remote sensing

MDPI

Article

Self-Adaptive Aspect Ratio Anchor for Oriented Object Detection in Remote Sensing Images

Jie-Bo Hou, Xiaobin Zhu * and Xu-Cheng Yin

School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China; b20160306@xs.ustb.edu.cn(J.-B.H); xuchengyin@ustb.edu.cn(X.-C.Y)
* Correspondence: zhuxiaobin@ustb.edu.cn

Abstract: Object detection is a significant and challenging problem in the study of remote sensing. Since remote sensing images are typically captured with a bird's-eye view, the aspect ratios of objects in the same category may obey a Gaussian distribution. Generally, existing object detection methods ignore exploring the distribution character of aspect ratios for improving performance in remote sensing tasks. In this paper, we propose a novel Self-Adaptive Aspect Ratio Anchor (SARA) to explicitly explore aspect ratio variations of objects in remote sensing images. To be concrete, our SARA can self-adaptively learn an appropriate aspect ratio for each category. In this way, we can only utilize a simple squared anchor (related to the strides of feature maps in Feature Pyramid Networks) to regress objects in various aspect ratios. Finally, we adopt an Oriented Box Decoder (OBD) to align the feature maps and encode the orientation information of oriented objects. Our method achieves a promising mAP value of 79.91% on the DOTA dataset.

Keywords: remote sensing images; object detection; aspect ratio; anchor

Citation: Hou, J.-B.; Zhu, X.; Yin, X.-C. Self-Adaptive Aspect Ratio Anchor for Oriented Object Detection in Remote Sensing Images. *Remote Sens.* **2021**, *13*, 1318. https://doi.org/10.3390/rs13071318

Academic Editor: Sébastien Lefèvre

Received: 25 February 2021
Accepted: 26 March 2021
Published: 30 March 2021

1. Introduction

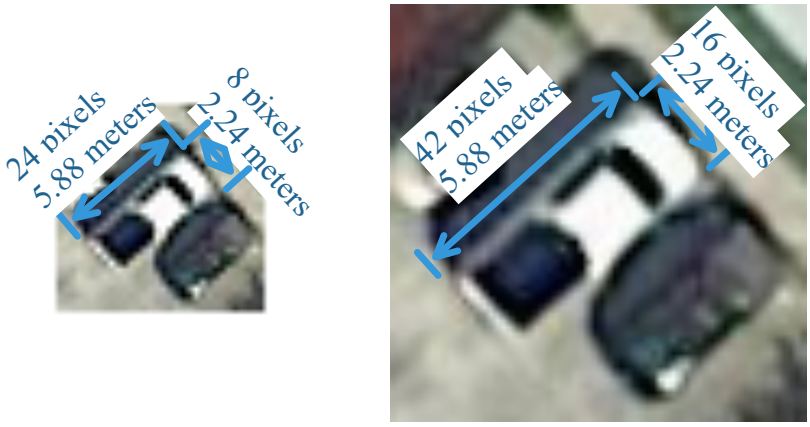
In recent years, with the developing of spaceborne sensors, the resolution of remote sensing images has greatly increased. This provides us a lot of high remote sensing images for researching and understanding. Object (i.e., plane, baseball diamond, bridge) detection in remote sensing images has become a hot research topic, and it is widely used in many applications, such as urban planning, ship detection, traffic controlling, and resource discovery [1,2].

Most of the existing object detection methods for remote sensing images are upon the popular methods designed for natural scene images. However, objects in remote sensing images (typically taken with a bird's-eye view) are quite different from objects in natural scene images [3] and are facing the following challenges [1,3–9]:

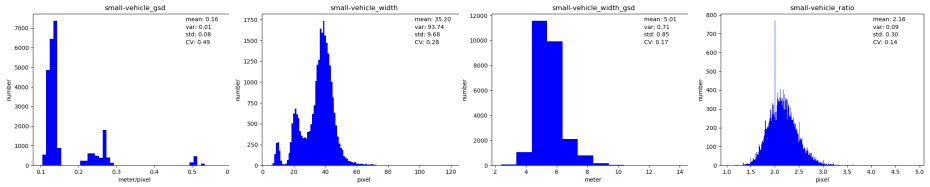
- **Arbitrary orientations:** Objects in natural scenes are observed from the horizontal view and annotated in horizontal bounding boxes (HBBs). However, objects in remote sensing images can appear in arbitrary orientations and are generally annotated in oriented bounding boxes (OBBs).
- **Background complexity:** The complex background in remote sensing images often contains noise or uninteresting objects which may lead to false positives.
- **Scale variations:** Due to the resolutions of spaceborne sensors are not completely consistent, the ground sample distance (GSD) (the physical size of one image pixel in meters, i.e. meter per pixel) of images is often in variation. Thus, the scales of the same category of objects, such as vehicles, are often with different number pixels even they are the same type of vehicles. This will cause scale variations in detection.
- **Dense objects:** Some objects in remote sensing images are always densely arranged, such as vehicles in parking lots or ships in harbors. It is hard to separate dense and small objects in images.

