remote sensing

MDPI

Article

An Efficient Object Detection Algorithm Based on Improved YOLOv5 for High-Spatial-Resolution Remote Sensing Images

Feng Cao [1], Bing Xing [1], Jiancheng Luo [2,*], Deyu Li [1], Yuhua Qian [1], Chao Zhang [1], Hexiang Bai [1] and Hu Zhang [1]

[1] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China; caof@sxu.edu.cn (F.C.); 202122408085@email.sxu.edu.cn (B.X.); lidy@sxu.edu.cn (D.L.); jinchengqyh@sxu.edu.cn (Y.Q.); czhang@sxu.edu.cn (C.Z.); baihx@sxu.edu.cn (H.B.); zhanghu@sxu.edu.cn (H.Z.)

[2] State Key Laboratory of Remote Sensing Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100101, China

* Correspondence: luojc@aircas.ac.cn

Abstract: The field of remote sensing information processing places significant research emphasis on object detection (OD) in high-spatial-resolution remote sensing images (HSRIs). The OD task in HSRIs poses additional challenges compared to conventional natural images. These challenges include variations in object scales, complex backgrounds, dense arrangement, and uncertain orientations. These factors contribute to the increased difficulty of OD in HSRIs as compared to conventional images. To tackle the aforementioned challenges, this paper introduces an innovative OD algorithm that builds upon enhancements made to the YOLOv5 framework. The incorporation of RepConv, Transformer Encoder, and BiFPN modules into the original YOLOv5 network leads to improved detection accuracy, particularly for objects of varying scales. The C3GAM module is designed by introducing the GAM attention mechanism to address the interference caused by complex background regions. To achieve precise localization of densely arranged objects, the SIoU loss function is integrated into YOLOv5. The circular smooth label method is used to detect objects with uncertain directions. The effectiveness of the suggested algorithm is confirmed through its application to two commonly utilized datasets, specifically HRSC2016 and UCAS-AOD. The average detection accuracies achieved on these datasets are 90.29% and 90.06% respectively, surpassing the performance of other compared OD algorithms for HSRIs.

Keywords: high-spatial-resolution remote sensing images; object detection; deep learning; feature fusion; attention mechanism

Citation: Cao, F.; Xing, B.; Luo, J.; Li, D.; Qian, Y.; Zhang, C.; Bai, H.; Zhang, H. An Efficient Object Detection Algorithm Based on Improved YOLOv5 for High-Spatial-Resolution Remote Sensing Images. *Remote Sens.* **2023**, *15*, 3755. https://doi.org/10.3390/rs15153755

Academic Editors: Jiao Shi, Maoguo Gong, Kai Qin and Yu Lei

Received: 15 June 2023
Revised: 25 July 2023
Accepted: 25 July 2023
Published: 28 July 2023

1. Introduction

Technological advancements in the field of remote sensing have led to a substantial growth in the volume of high-spatial-resolution remote sensing images (HSRIs). These images encompass a vast amount of valuable information for the purpose of earth observation. As a result, effectively acquiring and utilizing this information has become a crucial area of research in remote sensing information processing. The process of object detection (OD) for HSRIs entails extracting features from such images to identify ground targets' categories and obtain their rectangular bounding box coordinates. In the field of remote sensing information processing, this subject has gained substantial popularity and garnered considerable attention from researchers. The research results of OD in HSRIs have found extensive utilization in diverse domains, encompassing urban planning, disaster prediction, natural disaster response, disaster assessment, and military decision-making [1].

The intricate characteristics of the earth's surface make OD in HSRIs a formidable challenge. Several factors impede the process, including variations in target scales, complex backgrounds, dense arrangements, and uncertain orientations [2]. The unique characteristics of HSRIs pose challenges in attaining high precision for OD [3]. Currently, in the

field of OD in HSRIs, there are two primary categories of algorithms. The first category comprises traditional algorithms, including sliding window and template matching. The second category consists of deep learning (DL)-based algorithms. Presently, DL-based algorithms have gained significant popularity due to their superior performance in terms of both accuracy and speed, surpassing traditional algorithms. However, most OD algorithms based on DL cannot recognize the orientation of objects, making them no longer suitable for more challenging HSRIs. Consequently, algorithms that detect arbitrary-oriented bounding boxes have become the prevailing standard for OD in HSRIs. Two distinct categories can be used to classify these algorithms: two-stage [4–6] algorithms and one-stage [7,8] algorithms.

In two-stage algorithms, the detection process consists of two stages: the first stage involves generating candidate regions, while the second stage focuses on extracting features from these regions to aid in object recognition. To represent arbitrary-oriented bounding boxes, these detection algorithms commonly utilize the five-parameter representation (x, y, w, h, θ). In this representation, (x, y) indicates the center position of the bounding box, (w, h) indicates the width and height of the bounding box, and θ represents the angle of the bounding box. In the algorithms, the angle of the bounding box can be learned in either the first or second stage of the algorithm. The angle of the bounding box can be acquired during either the initial or subsequent stage. As an illustration, algorithms like R^2CNN [9] and ROI Transformer [10] generate candidate regions in a horizontal orientation during the first stage. Subsequently, in the second stage, these algorithms perform angle regression to determine the bounding box orientation. In contrast, R^2PN [11], R-DFPN [12], and ICN [13] directly generate oriented candidate regions in the first stage.

Unlike two-stage detection algorithms, one-stage algorithms are specifically designed as end-to-end detection algorithms. They bypass the need for a separate candidate region generation stage and can directly classify and estimate the position of objects with any orientation in the image. They also have fewer parameters and are easier to converge. Common one-stage OD algorithms include RetinaNet-O [14], DAL [15], RSDet [16], R^3Det [17], and S^2A-Net [18]. RetinaNet-O is an improved algorithm based on RetinaNet. It achieves arbitrary-oriented OD through five-parameter regression. DAL algorithm adopts a dynamic anchor learning strategy, which assigns labels more efficiently by evaluating the localization potential of anchors. In order to tackle the problem of loss discontinuity resulting from the periodic nature of angles in five-parameter regression and the inconsistency of regression parameters, the RSDet algorithm directly regresses the four points of rotated boxes using an eight-parameter representation. The R^3Det algorithm combines horizontal anchor boxes and rotated anchor boxes. During the initial detection stage, the algorithm utilizes horizontal anchor boxes to accelerate the process and generate a larger quantity of candidate boxes. In the refinement stage, it uses rotated anchor boxes to adapt to dense target scenarios. Consisting of two distinct modules, namely the feature alignment module and the detection module, the S^2A-Net algorithm leverages an anchor refinement network within the feature alignment module to produce anchors of exceptional quality. Conversely, the detection module incorporates active rotation filters to encode orientation information, resulting in the generation of orientation-sensitive and orientation-invariant features. By employing this approach, the inconsistency between classification scores and localization accuracy is effectively resolved. When confronted with orientation uncertainty, the majority of the OD algorithms mentioned above employ angle regression to forecast the orientation of detection boxes. However, this approach ignores the issue of boundary discontinuity. If the predicted result falls outside the predefined range, it leads to a substantial loss value, resulting in unstable training outcomes and impacting the model's detection performance.

As one of the representatives of one-stage OD algorithms, the YOLO series has garnered widespread attention and has been updated at an increasingly rapid pace. Since the introduction of YOLOv1 [19], the YOLO series of algorithms have undergone rapid updates and have reached the latest version, YOLOv8, with continuously improving accuracy and speed. Researchers have also made further advancements based on the YOLO framework in their respective fields of study. For example, Choi et al. [20]. proposed Gaussian YOLOv3,

building upon YOLOv3 [21], to effectively apply the improved algorithm to the field of autonomous driving. Similarly, Wang et al. [22] made advancements in YOLOv3 to address the challenges in pavement surface pothole measurement. Wu et al. [23] combined local fully convolutional neural networks with YOLOv5, achieving progress in small object detection in HSRIs. Zhang et al. [24]. improved upon YOLOv5, effectively reducing the false detection rate of occluded vehicle targets. Zhao et al. [25] combined YOLOv5 with Transformers to effectively address the challenge of OD in images captured by drones.

The progression from YOLOv1 to YOLOv8 can be categorized into two stages: the first stage encompasses YOLOv1 to YOLOv5, while the second stage encompasses YOLOv6 [26] to YOLOv8. In the first stage, YOLOv5 emerges as the algorithm of utmost innovation and representation, successfully attaining a commendable equilibrium between accuracy and speed. The improvements in YOLOv6, YOLOv7 [27], and YOLOv8 are all based on YOLOv5. Therefore, by harnessing the robust OD capabilities of YOLOv5 and taking into account the distinctive traits of OD in HSRIs, this paper presents a refined algorithm based on YOLOv5 for detecting objects in HSRIs.

