remote sensing

MDPI

Article

LR-TSDet: Towards Tiny Ship Detection in Low-Resolution Remote Sensing Images

Jixiang Wu [1,2,3], Zongxu Pan [1,2,3,*], Bin Lei [1,2,3] and Yuxin Hu [1,2,3]

[1] Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China; wujixiang19@mails.ucas.ac.cn (J.W.); leibin@mail.ie.ac.cn (B.L.); yxhu@mail.ie.ac.cn (Y.H.)
[2] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
[3] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: zxpan@mail.ie.ac.cn; Tel.: +86-010-58887208

Abstract: Recently, deep learning-based methods have made great improvements in object detection in remote sensing images (RSIs). However, detecting tiny objects in low-resolution images is still challenging. The features of these objects are not distinguishable enough due to their tiny size and confusing backgrounds and can be easily lost as the network deepens or downsamples. To address these issues, we propose an effective Tiny Ship Detector for Low-Resolution RSIs, abbreviated as LR-TSDet, consisting of three key components: a filtered feature aggregation (FFA) module, a hierarchical-atrous spatial pyramid (HASP) module, and an IoU-Joint loss. The FFA module captures long-range dependencies by calculating the similarity matrix so as to strengthen the responses of instances. The HASP module obtains deep semantic information while maintaining the resolution of feature maps by aggregating four parallel hierarchical-atrous convolution blocks of different dilation rates. The IoU-Joint loss is proposed to alleviate the inconsistency between classification and regression tasks, and guides the network to focus on samples that have both high localization accuracy and high confidence. Furthermore, we introduce a new dataset called GF1-LRSD collected from the Gaofen–1 satellite for tiny ship detection in low-resolution RSIs. The resolution of images is 16m and the mean size of objects is about 10.9 pixels, which are much smaller than public RSI datasets. Extensive experiments on GF1-LRSD and DOTA-Ship show that our method outperforms several competitors, proving its effectiveness and generality.

Keywords: tiny ship detection; remote sensing images (RSIs); convolutional neural network (CNN); self-attention; atrous convolution

Citation: Wu, J.; Pan, Z.; Lei, B.; Hu, Y. LR-TSDet: Towards Tiny Ship Detection in Low-Resolution Remote Sensing Images. *Remote Sens.* **2021**, *13*, 3890. https://doi.org/10.3390/rs13193890

Academic Editor: Peter Hofmann

Received: 16 August 2021
Accepted: 23 September 2021
Published: 28 September 2021

1. Introduction

Object detection [1–3] in remote sensing images (RSIs) aims to locate objects of interest (e.g., ships [4,5], airplanes [6,7] and storage tanks [8,9]) and identify corresponding categories, playing an important role in urban planning, automatic monitoring, geographic information system (GIS) updating, etc. With the rapid development and large-scale application of earth observation technologies, the RSIs obtained from satellites have become increasingly diversified, and the amount of RSIs has greatly increased. Among them, the very high-resolution (VHR) RSIs provide abundant spatial and textural information regarding their targets, and are widely used in target extraction and recognition [10], landcover classification [11], etc. The low-resolution RSIs tend to have a large field of view and contain more targets than VHR images of the same size, therefore attracting much attention in object detection [4,12,13] and tracking [14] tasks.

However, due to the limitations of low-resolution images, objects in low-resolution RSIs only occupy a few pixels (e.g., ships of 8 pixels) which are much smaller than normal, making it difficult to extract sufficient information. Moreover, in real-world scenarios, the

quality of RSIs is always affected by the imaging conditions (e.g., illumination and clouds) and the characteristics of the sensors. These distractors make the image background more complicated, further increasing the difficulty of detection. Thus, detecting tiny objects in low-resolution RSIs is still a uniquely challenging task.

In general, we define the type of objects as follows: tiny objects are <16 pixels, small objects are 16~32 pixels, medium objects are 32~96 pixels and large objects are >96 pixels. Much research has been conducted to improve the performance of object detection, which can be roughly divided into traditional methods and deep learning-based methods. Specifically, traditional approaches mostly rely on prior information and handcrafted features to extract and classify regions of interest. Taking ship detection as an example, Gang et al. [15] presented a harbor-based method based on the assumption that the harbor layout is relatively stable. This method uses geographic information template matching technology to complete sea-land segmentation. Xu et al. [16] utilized a special threshold for segmentation, because the gray values and distributions of sea and land are usually different. Geometric features (e.g., aspect ratios and edge contours) and statistical features (e.g., HoG [17], LBP [18] and SIFT [19]) are adopted to represent candidate reigons, and classifiers (e.g., SVM [20] and AdaBoost [21]) are exploited to distinguish them.

With the release of large-scale datasets (e.g., ImageNet [22], PASCAL VOC [23] and MSCOCO [24]) and the benefits of the great power of feature representation capabilities of convolutional neural networks (CNN), deep learning-based methods, such as Faster-RCNN [25] and RetinaNet [26], have become mainstream in the nature image community. Likewise, researchers have also built remote sensing benchmarks [2,3,27,28], such as HRSC2016 [28], DOTA [2] and DIOR [3], successfully transferring deep learning-based object detection methods into the remote sensing community [29–32]. For example, Pang et al. [13] proposed $\mathcal{R}^2$-CNN for real-time detection with a lightweight backbone, Tiny-Net. Ding et al. [29] introduced RoI Transformer, which converts horizontal RoIs to rotated RoIs and extracts rotation-invariant features. R^3Det [30] added the feature refine module between two cascaded detection heads to further encode rotated position information.

Nevertheless, most existing methods focus on general object detection (e.g., scale variations and oriented bounding box regression) while ignoring the poor performance and special demands of tiny object detection. Meanwhile, the available remote sensing datasets are not perfectly suitable for tiny object detection in many aspects. As shown in Figure 1b,c, the resolution of most images is much higher (e.g., 0.20 m), and most labeled objects are larger than 32×32 pixels, which are defined as medium targets in MS COCO [24].

In this article, we seek to solve the remaining problems of detecting tiny ships in low-resolution RSIs. To this end, we propose a novel tiny ship detection framework called LR-TSDet. The objects are always surrounded by various backgrounds due to their small-scale characteristics, which will affect the feature expression. However, the background information can also provide certain indicative information for target identification. Therefore, we utilize global contextual information to capture the correlation between backgrounds and objects, thereby enhancing responses of objects in the feature maps. We propose the filtered feature aggregation (FFA) module to make use of complex backgrounds, which can be plugged into the feature pyramid network (FPN) [33]. As a self-attention mechanism [34], FFA calculates a similarity matrix to suppress background noise and strengthen features of objects. Secondly, we can only obtain limited information from low-resolution images, because the tiny objects can easily disappear due to the downsampling of the network. Thus, we have designed a hierarchical-atrous spatial pyramid (HASP) module to obtain deep semantic information while avoiding network downsampling. We reconstruct an enhanced atrous convolution layer [35] called hierarchical-atrous convolution block (HACB) using group convolution and hierarchical residual connection [36]. The HASP module aggregates four parallel HACBs, wherein each HACB represents different receptive fields, thereby enriching the semantic information of feature maps. Thirdly, in order to tackle the inconsistency between classification and regression subnets, we propose

the IoU-Joint loss to guide the training of a classification network, in which the labels used to mark samples are replaced with an IoU score, inspired by [37]. The IoU is defined as the coincidence quality between the predicted bounding box and the ground-truth box. In this way, the network would prefer to predict high classification scores for positive samples with high IoU scores, thereby further improving the sample quality and localization accuracy. Furthermore, we have developed a new dataset named GF1-LRSD for the evaluation of tiny ship detection in low-resolution RSIs. It contains 4406 images with a resolution of 16 m and 7172 labeled instances, of which the mean size is 10.9 pixels; Figure 1 displays some samples. Our main contributions are summarized as follows.

- An effective detector, LR-TSDet, is proposed to achieve tiny ship detection in low-resolution RSIs; this detector is equipped with a filtered feature aggregation (FFA) module, a hierarchical-atrous spatial pyramid (HASP) module and the IoU-Joint loss.
- The FFA module is plugged into the FPN, which aims to suppress the interference of redundant background noise and highlight the response of regions of interest by learning global context information.
- The HASP module is designed to extract multi-scale local semantic information through aggregating features with different receptive fields.
- The IoU-Joint loss utilizes the IoU score to jointly optimize the classification and regression subnets, further refining the multi-task training process.
- Extensive experiments on our built datasets, GF1-LRSD and DOTA-Ship, validate the performance of our proposed method, which outperforms other comparison methods by a large margin.

