 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, Glasgow, UK, 23–28 August 2020.
54. Wei, H.; Zhang, Y.; Chang, Z.; Li, H.; Wang, H.; Sun, X. Oriented Objects as pairs of Middle Lines. *J. Photogramm. Remote. Sens.* **2020**, *169*, 268–279. [CrossRef]
55. Pu, Y.; Wang, Y.; Xia, Z.; Han, Y.; Wang, Y.; Gan, W.; Wang, Z.; Song, S.; Huang, G. Adaptive Rotated Convolution for Rotated Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023.
56. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Salt Lake City, UT, USA, 18–22 June 2018.
57. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; et al. Hybrid Task Cascade for Instance Segmentation. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
58. 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. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
59. Zhou, Q.; Yu, C.; Wang, Z.; Li, H. Point RCNN: An Angle-Free Framework for Rotated Object Detection. *Remote Sens.* **2022**, *14*, 2605. [CrossRef]
60. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.