The primary contributions can be summarized as follows:

1. Our proposal designs a RepConv module that enhances the detection accuracy of small-scale objects without introducing additional inference time. Additionally, we incorporate a Transformer Encoder structure to capture global contextual information, thereby improving the detection accuracy of large-scale objects. In order to achieve a balance in feature information across various scales and enhance the detection accuracy of multi-scale objects, we substitute the PANet structure in YOLOv5 with BiFPN.
2. To address the interference caused by complex background regions in HSRIs, we design a C3GAM module by introducing the GAM attention mechanism, which aids the model in effectively localizing regions that contain the target.
3. To enhance the localization accuracy of anchor boxes and improve the precision of boundary recognition in HSRIs with dense object arrangements, we incorporate the SIoU loss function.
4. To tackle the issue of uncertain target direction and mitigate the problem of disjointed boundaries caused by angle regression, we suggest the adoption of the circular smooth label method as an effective solution.

2. Network Structure of YOLOv5

YOLOv5 is one of the most representative algorithms in YOLO target detection, including five network structures: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Among them, YOLOv5n stands out with its comparatively lower depth and feature map width. The remaining four networks progressively increase both depth and feature map width in comparison to YOLOv5n. For the purpose of enhancement, this paper selects YOLOv5s as the foundational model. The network structure of YOLOv5s can be categorized into three components: Backbone, Neck, and Head, as depicted in Figure 1.

2.1. Backbone

The Backbone consists of three modules: Conv, C3, and SPPF. The Conv module further encapsulates three functional modules: the convolutional layer (Conv2d), normalization layer (BatchNorm2d), and activation function (SiLU). By applying convolution, normalization, and activation to the input features, the module produces output features. The C3 module comprises multiple Bottleneck modules and three standard convolutional layers. The number of Bottlenecks varies depending on the network depth. The C3 module, which consists of two branches, plays a crucial role in learning residual features. One branch contains multiple stacked Bottlenecks and three standard convolutional layers, while the other branch contains a basic convolutional block. The results from the two branches are merged by concatenating them. The SPPF module sequentially applies multiple small-size

pooling kernels to fuse feature maps with varying receptive fields. This enhances feature map representation and further improves computational speed.

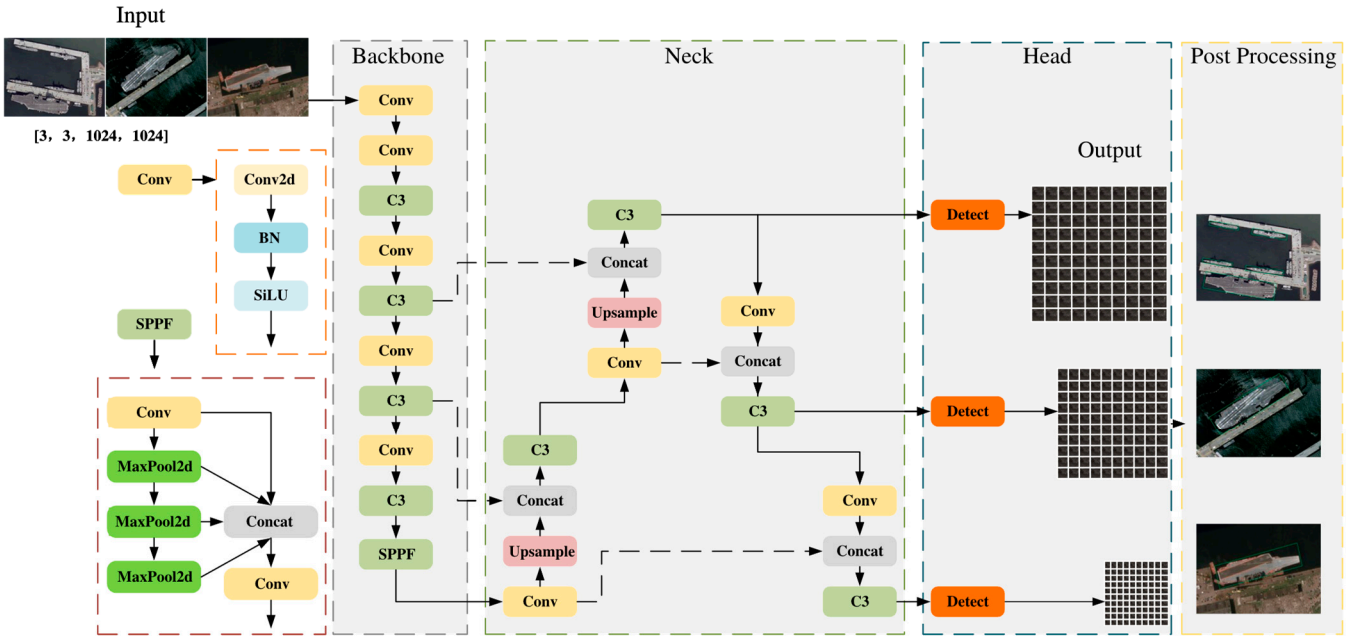


Figure 1. Architecture of YOLOv5s.

2.2. Neck

The Neck component comprises two elements: the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN). FPN represents a top-down feature pyramid network, whereas PAN stands as a path aggregation network. FPN integrates features at different levels in a hierarchical manner, progressing from top to bottom. By capitalizing on the high resolution of low-level features and the abundant semantic information from high-level features, it independently predicts multi-scale features. PAN, as an enhancement of FPN, introduces a bottom-up feature pyramid structure to augment its capabilities. It preserves more shallow positional features, further enhancing the overall feature extraction capability.

2.3. Head

The Head section serves as the output layer of the algorithm and consists primarily of three Detect detectors. It performs OD by using grid-based anchors on feature maps at different scales. Each Detect module receives features from the Neck layer at three different scales. It uses convolutional operations to adjust the channel dimension of the output layer and then refines the position of the anchors based on the predicted results. Ultimately, the predicted results undergo mapping back to the original image after traversing a post-processing module. The ultimate detection results are obtained by applying non-maximum suppression, which helps eliminate a significant number of overlapping candidate boxes.

3. Our Work

An efficient OD algorithm for HSRIs is introduced, utilizing the YOLOv5s framework. The proposed algorithm's network architecture, depicted in Figure 2, closely resembles that of YOLOv5s, consisting of three main components. The subsequent sections will provide a detailed description of the key modules in this algorithm.

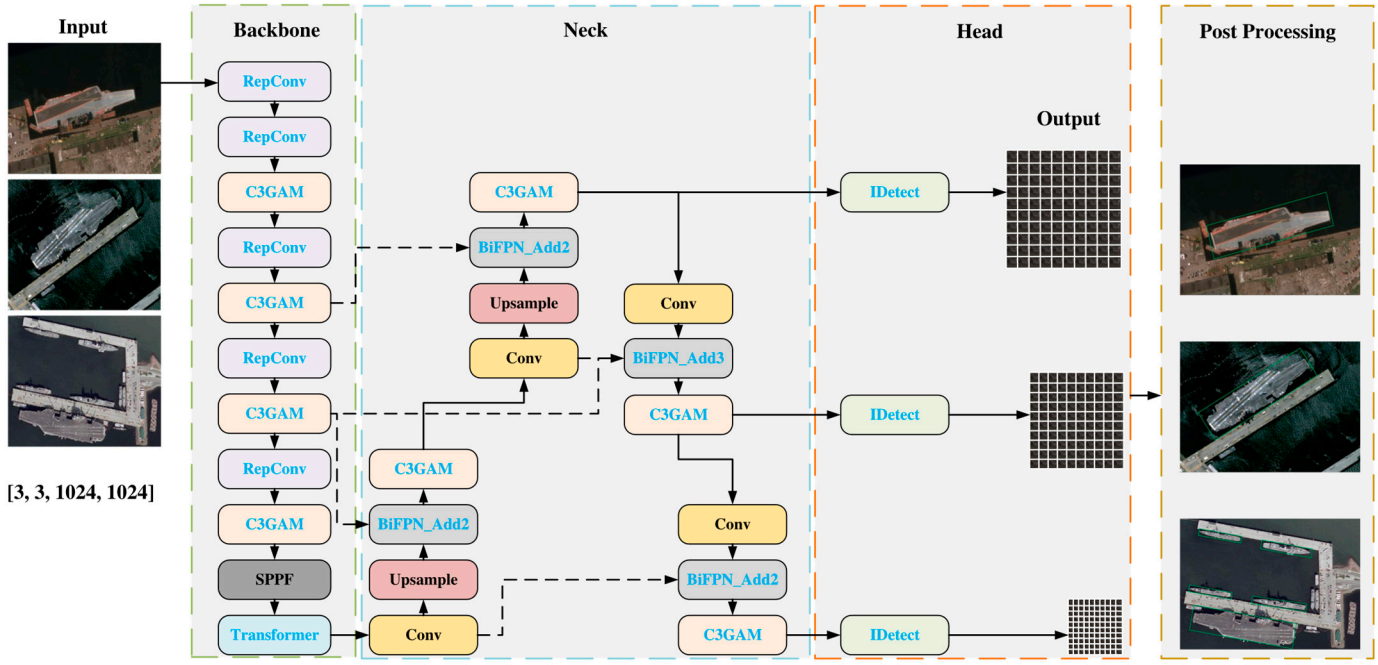


Figure 2. Architecture of the improved YOLOv5s.