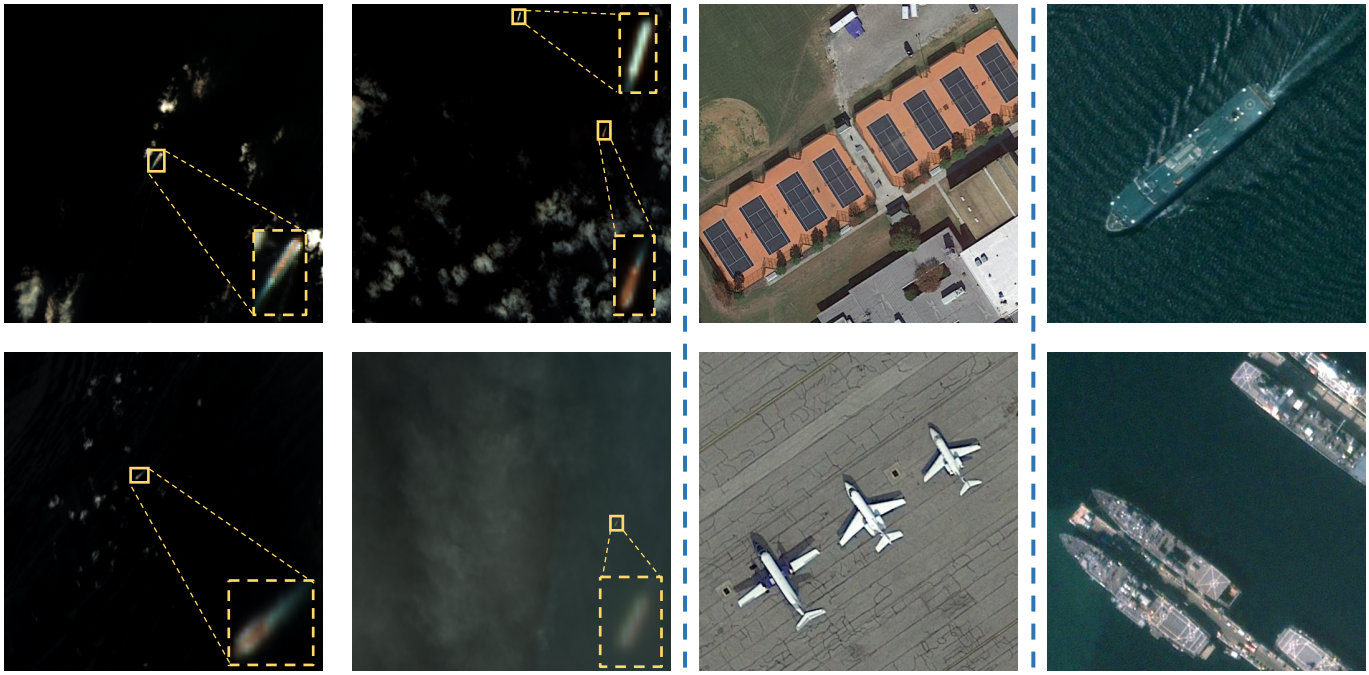


Figure 1. Image samples of different datasets, including (**a**) GF1-LRSD, (**b**) DOTA and (**c**) HRSC2016. For fair comparison, images have been cropped to the same size of 512 × 512. As can be seen, the objects in GF1-LRSD are significantly smaller than others.

The rest of this article is organized as follows. Section 2 briefly introduces the related works. Section 3 illustrates the proposed tiny ship detector, LR-TSDet, in detail, including the structure of each module, the design of the loss function, etc. In Section 4, we first describe the construction process and statistics of the collected dataset, and then present

experimental results and discussions, respectively. Finally, conclusions are drawn in Section 5.

2. Related Work

2.1. Object Detection in Remote Sensing Images

With the application of deep learning-based methods, we have witnessed the rapid development of object detection in remote sensing images in the past few years. Xia et al. [2] introduced a large-scale dataset named DOTA, which has gradually developed into a benchmark to evaluate the performance of various algorithms. Li et al. [3] built a more comprehensive dataset on both object categories and amount of images, further promoting the research of remote sensing. Generally, the CNN-based object detection methods can be approximately divided into two categories: anchor-based methods and anchor-free methods. The anchor-based methods [25,26,38–41] utilize preset anchor boxes with different scales and aspect ratios to match and locate objects. YOLT [12] inherited and fine-tuned the YOLO network [40] and partitions large-scale images into slices for rapid detection. Zhang et al. [42] presented the CAD-Net, which revealed a special relationship between the background and object by capturing global and local contextual information. Furthermore, many studies have been performed to encode rotated features better, such as those on RoI transformer [29], DRBox-v2 [43], GWD Loss [44], Gliding vertex [45], S^2A-Net [46], etc. The anchor-free methods [47–50] have been given more attention recently. These methods cancel all kinds of hyperparameters of anchors and provide a more concise pipeline for detection. For example, Wei et al. [51] proposed O^2DNet, which encodes oriented objects as pairs of middle lines. Other models [52–54] have also been adopted using anchor-free strategies.

2.2. Tiny Object Detection

Researchers have attempted to alleviate the problem of tiny object detection from all aspects, including data augmentation [55], image pyramids [56] and super-resolution [57]. Yu et al. [58] proposed the Scale Match (SM) strategy, which aligns the scale distribution of a used dataset to be consistent with the pre-training dataset. SCRDet [59] obtained features of small objects by a tailored feature fusion structure. Hu et al. [60] found tiny faces by utilizing the contextual information around objects. In our work, we first exploit the FFA module to highlight useful features of tiny objects by capturing the mutual information between each pixel in the feature map. It can be observed that the background could indicate categories or locations of candidate objects, e.g., ships usually sail in the ocean. Furthermore, we apply improved atrous convolutions with different receptive fields to gather the features and capture deeper semantic information.

3. Methods

In this section, we first give an overview of the proposed network, LR-TSDet, for tiny ship detection, and show how it works. Next, we detail the design of the filtered feature aggregation (FFA) module for noise suppression and feature enhancement. Then, the hierarchical-atrous spatial pyramid (HASP) module is introduced to acquire larger receptive fields. Finally, we elaborate on the IoU-Joint loss function for high-precision detection.

3.1. Overview

Figure 2 illustrates the details of the proposed LR-TSDet. We adopted the one-stage detector *RetinaNet* [26] as the baseline, which is a widely used anchor-based detector. Given an input image, we fed it into a backbone network to extract multi-scale features, which can usually take different forms of CNNs from existing detectors, such as ResNet [61], EfficientNet [62], Swin-Transformer [63], etc. Taking ResNet [61] as an example, different residual stages represent hierarchical semantic information. Therefore, we applied the feature pyramid network (FPN) [33] to construct a multi-scale convolutional feature pyramid

with a top-down pathway and lateral connections. Finally, each FPN level was followed by a detection head, which included two different branches, named the classification subnet and box regression subnet. These two subnets are small fully convolutional networks (FCN) [64] with four stacked 3 × 3 convolution layers for predicting the probability and location of the object, respectively.

Different from RetinaNet, we constructed the pyramid from P_3 to P_5 using {C_3, C_4, C_5} in ResNet, where P_l and C_l indicate the pyramid level and residual stage, respectively (l means the feature map resolution is 2^l lower than the input). ResNet50 pre-trained by *ImageNet* [22] was used as a backbone network. In our network design, we adopted a filtered feature aggregation (FFA) module in lateral connections to improve the quality of feature maps produced by FPN. In order to capture deeper semantic information better, we presented the hierarchical-atrous spatial pyramid (HASP) module before the detection heads, which uses dilated convolution [35] to obtain multiple receptive fields with different dilation rates while maintaining the spatial resolution of the features.