In general, object detection methods [10–12] for natural scenes images contain four parameters dimensions ($[x, y, w, h]$), such as Faster R-CNN [13], YOLO V3 [14], RetinaNet [15], etc. However, due to **arbitrary orientations**, methods for remote sensing images usually utilize five parameters dimensions ($[x, y, w, h, \theta]$). Thus, most of existing methods for remote sensing images, e.g., RRPN [16], SCRDet [4], S^2A-Net [17], modified the object detection methods and add the angle dimension. For the **background complexity** problem, KARNET [18], SCRDet [4], and SCRDet++ [19] utilize attention mechanism to denoise. For **scale variations** and dense **objects problems**, enhanced Feature Pyramid Networks (OWSR [20], FFA [21], and RADet [22]) and novel anchor mechanism (A^2S-Det [23] and DAL [24]) are proposed. The above methods achieve excellent performance in remote sensing images.

However, most of the existing methods lose sight of the distinguished features of objects in remote sensing images. Objects in natural scene images can be taken 360° around, the aspect ratios will change due to perspective transformation. However, remote sensing images are typically taken with a bird's-eye view, the aspect ratios of objects in the same category may obey a Gaussian distribution which may be related to its category (as shown in Figures 1f and 2). And as shown in Figure 2 and listed in Table 1, different categories have different aspect ratio distributions. From this key observation, we fit category-wise aspect ratios to benefit the regression of objects in remote sensing images.



(**a**) GSD = 0.28 m/pixel; aspect ratio = 2.625. (**b**) GSD = 0.14 m/pixel; aspect ratio = 2.625.

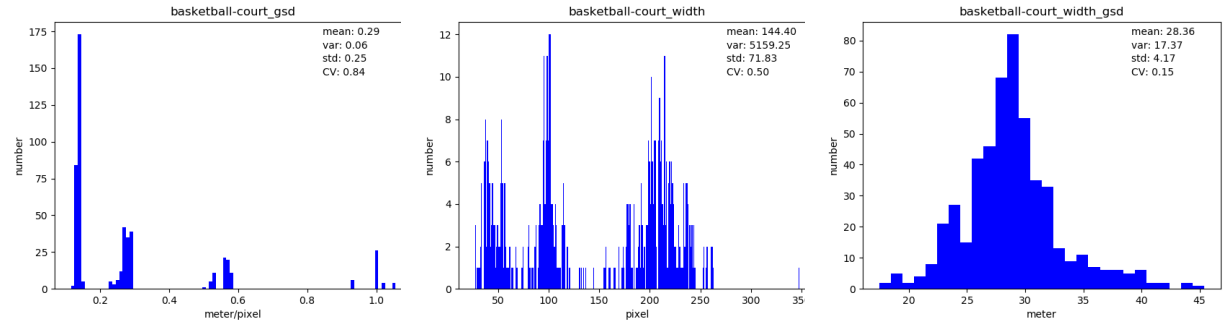


(**c**) GSD. (**d**) Width in pixels. (**e**) Width in meters. (**f**) Aspect ratio.

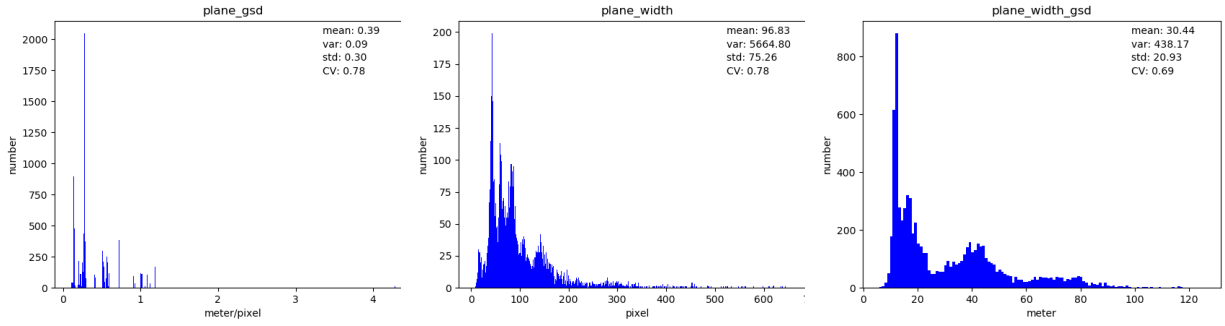
Figure 1. GSD, width in pixel, width in meter, and aspect ratio of small-vehicle.

Table 1. Statistical results of aspect ratios, visualization are shown in Figure 2. "Var" represents Variance, "Std" represents Standard Deviation, and "CV" represents Coefficient of Variation.