3.1. RepConv Module

By incorporating the RepConv module into YOLOv5, the feature representation capability for small objects is enhanced due to the multi-branch structure of RepConv. This leads to improved accuracy in recognizing small objects. Moreover, during inference, the parallel branches of RepConv are transformed into a single branch using reparameterization techniques [28], maintaining the same structure as YOLOv5 without increasing the inference time.

This module incorporates a parallel 1×1 Conv layer within the 3×3 Conv layer of the Backbone, effectively widening the convolutional module. During the inference phase, the outputs of the parallel branch are consolidated into the 3×3 Conv layer. With the inclusion of this modification, the detection accuracy of small objects in HSRIs is enhanced, without introducing any extra inference time in the algorithm. Throughout the training process, the RepConv module utilizes a multi-branch structure. Figure 3a illustrates the module's structure when there exists a discrepancy in the number of input feature channels and output feature channels. In contrast, Figure 3b depicts the module's structure when the number of input feature channels matches the number of output feature channels. In the inference phase, the outputs of the parallel branches are combined within the 3×3 Conv layer, resulting in the transformation of the multi-branch structure into a single-path structure, as illustrated in Figure 3c.

3.2. Transformer Encoder Module

The convolutional modules used in YOLOv5 primarily focus on local features, resulting in subpar detection performance for large-scale objects. The core of the Transformer Encoder module lies in its multi-head self-attention mechanism, which enables capturing global features and exhibits strong capabilities in detecting large-scale objects.

The CNN operator encounters the issue of limited local receptive fields when extracting features for OD. To capture global information, multiple layers need to be stacked. Nevertheless, with an increase in the number of layers, there exists the possibility of information degradation, resulting in a concentration of feature attention in specific regions. On the other hand, Transformers possess a self-attention mechanism that effectively captures global information. Additionally, the use of multiple heads enables mapping to various spatial positions, thereby enhancing the model's expressive capacity.

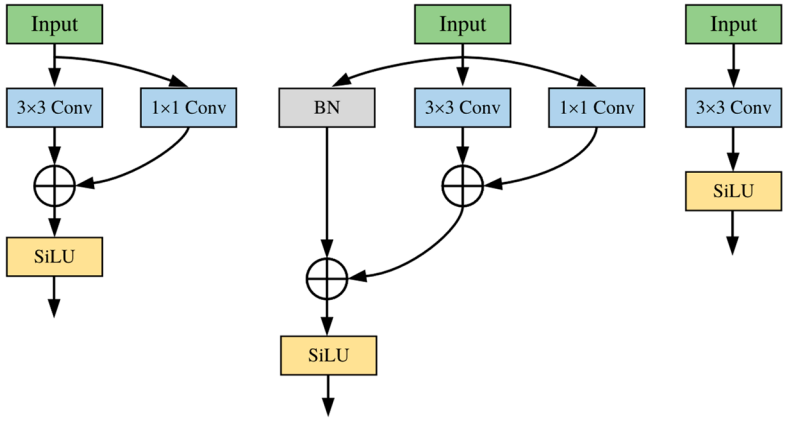


Figure 3. Architecture of the designed RepConv module.

To bolster the global feature extraction capabilities of YOLOv5s, this paper introduces a Transformer Encoder module into the Backbone component, as depicted in Figure 4. The figure clearly demonstrates that each Transformer Encoder is composed of two sub-layers. The initial sub-layer includes LayerNorm, Multi-Head Attention, and Dropout. The input to this layer is data with dimensions (n, b, c); in this context, n represents the result of multiplying the width and height of the feature map, b signifies the count of input images within the network, and c denotes the number of feature channels. The input data undergo normalization through the LayerNorm layer, followed by Multi-Head Attention to compute similarities between targets. Lastly, the data flow through the Dropout layer to alleviate overfitting. The second sub-layer consists of LayerNorm and Multi-Layer Perceptron (MLP). LayerNorm serves a similar purpose as in the first sub-layer, while the MLP employs fully connected layers for linear transformations. Residual connections are established between each sub-layer. With the inclusion of the Transformer Encoder module, the improved algorithm acquires enhanced capabilities to capture global information and contextual details.

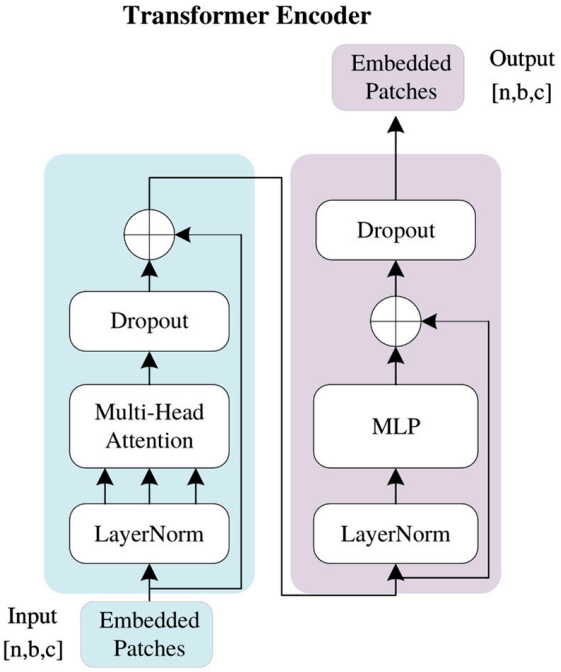


Figure 4. Architecture of the Transformer Encoder module.

3.3. C3GAM Module

HSRIs often contain a large amount of complex background, which introduces significant interference to the objects and leads to a decrease in detection accuracy. Therefore, to mitigate the interference caused by the background, this paper adopts GAM attention, which reduces the weights and weakens the features of the background through a weighted approach. This effectively eliminates the interference from the background.

The extraction of feature information for OD in HSRIs is significantly hindered by the presence of complex background information. To amplify the feature information within the target regions and mitigate the interference arising from the background, this paper introduces a new attention mechanism called the Global Attention Mechanism (GAM) embedded within the C3 module, resulting in the construction of the C3GAM module. The GAM [29] attention mechanism amplifies global interdependent features while reducing information diffusion. Figure 5 illustrates the network structure of GAM. The input feature is denoted as $F_1 \in R^{C\times H\times W}$, the intermediate state is denoted as F_2, and the output result is denoted as F_3. The GAM module can be defined as follows:

$$F_2 = M_c(F_1) \otimes F_1 \tag{1}$$

$$F_3 = M_s(F_2) \otimes F_2 \tag{2}$$

where M_c and M_s represent the channel attention module and spatial attention module, respectively, and $\otimes$ denotes the element-wise multiplication operation.