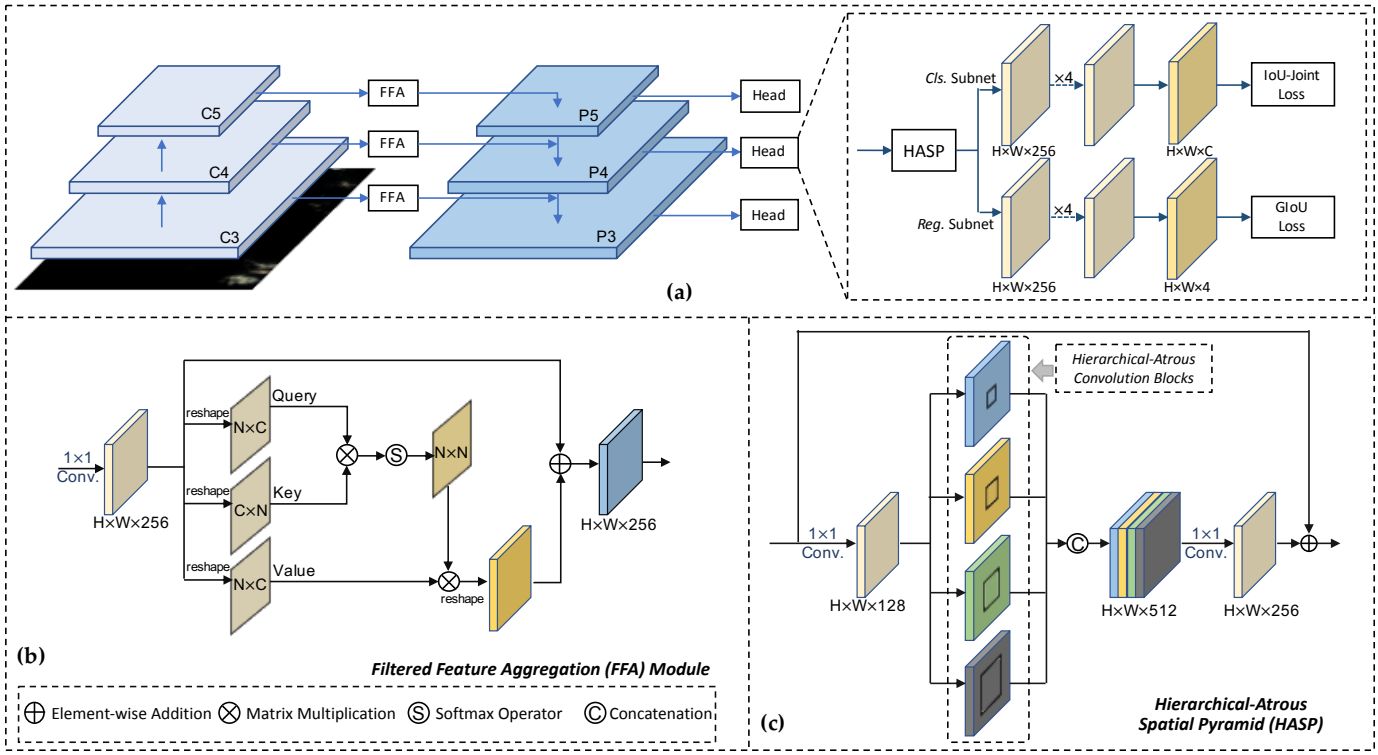


Figure 2. The network architecture of LR-TSDet. It consists of a backbone network, a feature pyramid network (FPN) [33] and multiple detection heads. The filtered feature aggregation (FFA) module is inserted between the backbone and FPN to enhance the capability of the top-down pathway. The detection head is appended to each FPN level, having a hierarchical-atrous spatial pyramid (HASP) and two subnets for object classification and box regression. (**a**) The pipeline of the network. (**b**) Filter Feature Aggregation (FFA) Module. (**c**) Hierarchical-Atrous Spatial Pyramid (HASP).

During training, the classification and regression losses were calculated by the defined loss function, and we applied the back-propagation algorithm to update network weights. We presented an IoU-Joint loss to evaluate the network classification ability better, which merges the detection confidence and intersection-over-union (IoU) between the predicted result and the ground truth as the class label. For model inference, our LR-TSDet is straightforward. An image is fed and passed through the network to obtain the final results. We employed the non-maximum suppression (NMS) strategy with a threshold of 0.6 for removing redundant detections.

3.2. Filtered Feature Aggregation (FFA) Module

Convolutional neural networks extract features through locally connected layer and weight sharing while ignoring the long-range dependencies. Meanwhile, the feature maps obtained by the backbone often come with some disadvantages, such as the error response of the non-object with object-like and ambiguous responses of the objects to be detected. Concretely, the tiny objects in low-resolution RSIs do not always have sufficient discriminative features due to limitations in size, which makes them easy to confuse with backgrounds and other distractors. From the perspective of human vision, we distinguish the objects with the help of their surrounding environment information, and this indicates that global information is helpful to detect tiny objects.

To address these issues, we introduced the Filtered Feature Aggregation (FFA) module, which helps to suppress background noise and capture global contextual information. As illustrated in Figure 2b, the FFA exploits the non-local block [34] as the main component. Given an input feature map $X \in \mathcal{R}^{H \times W \times C}$, where C, H and W denote the channel number, height and weight of the feature map, respectively, we first employed a 1×1 convolution layer to reduce channel dimensions to 256 (we set the channel $C = 256$ in all pyramid levels following [26]). Then, we transformed $\hat{X} \in \mathcal{R}^{H \times W \times 256}$ to three different embeddings, marked as Query ($\mathcal{Q} \in \mathcal{R}^{H \times W \times \hat{C}}$), key ($\mathcal{K} \in \mathcal{R}^{H \times W \times \hat{C}}$) and value ($\mathcal{V} \in \mathcal{R}^{H \times W \times \hat{C}}$), calculated as below:

$$\mathcal{Q} = W_{\mathcal{Q}}(\hat{X}), \quad \mathcal{K} = W_{\mathcal{K}}(\hat{X}), \quad \mathcal{V} = W_{\mathcal{V}}(\hat{X}) \tag{1}$$

where $\hat{C}$ is the channel number of the three embeddings, and $W_{\mathcal{Q}}$, $W_{\mathcal{K}}$ and $W_{\mathcal{V}}$ are weight matrices to be learned and implemented by different 1×1 convolution layers. Then, $\mathcal{Q}$, $\mathcal{K}$ and $\mathcal{V}$ were reshaped to size $\hat{C} \times N$, where $N = H \times W$ represents the number of the spatial pixels. Next, we computed the similarity matrix $\mathcal{S} \in \mathcal{R}^{N \times N}$ of $\mathcal{Q}$ and $\mathcal{K}$, which represents the relation between each pixel in the feature maps, formulated as follows:

$$\mathcal{S} = \mathcal{Q}^{\mathrm{T}} \otimes \mathcal{K} \tag{2}$$

where $\otimes$ denotes the matrix multiplication. Afterward, we obtained the spatial attention map by applying the softmax function, expressed as:

$$s_{ij} = \frac{\exp(s_{ij})}{\sum_{j=1}^{N} \exp(s_{ij})} \tag{3}$$

where s_{ij} represents the normalized pairwise relationship between position i and j. Thus, we computed the output matrix as follows:

$$\mathcal{O} = \mathcal{S} \otimes \mathcal{V}^{\mathrm{T}} \tag{4}$$

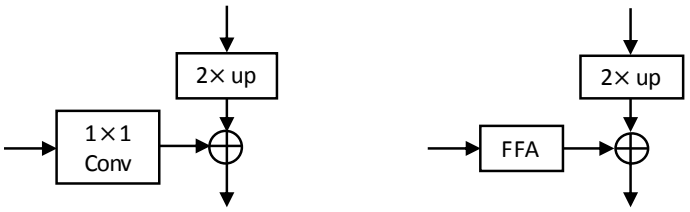
where $\mathcal{O} \in \mathcal{R}^{N \times C}$. Then, we reshaped $\mathcal{O}$ to the size $H \times W \times \hat{C}$, and a 1×1 convolution layer was employed to recover the initial dimension. Finally, we obtained the filtered feature map via a residual connection [61], calculated as follows:

$$Y = \mathcal{F}(\hat{X}) + \hat{X} \tag{5}$$

where $\mathcal{F}(\cdot)$ denotes the aforementioned self-attention mechanism [65].

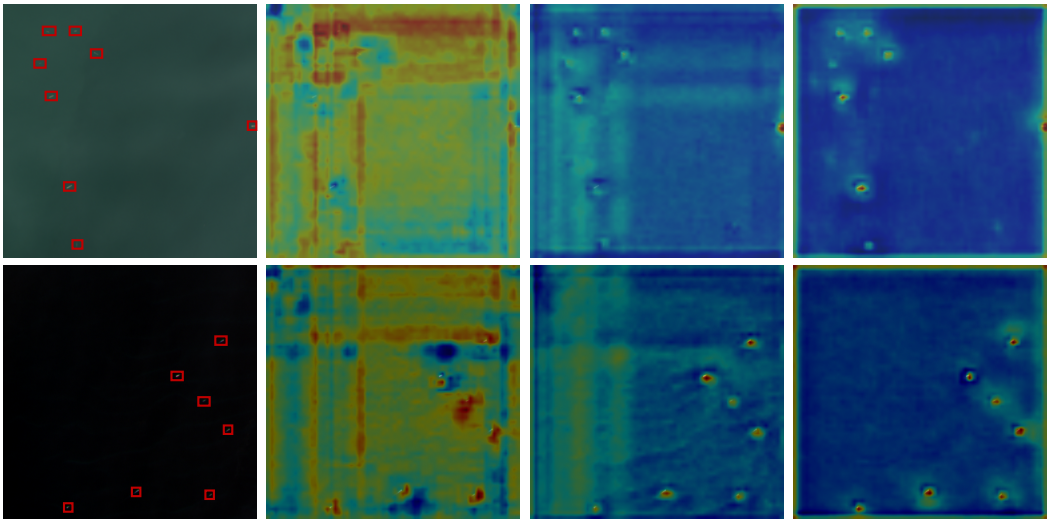
The FFA module leverages information from all locations to gain more discriminative feature representation, and we applied it to the top-down pathway, as shown in Figure 3. We replaced a 1×1 convolution layer with the FFA module to build a more robust FPN. The feature map was upsampled by a factor of 2 with bilinear interpolation. In particular, we visualized the feature maps of FPN after adopting the FFA module, as shown in Figure 4c. Figure 4b shows the original feature maps produced by RetinaNet. It can be observed that

the false responses of backgrounds are suppressed and the network focuses on the targets more choicely.



(**a**) Default top-down pathway. (**b**) Improved top-down pathway.

Figure 3. Comparison between the default top-down pathway and the improved method.



(a) (b) (c) (d)

Figure 4. The visualization of feature maps of FPN generated by different networks. (**a**) The input RGB images. The 'red boxes' in the images indicate the locations of the objects. (**b**) The original feature maps produced by RetinaNet. (**c**) The feature maps produced by adding the FFA module. (**d**) The feature maps produced by LR-TSDet.

3.3. Hierarchical-Atrous Spatial Pyramid (HASP)

Deeper networks usually require larger rates of downsampling to obtain richer semantic information. However, there is a trade-off between the scale of the object and the downsample rate. The tiny object may be lost in the feature map due to the decrease of spatial resolution. To this end, we propose the Hierarchical-Atrous Spatial Pyramid (HASP) module to mitigate this problem. Figure 2c describes the structure details.