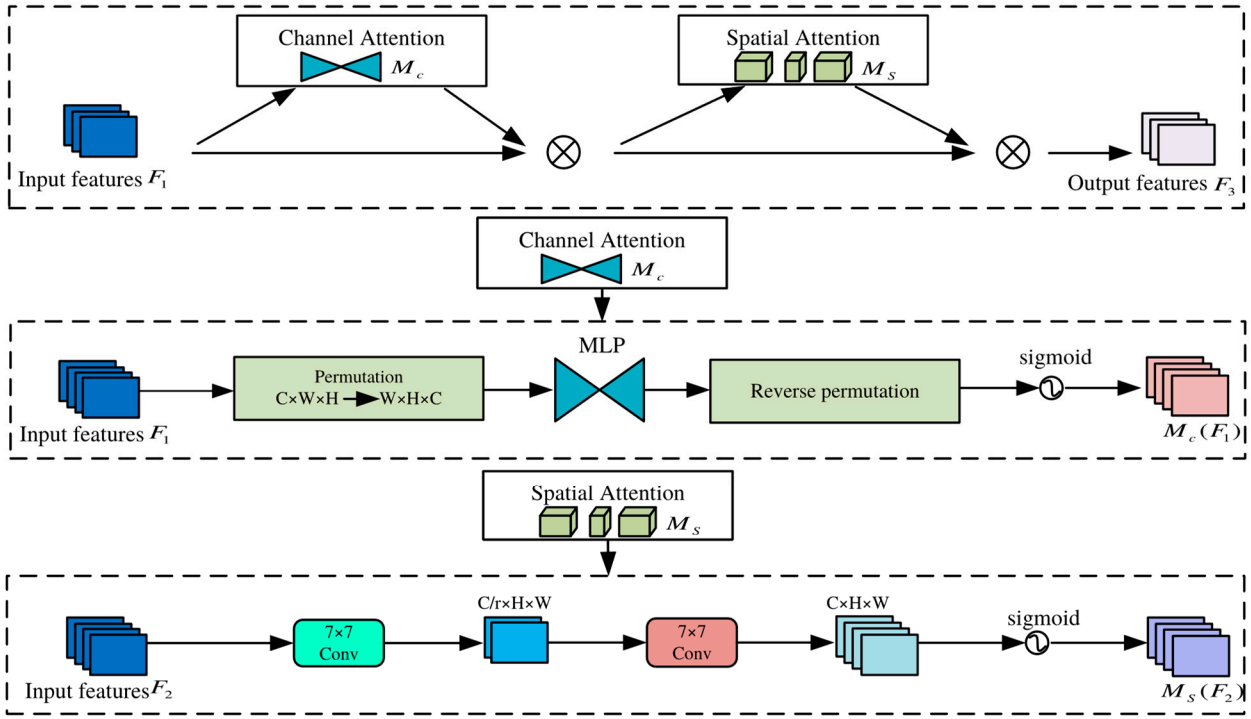


Figure 5. Architecture of the GAM module.

The comparison between the improved C3GAM module and the C3 module in the YOLOv5s network architecture is depicted in Figure 6. It reveals that the C3 module is composed of numerous stacked Bottleneck modules, as can be observed. This paper introduces the GAM attention into the Bottleneck module, effectively suppressing the interference caused by the background.

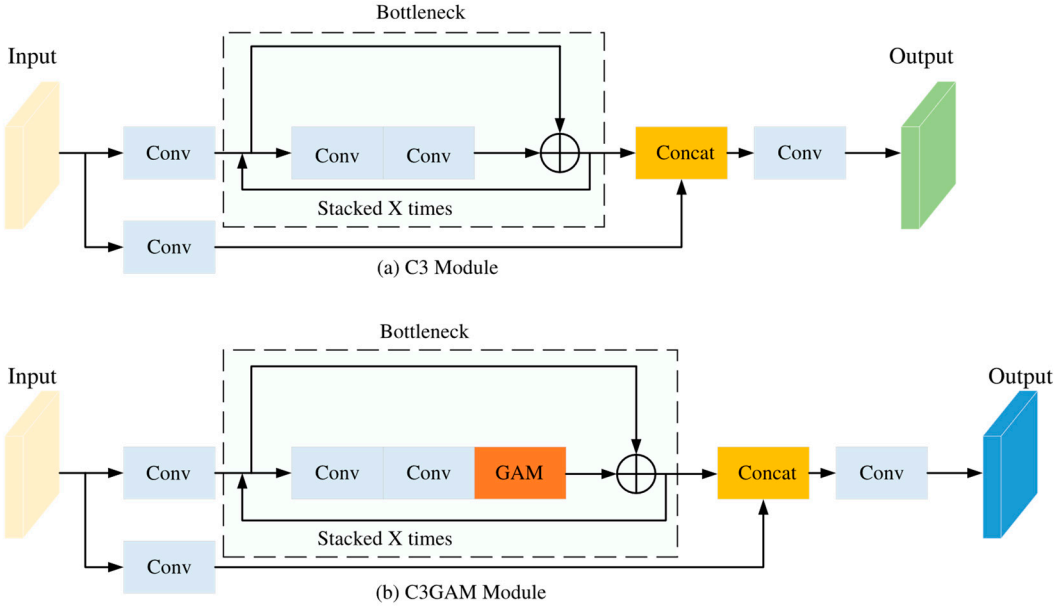


Figure 6. Comparison of architectures between C3 module and C3GAM module.

3.4. SIoU Loss Function Module

CIoU [30] and DIoU [31] are two commonly used regression loss functions in YOLO. CIoU builds upon DIoU by adding constraints on aspect ratios, resulting in faster convergence compared to DIoU. SIoU [32], on the other hand, further improves upon CIoU by redefining the penalty term and introducing angle loss. It achieves a faster training speed and higher inference accuracy compared to CIoU.

YOLOv5s adopts CIoU as the regression loss function for bounding boxes. CIoU improves upon DIoU by incorporating scale loss and aspect ratio loss for the bounding boxes, making the predicted boxes more aligned with the ground truth boxes. However, CIoU neglects the orientation matching between the ground truth and predicted boxes, focusing solely on the aggregation of bounding box regression metrics. As a result, its training speed and prediction accuracy are lower compared to SIoU. Hence, SioU is opted as the regression loss function for the predicted boxes, encompassing four components: angle cost, distance cost, shape cost, and IoU cost.

(1) Angle cost

The schematic diagram in Figure 7 illustrates the angle cost, where B represents the predicted box with center coordinates $\left(b_{c_x}, b_{c_y}\right)$, and B^{GT} represents the ground truth box with center coordinates $\left(b_{c_x}^{gt}, b_{c_y}^{gt}\right)$. σ represents the distance between the center coordinates of B and B^{GT}, as denoted by Equation (3). C_h represents the height between the center coordinates of B and B^{GT}, as expressed by Equation (4). $\frac{C_h}{\sigma}$ is essentially equal to $\sin(\alpha)$, as indicated by Equation (5). In the end, the formula for angle cost can be obtained, as depicted in Equation (6). When α equals $\frac{\pi}{2}$ or 0, it can be observed that the angle cost is 0. During the training process, if $\alpha < \frac{\pi}{4}$, α is minimized, otherwise, β is minimized.

$$\sigma = \sqrt{\left(b_{c_x}^{gt} - b_{c_x}\right)^2 + \left(b_{c_y}^{gt} - b_{c_y}\right)^2} \tag{3}$$

$$C_h = \max\left(b_{c_y}^{gt}, b_{c_y}\right) - \min\left(b_{c_y}^{gt}, b_{c_y}\right) \tag{4}$$

$$\frac{C_h}{\sigma} = \sin(\alpha) \tag{5}$$

$$\Lambda = 1 - 2 \times \sin^2\left(\arcsin\left(\frac{C_h}{\sigma}\right) - \frac{\pi}{4}\right) \tag{6}$$

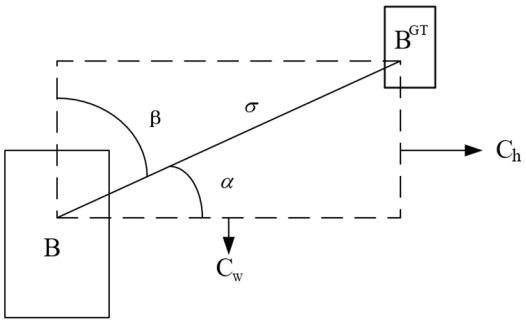


Figure 7. Angle cost diagram.

(2) Distance cost

The distance cost is shown in Figure 8, where B represents the predicted box with center coordinates $\left(b_{c_x}, b_{c_y}\right)$, B^{GT} represents the ground truth box with center coordinates $\left(b_{c_x}^{gt}, b_{c_y}^{gt}\right)$; C_w and C_h denote the width and height of the minimum bounding rectangle of B and B^{GT}, respectively. The term ρ_x represents the squared ratio of the difference in x-axis coordinates between the B and B^{GT} to C_w, while ρ_y represents the squared ratio of the difference in y-axis coordinates between B and B^{GT} to C_h, as described by Equation (7). The final expression of the distance cost is shown in Equation (8).

$$\rho_x = \left(\frac{b_{c_x}^{gt} - b_{c_x}}{C_w}\right)^2, \ \rho_y = \left(\frac{b_{c_y}^{gt} - b_{c_y}}{C_h}\right)^2, \ \gamma = 2 - \Lambda \tag{7}$$

$$\Delta = \sum_{t=x,y}\left(1 - e^{-\gamma\rho_t}\right) \tag{8}$$

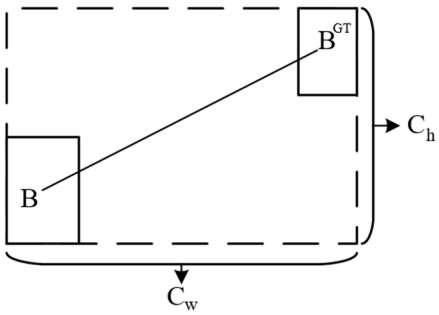


Figure 8. Distance cost diagram.

(3) Shape cost

According to Figure 7, w and h represent the width and height of B, while w^{gt} and h^{gt} represent the width and height of B^{GT}, respectively. ω_w represents the absolute difference between w and w^{gt} divided by the maximum value between w and w^{gt}, and ω_h represents the absolute difference between h and h^{gt} divided by the maximum value between h and h^{gt}, as shown in Equation (9). The term θ represents the importance of the shape cost. The final expression of the shape cost can be obtained as shown in Equation (10).

$$\omega_w = \frac{|w - w^{gt}|}{\max(w, w^{gt})}, \quad \omega_h = \frac{|h - h^{gt}|}{\max(h, h^{gt})} \tag{9}$$

$$\Omega = \sum_{t=w,h} \left(1 - e^{-w_t}\right)^{\theta} \tag{10}$$

(4) IoU cost

According to Figure 9, $B \cap B^{GT}$ and $B \cup B^{GT}$ represent the intersection and union of B and B^{GT}, respectively. The IoU expression is shown in Equation (11). Ultimately, the total loss value can be obtained as shown in Equation (12).

$$IoU = \frac{B \cap B^{GT}}{B \cup B^{GT}} \tag{11}$$

$$L_{\mathrm{SIoU}} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{12}$$

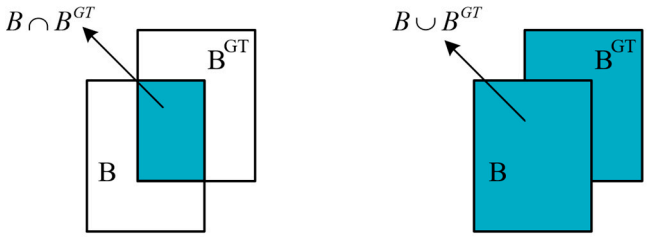


Figure 9. Illustration of Intersection and Union.

3.5. BiFPN Module

The FPN [33] has been extensively used for multi-scale feature fusion since its introduction, leading to the development of various cross-scale feature fusion networks such as PANet [34] and NAS-FPN [35]. These networks typically treat inputs from different scales equally. BiFPN [36] enhances FPN by incorporating learnable weights that help determine the significance of various input features, thus achieving a better balance of information across different scales. Hence, this paper utilizes BiFPN to substitute the feature fusion approach in the Neck section. Figure 10 provides an illustration of the structure of the BiFPN model, where P1, P2, and P3 represent features of different scales generated by the Backbone part. BiFPN_Add2 and BiFPN_Add3 are feature fusion modules that combine features from the current layer and the preceding layer, employing weighted aggregation for fusion.

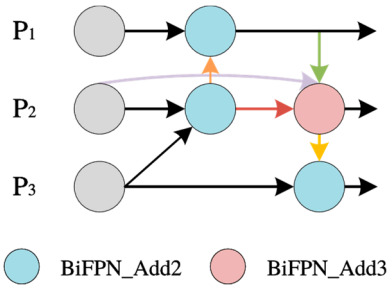


Figure 10. Architecture of the BiFPN module.

3.6. CSL (Circular Smooth Label) Module

YOLOv5s is a conventional object detector that primarily focuses on horizontal box detection and lacks the ability to handle uncertain object orientations in HSRIs. To overcome this limitation, this paper suggests integrating the CSL module into YOLOv5s, allowing for the prediction of target orientations. CSL transforms the defined range of angles into categories and achieves more robust angle prediction through classification. Please refer

to Figure 11 for visualization. In the figure, the lines of different colors represent different window functions, where the yellow line represents the pulse function, the green line represents the rectangular function, the blue line represents the triangular function, and the red line represents the Gaussian function.

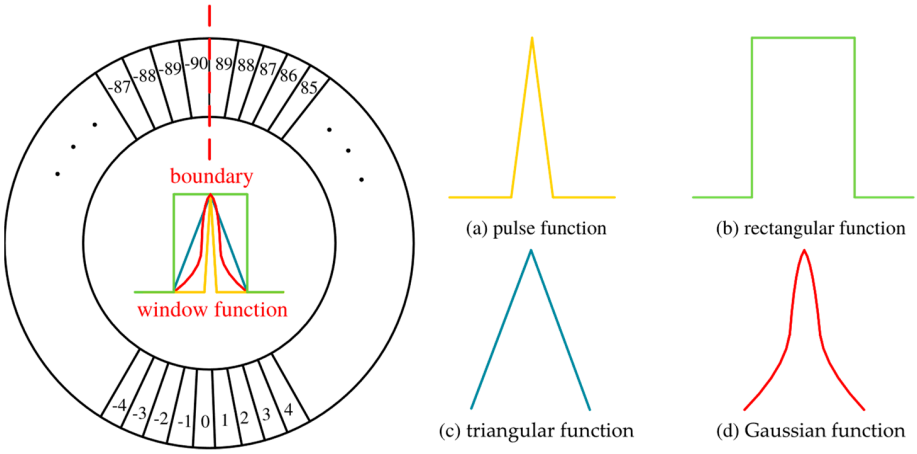


Figure 11. Architecture of the CSL module [37].

CSL consists of cyclic circular label encoding, where the assigned label values are smooth and have a certain variance. The expression for CSL is as follows:

$$CSL(x) = \begin{Bmatrix} g(x), \theta - r < x < \theta + r \\ 0, \quad otherwise \end{Bmatrix} \tag{13}$$

The function $g(x)$ symbolizes a window function, with the radius θ of the window function correlating to the angle of the current bounding box. A desirable window function exhibits the following characteristics: periodicity, symmetry, maximum value, and monotonicity. Commonly used window functions include the pulse function, rectangular function, triangular function, and Gaussian function. From Figure 11, it can be seen that the label values are continuous at the boundaries and are not affected by the periodicity of CSL, thus avoiding accuracy errors.

3.7. IDetect Module

Human analysis of the same object can be conducted from multiple perspectives. However, when training convolutional neural networks, typically only one perspective is provided, making it difficult for the obtained features to be applicable to other tasks. The primary factor contributing to this problem is that the model solely focuses on extracting neural features while neglecting the acquisition and utilization of implicit knowledge, which holds significant value in analyzing diverse tasks.

In the context of neural networks, the shallow features observed by the network, which correspond to explicit knowledge, are commonly referred to as explicit knowledge. The deep features, which are unobservable and unrelated to observations, are defined as implicit knowledge. As a result, the IDetect module is developed in this paper to blend implicit knowledge and explicit knowledge within the Head section, leading to a notable enhancement in the algorithm's overall performance. As shown in Figure 12, the structure of the IDetect module is divided into two branches: training and inference. During training, the input data are first fused through the ImplicitA module (initialized as a learnable variable with a value of 0) using addition. It then passes through the Conv module to adjust the output channels. Finally, it undergoes multiplication fusion through the ImplicitM module (initialized as a learnable variable with a value of 1) to obtain the output result. During inference, only one layer of the Conv module is applied to adjust the output channels.

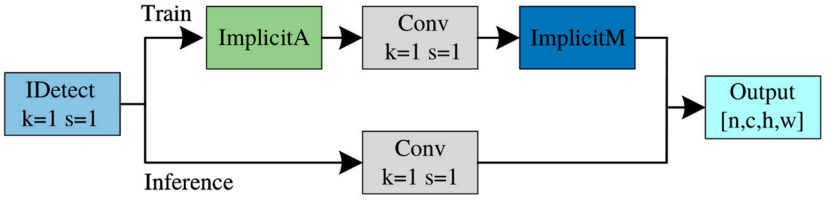


Figure 12. Architecture of the IDetect module.

4. Experiments

The experimental setup utilized CUDA 10.0 as the computing platform, Ubuntu 18.04 as the operating system, Intel i7-7700K as the processor, NVIDIA GTX 3090 with 24 GB of VRAM as the graphics card, and PyTorch 1.10.0 as the DL framework. The performance of the algorithm is assessed using two datasets specific to detect objects in HSRIs: HRSC2016 and UCAS-AOD.

4.1. Dataset

HRSC2016, which was introduced by Northwestern Polytechnical University in 2016, is acknowledged as one of the most challenging datasets for detecting ships in remote sensing. The dataset comprises 1061 HSRIs obtained from Google Earth, accompanied by 2976 instances annotated with rotated bounding boxes to facilitate the detection of targets. The dataset encompasses images with diverse resolutions, ranging from 2 m to 0.4 m. The images encompass a range of sizes, spanning from 300 × 300 to 1500 × 900, with a majority of them exceeding dimensions of 1000 × 600. For the experimental setup, single-class object recognition is conducted using three different sets of images. The training set comprises 436 images, with a total of 1207 samples. The validation set comprises 181 images, with a total of 541 samples. Lastly, the test set comprises 444 images, with a total of 1228 samples. Figure 13 displays a subset of the HRSC2016 dataset, highlighting the significant scale variations and complex background challenges present in remote sensing images.