An enhanced dilated convolution layer called Hierarchical-Atrous Convolution Block (HACB) was imported for stronger feature extraction capabilities. As shown in Figure 5, we replaced the standard convolution with the group convolution while connecting adjacent groups with residual connections. The HACB can capture deep semantic information in images from different depths, and the outputs of the current group were fed into the next group. Therefore, the equivalent receptive field increased consistently, and the module could integrate richer semantic information.

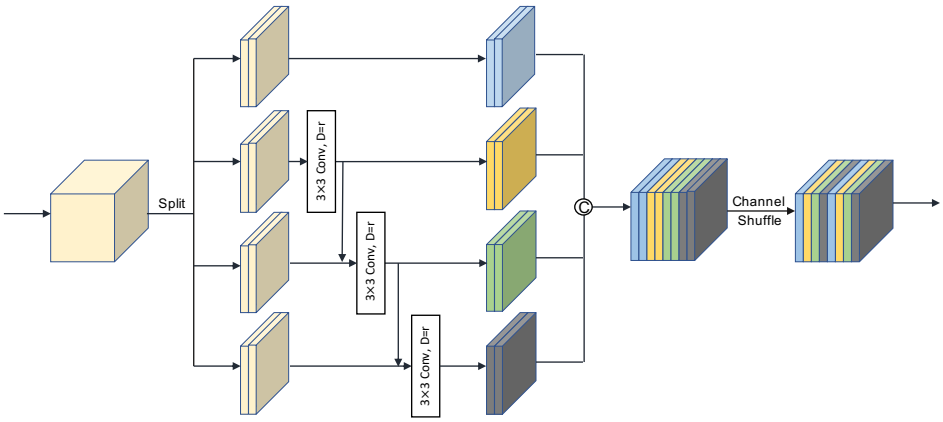


Figure 5. Illustration of the hierarchical-atrous convolution block (HACB). It adopts the group convolution and the hierarchical connection style to generate information, and the "channel shuffle" operator is used for information communication between different splits.

For a given feature map $\mathbf{X} \in \mathcal{R}^{C \times H \times W}$, the HACB first splits $\mathbf{X}$ into g groups, denoted by $\mathbf{x}_i \in \mathcal{R}^{\frac{C}{g} \times H \times W}$, where $i \in \{1, 2, \cdots, g\}$. Except for $\mathbf{x}_1$, each $\mathbf{x}_i$ is used to produce $\mathbf{y}_i$ through a 3×3 dilated convolution layer $\mathbf{D}_i(\cdot)$ with the same dilation rate r (shown in Figure 6). Specifically, if $i > 2$, the sub-feature $\mathbf{x}_i$ is first added with the output $\mathbf{y}_{i-1}$ and then fed into $\mathbf{D}_i(\cdot)$. Each $\mathbf{D}_i(\cdot)$ is followed by a group normalization (GN) layer [66] and a ReLU layer [67]. The implementation can be expressed as follows:

$$\mathbf{y}_i = \begin{cases} \mathbf{x}_i, & i = 1; \\ \text{ReLU}(\text{GN}(\mathbf{D}_i(\mathbf{x}_i))), & i = 2; \\ \text{ReLU}(\text{GN}(\mathbf{D}_i(\mathbf{x}_i + \mathbf{y}_{i-1}))), & 2 < i \leqslant g \end{cases} \tag{6}$$

Subsequently, all groups were aggregated by the concatenation operation, and the channel shuffle [68] operator was adopted for further information fusion, which can also be replaced with a simple 1×1 convolution layer for simplification. Notice that the feature information contained in each $\mathbf{y}_i$ is gradually enriched by the hierarchical residual connections. Meanwhile, the use of dilated convolution can retain more details without reducing the spatial resolution of the feature maps.

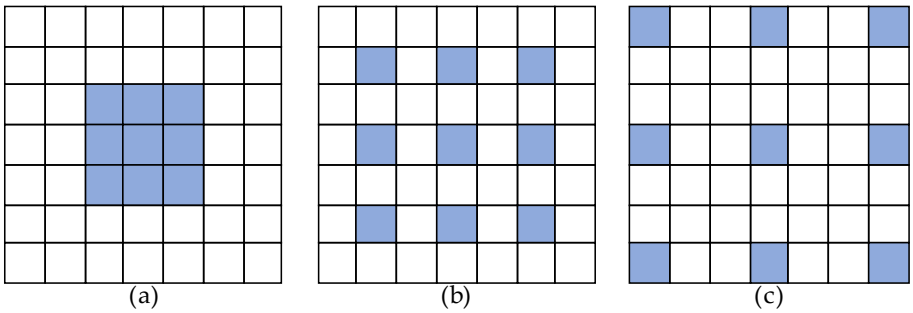


Figure 6. The dilated convolution with different dilation rates, representing the different receptive fields. (**a**) Rate = 1. (**b**) Rate = 2. (**c**) Rate = 3.

In the HASP design, we first used a 1×1 convolution layer to reduce the channel dimension for less computation. Then, four parallel HACBs with different dilation rates were applied to obtain multiple receptive fields. Next, the four branches were concatenated and passed through a 1×1 convolution layer, followed by a GN layer for adjusting the channel dimension. Finally, a skip-connection with an element-wise sum operator was utilized to gather the input and output for better information transmission.

3.4. Loss Function Design

In line with [25,26], our multi-task training loss function consists of two parts: the classification loss and the regression loss, formulated as follows:

$$\mathcal{L}_{total} = \frac{\lambda_1}{N_{pos}} \sum_{i=1}^{N} \mathcal{L}_{cls}(p_i, q_i^*) + \frac{\lambda_2}{N_{pos}} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{1}_{i,j}^* \mathcal{L}_{reg}(pb_i, gt_j^*) \tag{7}$$

where $\mathcal{L}_{total}$, $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$ denote the total training loss, classification loss and regression loss, respectively. N_{pos} is the number of positive samples, p_i represents the predicted probability value of the i-th anchor and q_i^* is the corresponding class "soft-label", which will be explained in the following subsection. pb_i is the i-th predicted bounding box and gt_j^* is the j-th ground-truth box corresponding to pb_i. $\mathbf{1}_{i,j}^*$ indicates the indicator function, being 1 for foreground and 0 for background, which means only positive samples contribute to the regression loss. The hyper-parameters $\{\lambda_1, \lambda_2\}$ are two balancing weights and are set to {1, 1} by default.

3.4.1. IoU-Joint Classification Loss

Most of the existing detectors adopt two independent subnets for classification and regression tasks. These two branches optimize their own loss function and are almost irrelevant. Before calculating the loss, we defined the positive and negative samples in the same way as most detectors, such as Faster-RCNN and RetinaNet (the IoU > 0.5 is for positive samples and the IoU < 0.4 is for negative samples; they stand for one of the two-stage and one-stage detectors, respectively). This division is coarse, ignoring the impact of IoU changes, where different IoUs indicate different overlaps with ground truths and the different features in use. To alleviate these inconsistencies, we propose the IoU-joint classification loss, which utilizes the IoU calculated by the regression subnet as an auxiliary object index.

To be specific, we replaced the standard one-hot category label with the localization quality (i.e., the IoU score). The label was softened to a continuous variable $q \in (0, 1)$, where $0 < q \leq 1$ indicates positive samples by IoU score and $q = 0$ is utilized for negative samples. In this way, each sample was weighted correspondingly, and the weight coefficient was directly correlated with the regression performance. Therefore, the network was guided more properly to suppress suboptimal results and predict detections having both high probability and high localization accuracy. Moreover, the discrete cross-entropy function $-\log(p)$ was needed to expand into a continuous form, written as $-q\log(p) + (1-q)\log(1-p)$. We defined the IoU-joint classification loss as:

$$\mathcal{L}_{cls}(p, q) = \begin{cases} -(1 - \log(1+pq))^{\beta}(q\log(p) + (1-q)\log(1-p)), & q > 0 \\ -p^{\beta}\log(1-p), & q = 0 \end{cases} \tag{8}$$

where p denotes the predicted probability of the object, and q is the localization quality score (IoU between the predicted box and ground truth). $(1 - \log(1+pq))^{\beta}$ is inherited from *Focal Loss* [26] as a modulating factor. We used the product of p and q to balance the contributions of samples, and the function `log` was used to smooth the decay of pq. When $q = 0$, the factor p^{β} would be adopted to scale the loss. The hyper-parameter β was set as 2 by default.

3.4.2. Bounding-Box Regression Loss

Similar to the anchor-based detectors [25,26,38], we needed to parameterize the coordinates of the bounding box, formulated as follows:

$$\begin{aligned} t_x &= (x - x_a)/w_a, & t_x^* &= (x^* - x_a)/w_a \\ t_y &= (y - y_a)/h_a, & t_y^* &= (y^* - y_a)/h_a \\ t_w &= \log(w/w_a), & t_w^* &= \log(w^*/w_a) \\ t_h &= \log(h/h_a), & t_h^* &= \log(h^*/h_a) \end{aligned} \tag{9}$$

where (x_a, y_a, w_a, h_a) represents the two coordinates of the box center, width and heigth of the anchor box. (x, y, w, h) and (x^*, y^*, w^*, h^*) represent the predicted box and ground-truth box, respectively.