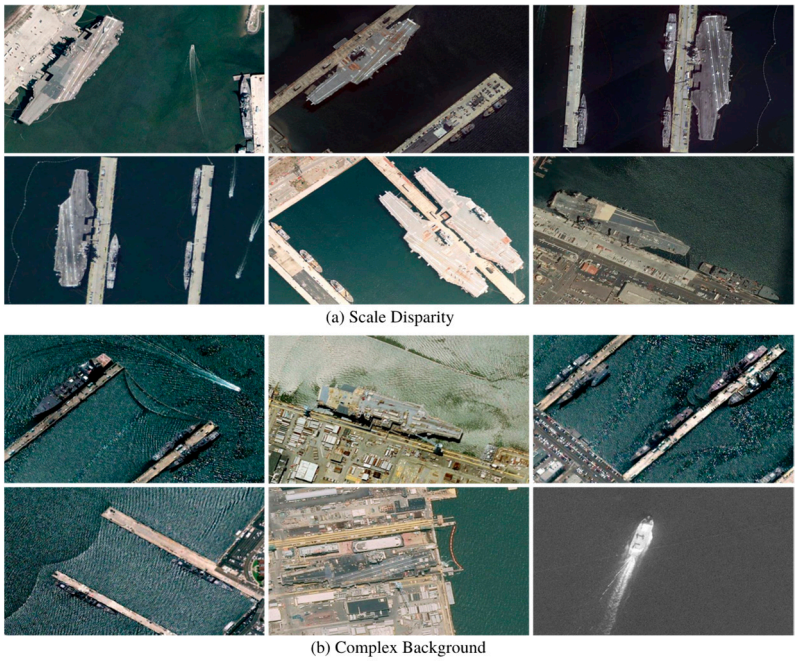


Figure 13. Sample images from the HRSC2016 dataset.

UCAS-AOD is a dataset specifically designed for aircraft and car detection, consisting of 1000 images containing 7482 instances of aircraft and 510 images containing 7114 instances of cars. The dataset is split into training, validation, and test sets in a ratio

of 5:2:3. The training set consists of 755 images, the validation set contains 302 images, and the test set comprises 453 images. All images have dimensions close to 1280 × 659. Figure 14 displays a subset of the UCAS-AOD dataset, primarily highlighting the dense arrangement of objects and the issue of orientation uncertainty in HSRIs.

(a) Dense Arrangement

(b) Uncertain Direction

Figure 14. Sample images from the UCAS-AOD dataset.

4.2. Experimental Parameter Settings

The Adam optimizer is employed in the experiments, using a momentum coefficient of 0.937 and a learning rate of 0.001. The IoU thresholds are set to 0.10, 0.20, 0.25, and 0.30, while the confidence thresholds for detecting targets are set to 0.10, 0.20, 0.30, and 0.40. The model's loss values reached a stable state after 350 iterations on the two experimental datasets, as illustrated in Figure 15. Hence, the number of iterations is set to 400.

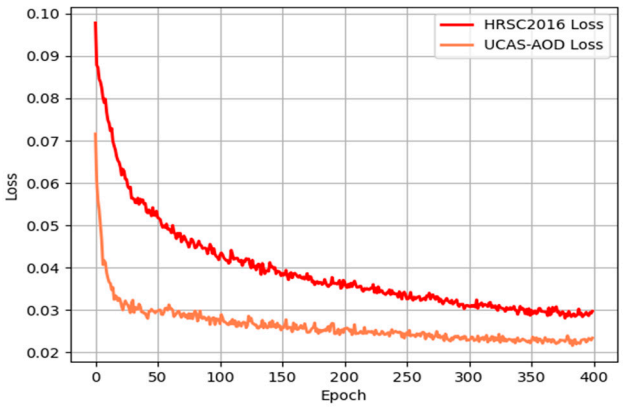


Figure 15. Loss function curves for the HRSC2016 and UCAS-AOD datasets.

4.3. Experimental Evaluation Metrics

The Intersect Over Union (IoU) threshold has a direct impact on the output prediction frame, with a higher threshold typically resulting in improved prediction accuracy. In this experiment, the mean Average Precision (mAP) metric is employed as the main evaluation indicator. mAP is calculated based on the precision–recall (P-R) curve in multi-class OD,

measuring the accuracy and recall for each class individually. The precision (P), recall (R), and mAP values are computed using Formulas (14)–(16).

$$P = \frac{TP}{TP + FP} \tag{14}$$

$$R = \frac{TP}{TP + FN} \tag{15}$$

$$mAP = \frac{1}{K}\sum_{K=1}^{K} AP(P,\ R,\ K) \tag{16}$$

Among these, TP represents true positives, which signifies the count of correctly detected positive samples. False positives, denoted as FP, represent the count of negative samples erroneously identified as positive detections. FN represents false negatives, indicating the count of positive samples erroneously identified as negative detections. K represents the count of target classes, whereas AP denotes the average precision.

4.4. Analysis of Experimental Results

4.4.1. Analysis of the Experimental Results on HRSC2016

The precision comparison results of the proposed OD algorithm, when evaluated against state-of-the-art one-stage and two-stage OD algorithms, are displayed in Table 1. Figure 16 exhibits the chosen detection results achieved by the proposed OD algorithm.

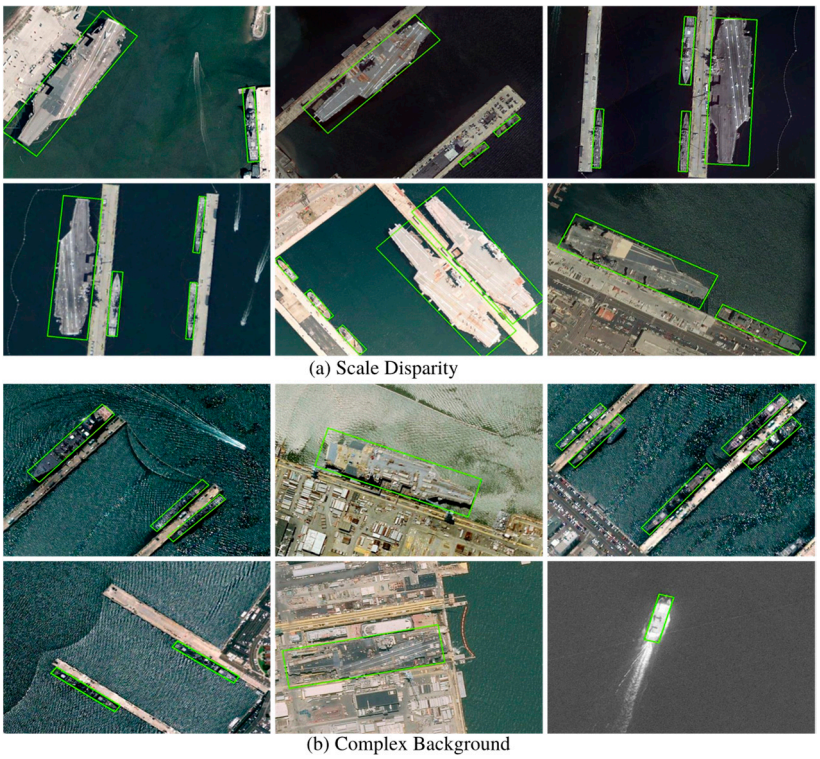
Table 1. Accuracy comparison of different OD algorithms on the HRSC2016 dataset.

Algorithms	Backbone	Size	Number of Anchors	mAP (%)
Two-stage:				
R^2CNN [9]	ResNet101	800 × 800	21	73.07
RC1 and RC2 [38]	VGG16	-	-	75.70
RRPN [39]	ResNet101	800 × 800	54	79.08
R^2PN [11]	VGG16	-	24	79.60
RoITrans [10]	ResNet101	512 × 800	5	86.20
Gliding Vertex [40]	ResNet101	512 × 800	5	88.20
One-stage:				
RRD [41]	VGG16	384 × 384	13	84.30
R^3Det [17]	ResNet101	800 × 800	21	89.26
R-Retinanet [14]	ResNet101	800 × 800	121	89.18
PIOU [42]	DLA-34	512 × 512	-	89.20
R^3Det-DCL [43]	ResNet101	800 × 800	21	89.46
FPN-CSL [37]	ResNet101	800 × 800	21	89.62
DAL [15]	ResNet101	800 × 800	3	89.77
S^2A-Net [18]	ResNet101	1024 × 1024	1	90.17
BBAVectors [44]	ResNet101	608 × 608	-	88.60
YOLOv6 [26]	EfficientRep	1024 × 1024	-	85.42
YOLOv7 [27]	ELAN-Net	1024 × 1024	3	86.11
YOLOv8	CSP-DarkNet	1024 × 1024	-	85.70
Ours	**YOLOv5s**	**1024 × 1024**	**3**	**90.29**

Based on the data displayed in Table 1, the proposed algorithm outperforms all the compared algorithms, achieving an mAP of 90.29%. Compared to the algorithms in Table 1, our algorithm performs OD on large-scale images of 1024 × 1024, which is advantageous for object recognition. By using a preset number of three Anchors, the algorithm effectively reduces computational complexity and achieves higher detection accuracy at a lower cost.