The width and height of tiny objects in our dataset are generally about 10 pixels, and the smooth L_1 loss [38] is sensitive to scale variance, leading to difficulty in convergence. IoU evaluates the quality of the predicted box as a whole unit rather than four independent parameters, showing robustness to scale changes. Thus, we adopted the GIoU loss [69] for the bounding box regression. It is calculated as follows:

$$\mathcal{L}_{reg} = 1 - \text{GIoU}, \quad \text{GIoU} = \text{IoU} - \frac{A_c - \mathcal{U}}{A_c} \tag{10}$$

where A_c denotes the area of the smallest convex enclosing both the predicted box pb and the ground-truth box gt, and $\mathcal{U}$ is the area of union of pb and gt.

4. Experiments

In this section, we conduct different experiments to investigate the effectiveness of the proposed LR-TSDet. First, we introduce the datasets used in experiments. For the special demand of detecting tiny objects in low-resolution RSIs, we build a novel dataset called GF1-LRSD. The construction process and statistics of GF1-LRSD are also described. Furthermore, we build the DOTA-Ship dataset from DOTA-v1.5 [2] for further evaluation. Then, the implementation details and evaluation metrics are presented. Next, we conduct sufficient ablation experiments to evaluate the proposed modules. Finally, we compare the proposed method with other state-of-the-art (SOTA) methods and achieve the best performance. The implementation of this study will be publicly available after the article is accepted and the check procedure is completed.

4.1. Dataset

4.1.1. GF1-LRSD

Object detection in remote sensing images has made great progress with the help of open-source aerial images datasets, such as HRSC2016 [28], DOTA [2], DIOR [3], etc. Nonetheless, the image resolution in these datasets tends to be very high (e.g., 0.20 m, 1.07 m), and objects are always multi-scale. These characteristics are more suitable for evaluating general detection tasks rather than tiny object detection. There is still a lack of a reliable dataset that can meet the practical migration application. To this end, we built the GF1-LRSD dataset to promote the research of the problem.

Construction Process

1. Raw Data Acquisition and Preprocessing
   Gaofen–1 (GF–1) is an optical remote sensing satellite equipped with four 16 m resolution multispectral cameras which can obtain rich remote sensing images. Meanwhile, its complex imaging environment increases the difficulty of detection compared to other data. In order to build a sufficiently effective dataset, we collected a total of 145 wide–field–of–view (WFV) scenes of 1A level with a resolution of 16 m to filter the needed targets. The images with 12,000 × 12,000 pixels are 16-bit and have four bands (the extra is near-infrared), which are difficult to directly apply to the network. Figure 7 shows the detailed data processing flow. We converted the 16-bit data into 8-bit and cropped the large-scale images into a set of slices with the size of 512 × 512. Different from the regular sliding window mechanism, we directly cut the image without overlap for efficiency. As a result, nearly 83,520 sub-images were obtained. To enhance the contrast of images, we used the 2% truncated linear stretch method for quantification, calculated as follows:

$$\begin{aligned} R_{x,y,c} &= \left(I_{x,y,c} - T_{down}\right) \times \frac{255}{T_{up} - T_{down}} \\ R_{x,y,c} &= \min\left(\max\left(R_{x,y,c}, 0\right), 255\right) \end{aligned} \tag{11}$$

where $I_{x,y,c}$ and $R_{x,y,c}$ denote the pixel value at (x, y) in the c-th band of the input and output image, respectively. The $R_{x,y,c}$ is finally limited to 0∼255 to meet the standard format. T_{up} and T_{down} are the truncated upper and lower thresholds.

2. Image Annotation
We kept the data organization the same as PASCAL VOC [23] for convenience, wherein $(xmin, ymin, xmax, ymax)$ is used to describe the labeled bounding box. Let $(xmin, ymin)$ and $(xmax, ymax)$ denote the coordinates of the top-left and bottom-right corners of the bounding box, respectively. The toolbox LabelImg [70] was used to finish the annotation, and we used the horizontal rectangular box to locate the objects. After the identification and correction by experts, we collected, in total, 4406 images and 7172 labeled instances labeled as `ship`. For dataset splits, 3/5, 1/5, 1/5 of the images were used to form the `training set`, `validation set` and `test set`. Some samples are shown in Figure 1a.

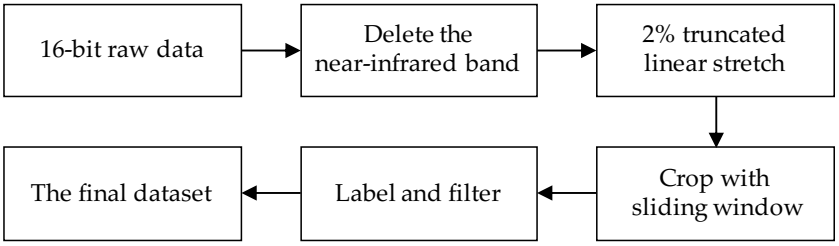


Figure 7. The flowchart of the dataset construction.

Dataset Statistics

In this subsection, the statistical characteristics of the proposed GF1-LRSD are analyzed and compared with other representative datasets. Specifically, we define the absolute size $S_a(\cdot)$ and relative size $S_r(\cdot)$ to describe the scales of instances, which can be formulated as follows [58]:

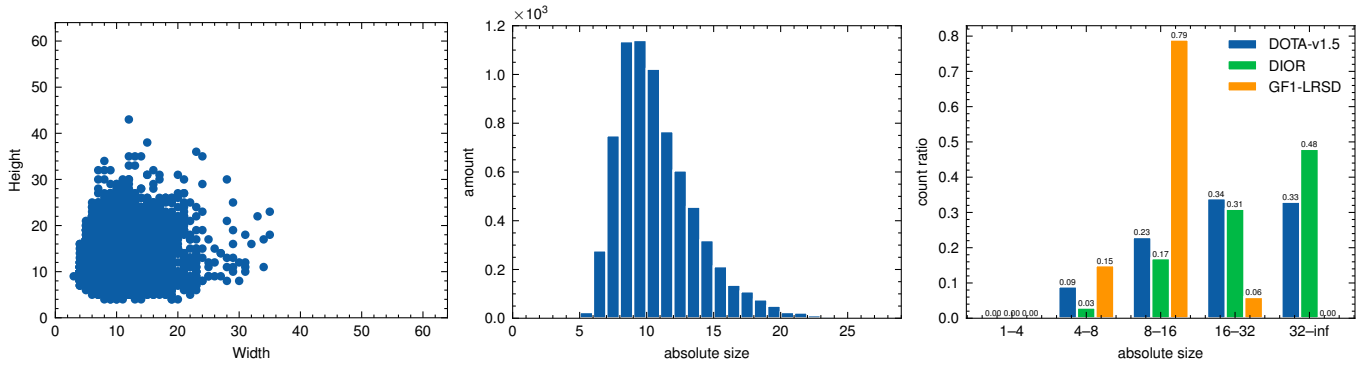
$$\begin{aligned} S_a(\mathcal{B}_{ij}) &= \sqrt{w_{ij} \times h_{ij}} \\ S_r(\mathcal{B}_{ij}) &= \sqrt{\frac{w_{ij} \times h_{ij}}{W_i \times H_i}} \end{aligned} \tag{12}$$

where $\mathcal{B}_{ij}$ represents the j-th instance's bounding box of the i-th image I_i in the dataset, and w_{ij}, h_{ij} are the width and height of $\mathcal{B}_{ij}$. W_i, H_i denote the width and height of I_i, respectively. The mean and standard deviation of the instance size for different datasets are shown in Table 1. The absolute size of 10.9 ± 3.0 pixels in GF1-LRSD is much smaller than the other datasets.

Table 1. Mean and standard deviation of instance size on different datasets.

Dataset	Absolute Size	Relative Size
DOTA-v1.0 `trainval`	55.3 ± 63.1	0.028 ± 0.034
DOTA-v1.5 `trainval`	34.0 ± 47.8	0.016 ± 0.026
DIOR	65.7 ± 91.8	0.082 ± 0.115
HRSC2016	140.6 ± 67.9	0.149 ± 0.072
GF1-LRSD	10.9 ± 3.0	0.021 ± 0.006

As shown in Figure 8c, most objects in GF1-LRSD are smaller than 16 pixels, accounting for about 94% of the objects, while more than 50% of the objects in other datasets have scales greater than 16 pixels, such as 79% of the objects in DIOR. Figure 8a,b further describe the main characteristics of GF1-LRSD. The width and height of the objects are mostly smaller than 25 and 30 pixels, respectively. The top 3 sizes are 9, 10 and 11 pixels.



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

Figure 8. Statistics of instances in GF1-LRSD and other datasets. (**a**) Scatter plot of width and height distribution in GF1-LRSD. (**b**) Histgram of scale distribution in GF1-LRSD. (**c**) Comparison of scale distribution between GF1-LRSD and other datasets.

4.1.2. DOTA-Ship

DOTA [2] is a large-scale dataset for object detection in remote sensing images. DOTA-v1.5 contains 2806 images and 403,318 instances of 16 object categories. It is an updated version of DOTA-v1.0, where the tiny instances (less than 10 pixels) are additionally annotated. To evaluate our LR-TSDet more accurately, we selected the objects labeled as `ship` and built a new dataset, named DOTA-Ship, which includes 573 images and 43,738 instances in total. DOTA-Ship was divided into training and test sets, consisting of 435 and 138 images, respectively. During training, we cropped the original images into 800×800 patches with an overlap of 200 pixels and subsequently ignored the sub-images that do not contain targets.

4.2. Implementation Details

We implemented LR-TSDet based on mmdetection [71], and the pre-trained ResNet-50 was adopted as the backbone network for all experiments. The models were trained for 100 epochs using the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.005. The learning rate was divided by a factor of 10 at the 70th and 90th epochs. The momentum and weight decay were set as 0.9 and 1×10^{-4}, respectively. We applied the linear warm-up strategy for the first 500 iterations with a ratio of 0.001 to stabilize the training process. The batch size was set as 8 on 2 RTX 2080Ti GPUs (4 images per GPU) and the random image flipping was adopted for data augmentation. The other hyper-parameter was the same as mmdetection unless specified.

4.3. Evaluation Metrics

In all experiments, the average precision (AP) was adopted to evaluate the model performance. We followed the setup proposed in the VOC2010 challenge [23] using a threshold of 0.5. The AP is formulated as follows:

$$AP = \int_0^1 p(r)dr \tag{13}$$

where r denotes different recalls, and $p(r)$ is the precision-recall (PR) curve. The AP was calculated as the area under the PR curve. In general, we defined the recall and precision as follows:

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

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

where TP, FP and FN represent the true positive, false positive and false negative samples. For a detection result, if the IoU between it and the ground truth was greater than the set threshold, then it was defined as TP; otherwise, it was defined as FP. If a ground-truth did not have a matched predicted result, it was defined as FN.

4.4. Ablation Studies

In this subsection, we conduct several ablation experiments to explore the effectiveness of the proposed method. We adopt ResNet50 as the backbone network and the `flops computation` tool in mmdetection is utilized to analyze model performance.

4.4.1. RetinaNet as Baseline

Before applying RetinaNet [26] as the baseline, we first modified the default model (called RetinaNet-D) to reduce useless operations. In the RetinaNet-D design, {P6, P7} in FPN are obtained by strided convolutions. This is done with the aim of improving large object detection, which may miss tiny object information and generate many unmatched negative samples that adversely affect the network training; our experiments also proved this. We removed the last two stages, {P6, P7}, and added a GN [66] layer in each detection head, named RetinaNet-B. The GN layer normalizes the data distribution by dividing the channels into groups and computing the mean and variance respectively. GN is a useful trick [47] and we applied it to stabilize the training process. The results are presented in Table 2. Compared with RetinaNet-D, our RetinaNet-B produced +0.75 AP gains with less computation complexity (51.57 G vs. 52.28 G), and the model size was reduced from 36.1 M to 30.8 M, indicating that the RetinaNet-D is more suitable as a baseline.

Table 2. Results of different RetinaNet architecture on GF1-LRSD. RetinaNet-D and RetinaNet-B denote the default network and the improved network, respectively. 'Head' means the detection subnet, including regression and classification branches.

Model	FPN	Head	AP (%)	FLOPs (G)	#Params (M)
RetinaNet-D	{P3, P4, P5, P6, P7}	-	79.00	52.28	36.1
RetinaNet-B	{P3, P4, P5}	+ GN	79.75	51.57	30.8

4.4.2. Individual Contributions of Each Component

In order to investigate the performance of design elements, we conducted a series of experiments with different combinations of modules. The quantitative results are reported in Table 3. We adopted RetinaNet-B (mentioned before) as the baseline, and it achieved an AP of 79.75, as shown in Experiment #1 in Table 3.

Table 3. Ablation studies on LR-TSDet. We adopted RetinaNet-B as the baseline and applied each module gradually to evaluate the effectiveness.

ID	FFA	HASP	IoU-Joint Loss	GIoU Loss	AP (%)
#1	-	-	-	-	79.75
#2	✓	-	-	-	81.04
#3	✓	✓	-	-	81.87
#4	✓	✓	✓	-	82.66
#5	✓	✓	-	✓	82.62
#6	✓	✓	✓	✓	83.87

- Efficacy of FFA. As discussed in the previous section, FFA processes the feature maps of each residual stage, aiming to suppress the non-object responses. We first applied FFA to the baseline, and it lead to a considerable gain of 1.29 AP, as shown in Experiment #2 in Table 3. FFA utilizes non-local blocks [34] to capture long-range dependencies, which is helpful to obtain global contextual information. Meanwhile, as a kind of attention mechanism, FFA could enhance the feature expression of targets and make it more discriminative.
- Influence of HASP. The purpose of designing HASP was to obtain more abundant semantic information while maintaining the resolution of feature maps. To verify its impact, we added HASP on the basis of Experiment #2, and the result is shown in Experiment #3 in Table 3. The detection AP was improved by 0.83 (81.04 $\rightarrow$ 81.87). HASP aggregates four parallel hierarchical-atrous convolution blocks (HACB) to obtain multi-scale information, where HACB consists of cascaded group dilation convolution. Furthermore, the integration of FFA and HASP further boosted the performance by 2.12, as shown in Experiment {#1, #3} in Table 3.
- Effect of the loss function. As analyzed above, our network is optimized by a multi-task loss, including the classification loss and regression loss. The default setting was focal loss [26] and smooth L_1 loss [38] in our experiments. As can be seen in Experiment {#3, #4, #5, #6} in Table 3, both losses contributed to the improvement of the final detection AP. By replacing focal loss with IoU-Joint loss, the network achieved an AP of 82.66, 0.79 higher than the default setting (81.87 vs. 82.66). The reason is that the IoU-Joint loss merges the localization score (i.e., IoU) into the calculation of classification loss, which could strengthen the connection between the two detection branches. Similarly, the performance was improved by 0.75 (81.87 $\rightarrow$ 82.62) by replacing smooth L_1 loss with GIoU loss. The GIoU loss treats the position information as a whole during training, which could result in more accurate training effects. It is worth mentioning that the combination of the two losses brought different degrees of improvement of the detection AP (82.66 $\rightarrow$ 83.87, 82.62 $\rightarrow$ 83.87). Finally, the LR-TSDet achieved the best performance of 83.87 AP, which outperformed the baseline by 4.12 AP, demonstrating the effectiveness of the proposed strategies.

4.4.3. Evaluation of HASP

In this subsection, we study the choice of the dilation rates and the utility of HACB architecture in HASP. It can be observed from Table 4 that as the dilation rate increases, the performance first increases and then decreases. We conjecture that excessive rates would incur the "gridding problem" [72], where the useful local information may be lost; thus we choose {2, 4, 6, 8} as the final parameters. Moreover, two sets of controlled experiments {#2, #3} and {#4, #5} in Table 4 prove the superiority of the HACB over the standard atrous convolution by adopting the hierarchical residual connection structure, where the HACB brings considerable gains of +0.60 and +0.56 AP under different settings of dilation rates, respectively.

Table 4. Detailed ablation studies on the dilation rates and the hierarchical-atrous convolution block (HACB) of HASP. The item "-" in the table indicates replacing HACB with standard atrous convolution.

ID	Dilations	HACB	AP (%)
#1	{1, 1, 1, 1}	✓	82.98
#2	{1, 2, 3, 4}	-	82.84
#3		✓	83.44
#4	{2, 4, 6, 8}	-	83.31
#5		✓	83.87
#6	{6, 6, 6, 6}	✓	83.15

4.5. Comparisons with Other Approaches

4.5.1. Experiments on LR-TSDet

To verify the performance of our proposed LR-TSDet, we compared it with other methods on the GF1-LRSD dataset, including two-stage detectors (e.g., Faster-RCNN [25] and SCRDet [59]), one-stage detectors (e.g., YOLOv3 [40], SSD [39] and R^3Det [30]) and anchor-free detectors (e.g., FCOS [47] and ATSS [73]). SCRDet and R^3Det are two typical methods for detecting tiny objects in remote sensing images. It should be noted that we kept all training settings the same, except the network backbone. As observed in Table 5, we achieved the best performance of 83.87 AP with a competitive model size and computation complexity. For example, our LR-TSDet outperformed Faster-RCNN by a large margin (+23.38 AP) with fewer FLOPs (54.67 G vs. 63.25 G) and parameters (32.53 M vs. 41.12 M), and the LR-TSDet surpassed FCOS by 9.71 AP with a slight increase in FLOPs and parameters. Qualitative detection results of LR-TSDet on GF1-LRSD are presented in Figure 9. The data were collected from real satellite imaging scenes, including the occlusion and interference of clouds, and the presence of vast land backgrounds. According to the detection results, our method works well under different conditions, proving its robustness. Figure 10 displays the P-R curves of the different approaches. The LR-TSDet is shown to locate objects more accurately with higher confidence.

Table 5. Comparisons with other typical methods on the GF1-LRSD dataset.