Based on Table 1, the Gliding Vertex algorithm, a two-stage object detection algorithm, achieves the highest detection accuracy of 88.20%, which is improved by 2.09% compared to our algorithm, while the number of predefined prior boxes in our algorithm is also fewer than in Gliding Vertex. Our algorithm outperforms the compared two-stage object

detection algorithms in the table by a significant margin. When comparing with the one-stage object detection algorithms in the table, we also achieve a 0.12% improvement in accuracy compared to the highest-performing S^2A-Net.



(a) Scale Disparity

(b) Complex Background

Figure 16. Detection results obtained from the HRSC2016 dataset.

Furthermore, we compared our algorithm with representative one-stage object detection algorithms such as YOLOv6, YOLOv7, and YOLOv8. Our algorithm outperforms YOLOv6, YOLOv7, and YOLOv8 by 4.87%, 4.18%, and 4.59% in terms of accuracy improvement, respectively, indicating that their direct application to HSRIs is not effective. The YOLO series algorithms are primarily developed for conventional datasets, while HSRIs present greater challenges due to large object aspect ratios, complex backgrounds, and frequent object clustering. Therefore, conventional horizontal box object detection algorithms like the YOLO series cannot achieve the desired results when they are applied to HSRIs.

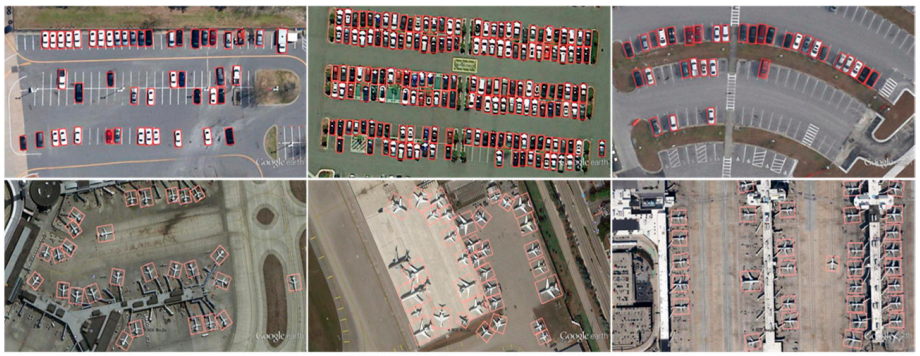
It is evident that the proposed algorithm demonstrates impressive detection performance when dealing with objects that exhibit significant scale variations when the detection results depicted in Figure 16 are examined. This observation highlights the effectiveness of integrating the RepConv, Transformer Encoder, and BiFPN modules into the algorithm. The algorithm also demonstrates efficient and accurate detection capabilities for objects in complex backgrounds, highlighting the effectiveness of utilizing the GAM and SIoU modules. In light of the aforementioned analysis, it can be inferred that our algorithm exhibits strong performance in detecting objects across a wide range of scales or in complex backgrounds. This provides validation for the effectiveness of the proposed approach.

4.4.2. Analysis of the Experimental Results on UCAS-AOD

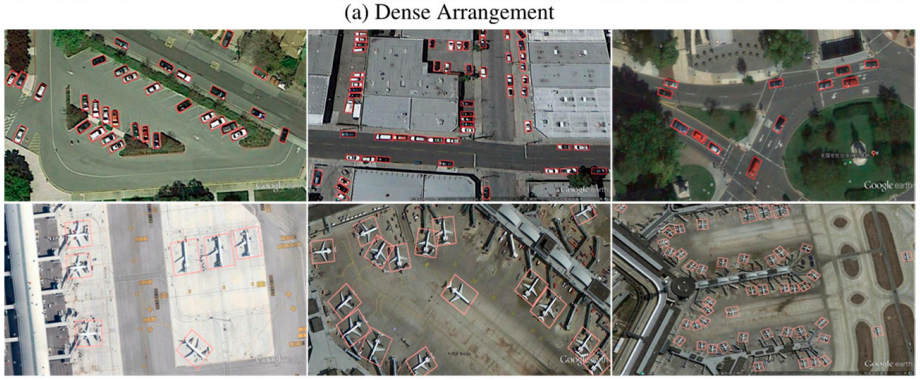
Experiments were conducted to compare our algorithm with the latest OD algorithms. The precision comparison results for each algorithm are displayed in Table 2. The partial detection results of our algorithm on the UCAS-AOD dataset are depicted in Figure 17.

Table 2. Comparison of accuracy of different OD algorithms on the UCAS-AOD dataset.

Algorithms	Car (%)	Airplane (%)	mAP (%)
YOLOv3-O [21]	74.63	89.52	82.08
RetinaNet-O [14]	84.64	90.51	87.57
Faster R-CNN-O [6]	86.87	89.86	88.36
RoITrans [10]	87.99	89.90	88.95
DAL [15]	89.25	90.49	89.87
YOLOv6 [26]	88.96	90.46	89.71
YOLOv7 [27]	89.05	90.42	89.73
YOLOv8	89.28	90.45	89.87
Ours	**89.60**	**90.53**	**90.06**



(a) Dense Arrangement



(b) Uncertain Direction

Figure 17. Detection results obtained from the UCAS-AOD dataset.

According to Table 2, the detection accuracy of our algorithm for Car and Airplane reaches 89.60% and 90.53%, respectively, with an overall mAP of 90.06%, which is higher than all the compared algorithms. The detection results depicted in Figure 17 demonstrate the strong performance of our algorithm in detecting densely arranged objects, confirming the effectiveness of the GAM and SIoU modules introduced in this paper. It also exhibits efficient and accurate detection capability for objects with uncertain orientations, confirming the effectiveness of the circular smooth label approach for handling angle-related issues. According to the aforementioned analysis, it can be inferred that our algorithm exhibits strong detection performance for densely arranged objects and objects with uncertain orientations. This outcome serves as evidence for the effectiveness of the proposed approach.

Furthermore, we compared our algorithm with YOLOv6, YOLOv7, and YOLOv8 in terms of accuracy. YOLOv8 achieves the highest detection accuracy of 89.28% for the "Car" category, while YOLOv6 achieves the highest detection accuracy of 90.46% for the "Airplane" category. In comparison, our algorithm demonstrates accuracy improvements of 0.32% and 0.07%, respectively, providing further evidence of the effectiveness of our algorithm compared to YOLOv6, YOLOv7, and YOLOv8.

4.5. Ablation Experiments

To evaluate the rationality and effectiveness of the recently incorporated functional modules in our OD algorithm, we perform ablation experiments on the two experimental datasets. Tables 3 and 4 display the experimental findings. On the HRSC2016 dataset, the baseline model YOLOv5s achieved an mAP of 88.57%. As shown in Table 3, the introduction of the SIoU, GAM, Transformer Encoder, BiFPN, RepConv, and IDetect modules resulted in mAP improvements of 0.46%, 0.52%, 0.25%, 0.27%, 0.05%, and 0.17%, respectively. The proposed algorithm achieved an mAP of 90.29%.

Table 3. The recognition accuracy changes with the increase in modules in the HRSC2016 dataset.

	Different Variants					
SIoU	√	√	√	√	√	√
GAM		√	√	√	√	√
Transformer Encoder			√	√	√	√
BiFPN				√	√	√
RepConv					√	√
IDetect						√
mAP(%)	89.03	89.55	89.80	90.07	90.12	90.29

Table 4. The recognition accuracy changes with the increase in modules in the UCAS-AOD dataset.

	Different Variants					
SIoU	√	√	√	√	√	√
GAM		√	√	√	√	√
Transformer Encoder			√	√	√	√
BiFPN				√	√	√
RepConv					√	√
IDetect						√
mAP(%)	87.32	87.93	88.60	89.44	89.91	90.06

The baseline model YOLOv5s achieved an mAP of 86.90% on the UCAS-AOD dataset. Similarly, as shown in Table 4, the introduction of the SIoU, GAM, Transformer Encoder, BiFPN, RepConv, and IDetect modules resulted in mAP improvements of 0.42%, 0.61%, 0.67%, 0.84%, 0.47%, and 0.15%, respectively. The proposed algorithm achieved an mAP of 90.06%. By analyzing both Tables 3 and 4, the positive impact of each newly introduced functional module in our algorithm on improving the accuracy of object recognition in HSRIs can be observed.