Method	Backbone	AP (%)	FLOPs (G)	#Params (M)
Faster-RCNN [25]	ResNet50	60.49	63.25	41.12
RetinaNet [26]	ResNet50	79.00	52.28	36.10
SSD512 [39]	VGG16	72.28	87.72	24.39
YOLOv3 [40]	DarkNet53	67.82	49.62	61.52
FCOS [47]	ResNet50	74.16	50.30	31.84
ATSS [73]	ResNet50	73.84	51.52	31.89
SCRDet [59]	ResNet50	69.29	-	-
R^3Det [30]	ResNet50	73.04	-	-
LR-TSDet (ours)	ResNet50	83.87	54.67	32.53

Furthermore, we evaluated the performance of our LR-TSDet under different scenarios, comparing it with RetinaNet. Experiments were conducted for offshore and inshore scenes. The results are shown in Table 6. It can be seen that our method produced a larger improvement under both scenes. Specifically, LR-TSDet improved the precision rate by 1.11 and the recall rate by 4.23 in inshore backgrounds, which indicates fewer false alarms and more correct predictions. In addition, it achieved 86.43 AP and 71.30 AP, outperforming the baseline by 4.81 AP in offshore backgrounds and 5.05 AP in inshore backgrounds. This set of controlled experiments shows the superiority of our LR-TSDet.

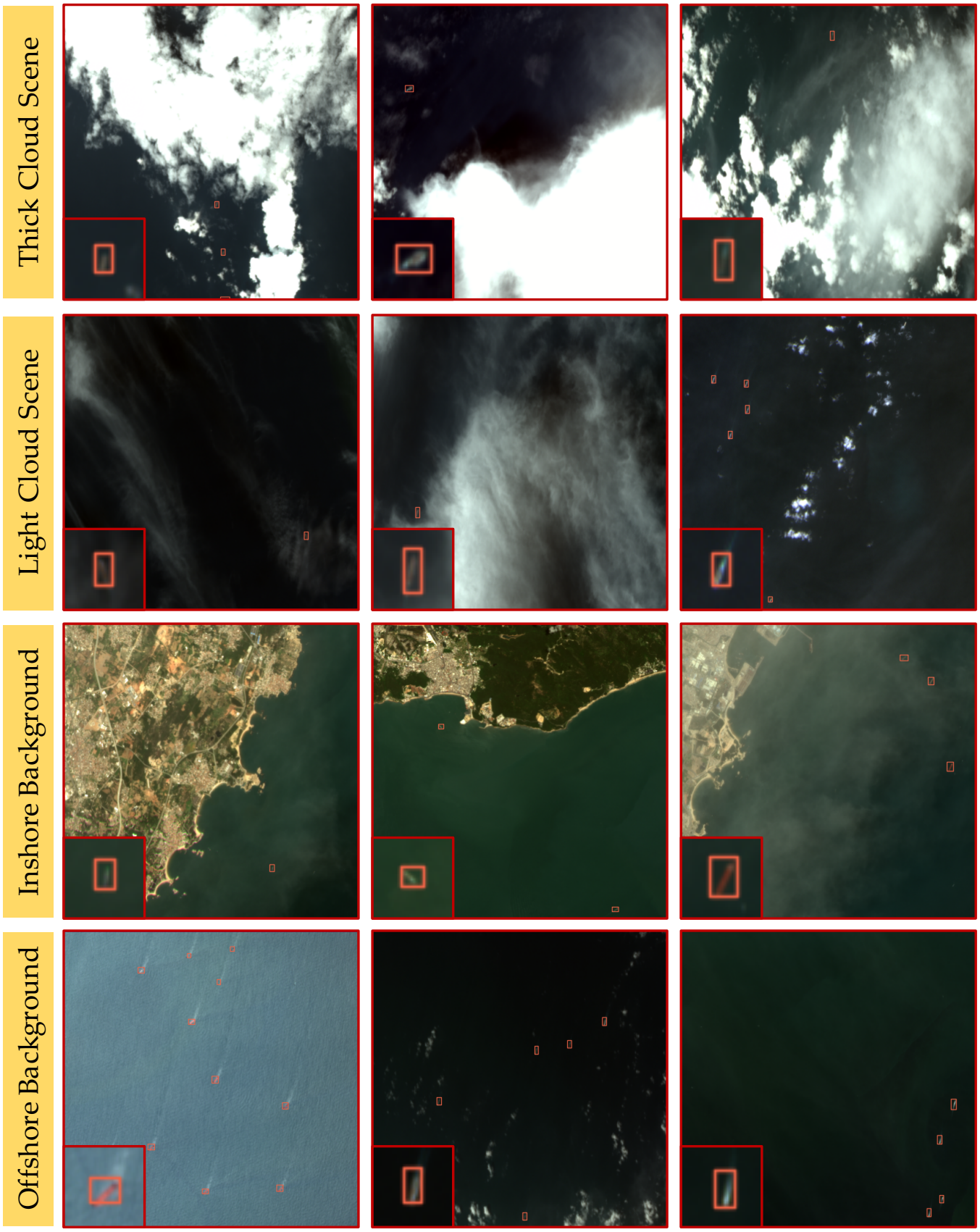


Figure 9. Detection results of LR-TSDet on GF1-LRSD with different backgrounds. The bottom-left corner of the image shows details of magnified results. The four scenes are thick cloud scenes, light cloud scenes, inshore backgrounds and offshore backgrounds, respectively.

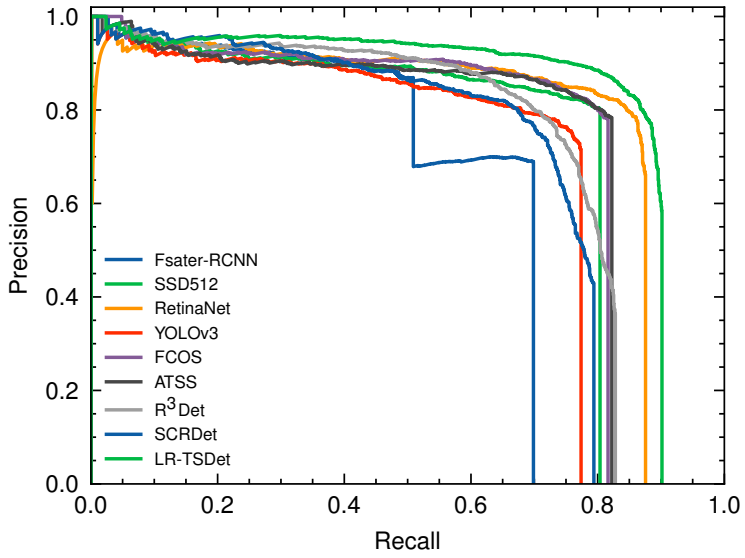


Figure 10. P-R curves of different approaches on GF1-LRSD dataset.

Table 6. Comparisons of different scenarios.

Scene	Method	Recall (%)	Precision (%)	AP (%)
Offshore	RetinaNet	89.64	65.96	81.62
	LR-TSDet	91.65	66.11	86.43
Inshore	RetinaNet	78.46	41.63	65.35
	LR-TSDet	82.69	42.74	71.30

4.5.2. Experiments on DOTA-Ship

To further demonstrate our proposed method, we also conducted experiments on the DOTA-Ship dataset. The models were trained for 48 epochs in total. The results are shown in Table 7. It can be observed that our LR-TSDet achieved an AP of 82.56 and performed better than other competitors. For example, our method produced considerable improvements of 6.98 AP by being carefully designed for tiny ship detection (e.g., the FFA module for global contextual information and the HASP module for deeper semantic information) compared with the baseline RetinaNet [26]. Some detection results are visualized in Figure 11.

Table 7. Comparisons with other typical methods on the DOTA-Ship dataset.

Method	Backbone	AP (%)
Faster-RCNN [25]	ResNet50	79.49
RetinaNet [26]	ResNet50	75.58
SSD512 [39]	VGG16	76.92
YOLOv3 [40]	DarkNet53	66.17
FCOS [47]	ResNet50	77.08
ATSS [73]	ResNet50	78.17
SCRDet [59]	ResNet50	79.57
R^3Det [30]	ResNet50	75.15
LR-TSDet (ours)	ResNet50	83.62

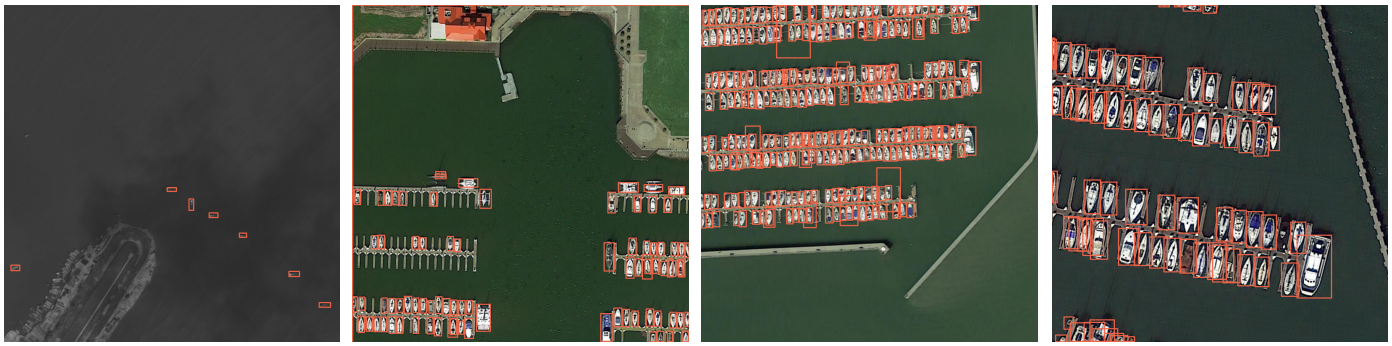


Figure 11. Detection results of LR-TSDet on the DOTA-Ship dataset.

5. Conclusions

In this article, we proposed an effective network architecture called LR-TSDet for improving the performance of tiny ship detection in low-resolution images. LR-TSDet includes three main components: the FFA module, the HASP module, and the IoU-Joint loss. Specifically, the FFA module was adopted to filter the background noise with the ability to capture long-range dependencies in feature maps in order to build a more robust FPN for detecting tiny objects. The HASP module was presented to obtain richer semantic information while maintaining the resolution of feature maps by aggregating four parallel HACBs, which is conductive to distinguishing tiny objects and the background. The IoU-Joint loss utilized the IoU score to alleviate the inconsistency between the classification and regression branches, and consequently improved the localization accuracy. To assess the feasibility of the proposed method, we constructed a dataset for low-resolution tiny ship detection in remote sensing images, called GF1-LRSD, in which the resolution (16 m) of images and the average size (10.9 ± 3.0 pixels) of instances are much smaller than available datasets. Comprehensive experiments on GF1-LRSD and DOTA-ship datasets demonstrated the efficacy of our LR-TSDet, which outperformed other comparison approaches.

Author Contributions: J.W. and Z.P. generated original ideas. J.W. designed and implemented the algorithm. Z.P., B.L. and Y.H. provided the experimental data and supervised the research. J.W. and Z.P. processed and analyzed the experimental results. The original draft was written by J.W. and reviewed by all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Cheng, G.; Han, J. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [CrossRef]
2. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
3. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
4. 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]
5. Zhang, Z.; Guo, W.; Zhu, S.; Yu, W. Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1745–1749. [CrossRef]
6. Liu, L.; Shi, Z. Airplane detection based on rotation invariant and sparse coding in remote sensing images. *Optik* **2014**, *125*, 5327–5333. doi: 10.1016/j.ijleo.2014.06.062. [CrossRef]
7. Li, Y.; Fu, K.; Sun, H.; Sun, X. An aircraft detection framework based on reinforcement learning and convolutional neural networks in remote sensing images. *Remote Sens.* **2018**, *10*, 243. [CrossRef]

8. Zhang, L.; Shi, Z.; Wu, J. A hierarchical oil tank detector with deep surrounding features for high-resolution optical satellite imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 4895–4909. [CrossRef]
9. Ok, A.O.; Başeski, E. Circular oil tank detection from panchromatic satellite images: A new automated approach. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1347–1351. [CrossRef]
10. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [CrossRef]
11. Diakogiannis, F.I.; Waldner, F.; Caccetta, P.; Wu, C. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS J. Photogramm. Remote Sens.* **2020**, *162*, 94–114. doi: 10.1016/j.isprsjprs.2020.01.013. [CrossRef]
12. Van Etten, A. You only look twice: Rapid multi-scale object detection in satellite imagery. *arXiv* **2018**, arXiv:1805.09512.
13. Pang, J.; Li, C.; Shi, J.; Xu, Z.; Feng, H. $\mathcal{R}^2$-CNN: Fast Tiny Object Detection in Large-Scale Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5512–5524. [CrossRef]
14. Shao, J.; Du, B.; Wu, C.; Zhang, L. Tracking objects from satellite videos: A velocity feature based correlation filter. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7860–7871. [CrossRef]
15. Long, G.; Chen, X.Q. A method for automatic detection of ships in harbor area in high-resolution remote sensing image. *Comput. Simul.* **2007**, *24*, 198–201.
16. Xu, J.; Sun, X.; Zhang, D.; Fu, K. Automatic detection of inshore ships in high-resolution remote sensing images using robust invariant generalized Hough transform. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 2070–2074.
17. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.
18. Liao, S.; Zhu, X.; Lei, Z.; Zhang, L.; Li, S.Z. Learning multi-scale block local binary patterns for face recognition. In Proceedings of the International Conference on Biometrics, Seoul, Korea, 27–29 August 2007; pp. 828–837.
19. Lowe, D. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157. [CrossRef]
20. Yang, F.; Xu, Q.; Li, B. Ship detection from optical satellite images based on saliency segmentation and structure-LBP feature. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 602–606. [CrossRef]
21. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
22. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
23. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
24. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
25. 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.* **2017**, *39*, 1137–1149. [CrossRef]
26. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007. [CrossRef]
27. Lam, D.; Kuzma, R.; McGee, K.; Dooley, S.; Laielli, M.; Klaric, M.; Bulatov, Y.; McCord, B. xview: Objects in context in overhead imagery. *arXiv* **2018**, arXiv:1802.07856.
28. 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—ICPRAM, Porto, Portugal, 24–26 February 2017; pp. 324–331. [CrossRef]
29. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning roi transformer for oriented object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 2849–2858.
30. Yang, X.; Yan, J.; Feng, Z.; He, T. R3Det: Refined Single-Stage Detector with Feature Refinement for Rotating Object. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 2–9 February 2021; Volume 35, pp. 3163–3171.
31. Wang, J.; Yang, W.; Guo, H.; Zhang, R.; Xia, G.S. Tiny Object Detection in Aerial Images. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 3791–3798.
32. Yang, X.; Hou, L.; Zhou, Y.; Wang, W.; Yan, J. Dense Label Encoding for Boundary Discontinuity Free Rotation Detection. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 15819–15829.
33. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. [CrossRef]
34. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local Neural Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803. [CrossRef]
35. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, PR, USA, 2–4 May 2016.

36. Gao, S.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P.H. Res2net: A new multi-scale backbone architecture. *arXiv* **2019**, arXiv:1904.01169.
37. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *arXiv* **2020**, arXiv:2006.04388.
38. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
39. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. *SSD: Single Shot MultiBox Detector*; ECCV: Amsterdam, The Netherlands, 8–16 October 2016.
40. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
41. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 821–830.
42. 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]
43. An, Q.; Pan, Z.; Liu, L.; You, H. DRBox-v2: An improved detector with rotatable boxes for target detection in SAR images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8333–8349. [CrossRef]
44. Yang, X.; Yan, J.; Qi, M.; Wang, W.; Xiaopeng, Z.; Qi, T. Rethinking Rotated Object Detection with Gaussian Wasserstein Distance Loss. In Proceedings of the International Conference on Machine Learning (ICML), Online, 18–24 July 2021.
45. 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. *arXiv* **2020**, arXiv:1911.09358.
46. Han, J.; Ding, J.; Li, J.; Xia, G.S. Align deep features for oriented object detection. *arXiv* **2021**, arXiv:2008.09397.
47. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
48. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
49. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
50. Qiu, H.; Ma, Y.; Li, Z.; Liu, S.; Sun, J. Borderdet: Border feature for dense object detection. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 549–564.
51. Wei, H.; Zhang, Y.; Chang, Z.; Li, H.; Wang, H.; Sun, X. Oriented objects as pairs of middle lines. *ISPRS J. Photogramm. Remote Sens.* **2020**, *169*, 268–279. doi: 10.1016/j.isprsjprs.2020.09.022. [CrossRef]
52. Lin, Y.; Feng, P.; Guan, J.; Wang, W.; Chambers, J. IENet: Interacting embranchment one stage anchor free detector for orientation aerial object detection. *arXiv* **2019**, arXiv:1912.00969.
53. Xiao, Z.; Qian, L.; Shao, W.; Tan, X.; Wang, K. Axis Learning for Orientated Objects Detection in Aerial Images. *Remote Sens.* **2020**, *12*, 908. [CrossRef]
54. Yang, Y.; Pan, Z.; Hu, Y.; Ding, C. CPS-Det: An Anchor-Free Based Rotation Detector for Ship Detection. *Remote Sens.* **2021**, *13*, 2208. [CrossRef]
55. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Augmentation for small object detection. *arXiv* **2019**, arXiv:1902.07296.
56. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3578–3587.
57. Noh, J.; Bae, W.; Lee, W.; Seo, J.; Kim, G. Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9725–9734.
58. Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; Han, Z. Scale match for tiny person detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, The Westin Snowmass Resort, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1257–1265.
59. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Sun, X.; Fu, K. Scrdet: Towards more robust detection for small, cluttered and rotated objects. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 8232–8241.
60. Hu, P.; Ramanan, D. Finding Tiny Faces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
61. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
62. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; PMLR: 2019; pp. 6105–6114.
63. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv* **2021**, arXiv:2103.14030.
64. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.

65. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Long Beach, CA, USA, 2017; Volume 30.
66. Wu, Y.; He, K. Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
67. Nair, V.; Hinton, G.E. *Rectified Linear Units Improve Restricted Boltzmann Machines*; Icml: Haifa, Israel, 21–24 June 2010.
68. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
69. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
70. Tzutalin, D. LabelImg. GitHub Repository. 2015. Volume 6. Available online: https://github.com/tzutalin/labelImg (accessed on 5 October 2015).
71. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
72. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the 2018 IEEE winter conference on applications of computer vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1451–1460.
73. 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. *arXiv* **2020**, arXiv:1912.02424.