5. Conclusions and Future Works

OD in HSRIs encounters various challenges due to the intricate nature of the earth's surface and the specific shooting distances and angles involved. The neglect of specific characteristics of HSRIs often leads to the failure of conventional OD algorithms to meet application requirements. In response to this, the present paper introduces an enhanced OD algorithm based on YOLOv5 specifically designed for HSRIs. By incorporating multiple functional modules, this algorithm preserves the strong OD capability of the original YOLOv5 while significantly improving the accuracy in detecting objects with diverse scales, complex backgrounds, dense arrangements, and uncertain orientations within HSRIs. By demonstrating a high detection accuracy on the two experimental HSRI datasets, the proposed algorithm's effectiveness is validated. Nonetheless, the algorithm continues to experience instances of overlooking small objects and exhibits a comparatively lengthy processing time.

To tackle the problem of missed detections for certain small objects in this study, our forthcoming efforts will concentrate on enhancing the precision of small object detection. This will be accomplished by employing multi-scale detection, improving feature representation, and implementing techniques such as data augmentation and sample balancing. Meanwhile, it is crucial to invest efforts in the development of OD algorithms for HSRIs that are both fast and accurate. Our plan entails exploring pruning and distillation techniques to not only optimize model performance and achieve exceptional results but also to minimize processing time. Conducting research on faster and more accurate OD algorithms can effectively cater to the requirements of real-world applications that involve extensive HSRI datasets.

Author Contributions: Conceptualization, F.C. and J.L.; methodology, B.X., D.L. and Y.Q.; software, B.X. and H.B.; validation, F.C., J.L. and C.Z.; formal analysis, H.B. and H.Z.; investigation, F.C. and B.X; resources, B.X. and H.Z.; data curation, B.X.; writing—original draft preparation, F.C., B.X. and H.B.; writing—review and editing, F.C, J.L., D.L., C.Z. and B.X.; visualization, B.X.; supervision, J.L. and D.L.; project administration, F.C., J.L. and D.L.; funding acquisition, F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Natural Science Foundation of China, grant numbers 62072291, 42071316, 62072294, 62272284, 61672332, 62176145 and 41871286; the Special Fund for Science and Technology Innovation Teams of Shanxi, grant number 202204051001015.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the editors and the anonymous reviewers for their valuable comments for greatly improving our manuscript.

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

References

1. Li, J.; Liu, H.; Du, J.; Cao, B.; Zhang, Y.; Yu, W.; Zhang, W.; Zheng, Z.; Wang, Y.; Sun, Y.; et al. Detection of Smoke from Straw Burning Using Sentinel-2 Satellite Data and an Improved YOLOv5s Algorithm. *Remote Sens.* **2023**, *15*, 2641. [CrossRef]
2. Qu, J.; Tang, Z.; Zhang, L.; Zhang, Y.; Zhang, Z. Remote Sensing Small Object Detection Network Based on Attention Me-chanism and Multi-Scale Feature Fusion. *Remote Sens.* **2023**, *15*, 2728. [CrossRef]
3. Yu, N.; Ren, H.; Deng, T.; Fan, X. A Lightweight Radar Ship Detection Framework with Hybrid Attentions. *Remote Sens.* **2023**, *15*, 2743. [CrossRef]
4. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
7. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single Shot Multiboot Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
8. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
9. 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.
10. Ding, J.; Xue, N.; Long, Y.; Xia, G.; Lu, Q. Learning Roi Transformer for Oriented Object Detection in Aerial Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2849–2858.
11. Zhang, Z.; Guo, W.; Zhu, S.; Yu, W. Toward Arbitrary-oriented Ship Detection with Rotated Region Proposal and Discrimination Networks. *IEEE Geosci. Remote Sens.* **2018**, *15*, 1745–1749. [CrossRef]
12. Yang, X.; Sun, H.; Fu, K.; Yang, J.; Sun, X.; Yan, M.; Guo, Z. Automatic Ship Detection in Remote Sensing Images from Google Earth of Complex Scenes based on Multiscale Rotation Dense Feature Pyramid Networks. *Remote Sens.* **2018**, *10*, 132. [CrossRef]
13. Azimi, S.M.; Vig, E.; Bahmanyar, R.; Körner, M.; Reinartz, P. Towards Multi-class Object Detection in Unconstrained Remote Sensing Imagery. In Proceedings of the 14th Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; pp. 150–165.
14. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

15. Ming, Q.; Zhou, Z.; Miao, L.; Zhang, H.; Li, L. Dynamic Anchor Learning for Arbitrary-oriented Object Detection. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 2355–2363. [CrossRef]
16. Qian, W.; Yang, X.; Peng, S.; Yan, J.; Guo, Y. Learning Modulated Loss for Rotated Object Detection. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 2458–2466. [CrossRef]
17. Yang, X.; Yan, J.; Feng, Z.; He, T. R3det: Refined Single-stage Detector with Feature Refinement for Rotating Object. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 3163–3171. [CrossRef]
18. Han, J.; Ding, J.; Li, J.; Xia, G.; Sensing, R. Align Deep Features for Oriented Object Detection. *IEEE Trans. Geosci Remote Sens.* **2021**, *60*, 1–11. [CrossRef]
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
20. Choi, J.; Chun, D.; Kim, H.; Lee, H.-J. Gaussian Yolov3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 502–511.
21. Redmon, J.; Farhadi, A. Yolov3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
22. Wang, D.; Liu, Z.; Gu, X.; Wu, W.; Chen, Y.; Wang, L. Automatic Detection of Pothole Distress in Asphalt Pavement Using Improved Convolutional Neural Networks. *Remote Sens.* **2022**, *14*, 3892. [CrossRef]
23. Wu, W.; Liu, H.; Li, L.; Long, Y.; Wang, X.; Wang, Z.; Chang, Y. Application of Local Fully Convolutional Neural Network Combined with YOLOv5 Algorithm in Small Target Detection of Remote Sensing Image. *PLoS ONE* **2021**, *16*, e0259283. [CrossRef]
24. Zhang, Y.; Guo, Z.; Wu, J.; Tian, Y.; Tang, H.; Guo, X. Real-time Vehicle Detection Based on Improved Yolov5. *Sustainability* **2022**, *14*, 12274. [CrossRef]
25. Zhao, Q.; Liu, B.; Lyu, S.; Wang, C.; Zhang, H. TPH-YOLOv5++: Boosting Object Detection on Drone-captured Scenarios with Cross-layer Asymmetric Transformer. *Remote Sens.* **2023**, *15*, 1687. [CrossRef]
26. Li, C.; Li, L.; Geng, Y.; Jiang, H.; Cheng, M.; Zhang, B.; Ke, Z.; Xu, X.; Chu, X. Yolov6 v3.0: A Full-scale Reloading. *arXiv* **2023**, arXiv:2301.05586.
27. Wang, C.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, Canada, 18–22 June 2023; pp. 7464–7475.
28. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. Repvgg: Making Vgg-style Convnets Great Again. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
29. Liu, Y.; Shao, Z.; Hoffmann, N. Global Attention Mechanism: Retain Information to Enhance Channel-spatial Interactions. *arXiv* **2021**, arXiv:2112.05561.
30. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586.
31. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and Better Learning for Bounding Box Regression. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12993–13000. [CrossRef]
32. Gevorgyan, Z. SIoU loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
33. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
34. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
35. Ghiasi, G.; Lin, T.-Y.; Le, Q.V. Nas-fpn: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7036–7045.
36. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and Efficient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
37. Yang, X.; Yan, J. Arbitrary-oriented Object Detection with Circular Smooth Label. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 677–694.
38. 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 6th International Conference on Pattern Recognition Applications and Methods, Porto, Portugal, 24–26 February 2017; pp. 324–331.
39. 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]
40. Xu, Y.; Fu, M.; Wang, Q.; Wang, Y.; Chen, K.; Xia, G.; Bai, X. Gliding Vertex on the Horizontal Bounding Box for Multi-Oriented Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 1452–1459. [CrossRef]
41. Liao, M.; Zhu, Z.; Shi, B.; Xia, G.; Bai, X. Rotation-sensitive Regression for Oriented Scene Text Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5909–5918.
42. 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; pp. 195–211.

43. Yang, X.; Hou, L.; Zhou, Y.; Wang, W.; Yan, J. Dense Label Encoding for Boundary Discontinuity Free Rotation Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15819–15829.
44. Yi, J.; Wu, P.; Liu, B.; Huang, Q.; Qu, H.; Metaxas, D. Oriented Object Detection in Aerial Images with Box Boundary-Aware Vectors. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 2150–2159.