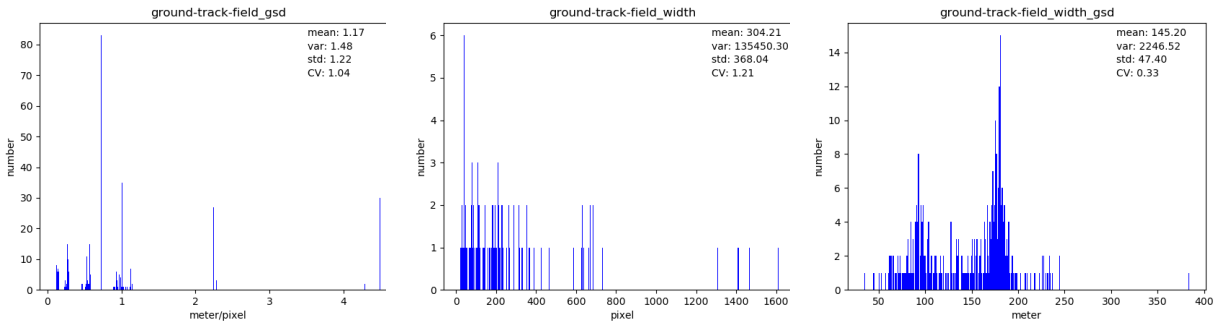
	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC
Mean	1.26	1.10	3.15	1.97	2.16	3.93	2.91	2.80	1.78	1.09	1.65	1.10	4.39	1.69	2.93
Var	0.04	0.01	12.85	0.10	0.09	1.44	0.65	0.01	0.04	0.05	0.06	0.02	17.35	0.24	0.65
Std	0.20	0.12	3.59	0.31	0.30	1.20	0.81	0.12	0.19	0.22	0.24	0.16	4.17	0.49	0.81
CV	0.16	0.11	1.14	0.16	0.14	0.31	0.28	0.06	0.11	0.21	0.15	0.14	0.95	0.29	0.28



(**a**) basketball-court.



(**b**) plane.



(**c**) ground-track-field.

Figure 2. The GSD, width in pixel, and width in meters of basketball-court, plane, and ground-track-field. The width in meter of basketball-court is close to Gaussian distribution. However, the width in meters of plane and ground-track-field are close to mixed Gaussian distribution.

Ground sample distance (GSD) is another specific property of remote sensing images; it is one of the causes for scale variations (GSD means how many meters per pixel), and object widths in pixels can be variable (Figure 1d, but could be more cohesive in meters (Figure 1e). GSD of the other two categories are shown in Figure 2, some categories, such as plane and ground-track-field, may still suffer from scale variations in meters, so we focus on aspect ratio and do not lucubrate GSD in this work. Anchor mechanism with several pre-defined ratios is one way to deal with ratio variations, e.g., RRPN [16], utilizes three ratios and totally defines 54 rotated-anchors. But the predefined coefficients may not meet the various shapes of different datasets, and different categories may obey different distributions.

In this paper, we propose a novel framework for oriented object detection in remote sensing images. Our method mainly consists of two components: Self-Adaptive Aspect Ratio Anchor Mechanism (SARA) and Oriented Box Decoder (OBD). We propose a novel Self-Adaptive Aspect Ratio Anchor (SARA) for matching the aspect ratio variations of

objects in remote sensing images, which can self-adaptively learn the appropriate aspect ratio for each category (i.e, category-wise aspect ratios). Specifically, SARA learns one aspect ratio value for each category, then utilizes the predicted classification scores as attention weights, i.e., the category-wise aspect ratios multiply by the classification scores, then the weighted sum of them generate the aspect ratio of a given sample. Then, our method can regress the oriented bounding boxes for objects by the reference of aspect ratio and a simple squared anchor (related to the stride of feature maps in Feature Pyramid Networks). In OBD, for better oriented boxes regression, we adopt Alignment Convolution Layer (ACL) [17] and active rotating filters (ARF) [25] to align the feature maps and encode the orientation information, respectively.

- We propose a novel Self-Adaptive Aspect Ratio Anchor (SARA) for matching the aspect ratio variations of objects in remote sensing images.
- Our SARA can be Plug-and-Play for methods with anchor-free or simple squared anchor without considering the information of aspect ratio.
- Our method achieves state-of-the-art on the DOTA dataset.

2. Related Work

2.1. Anchor-Based Methods

An Illustration of anchor-based methods is shown in Figure 3. Faster R-CNN [13] is the first method that introduces the novel "anchor" boxes that serve as references at multiple pre-defined scales and aspect ratios. Anchors enumerate the possible locations and shapes of objects in a sliding-window fashion. A classical anchor can be defined as $A = (x_a, y_a, w_a, h_a)$, where x_a, y_a is the position of anchor's center point, w_a, h_a represent the width and height of anchor, respectively. For oriented object detection, rotated anchors [16] $A_r = (x_a, y_a, w_a, h_a, \theta_a)$ can be adopted for better performance, where θ_a denotes the angle of anchor. Anchors are enumerated in each position of feature maps, if the intersection-over-union (IoU) between one sample's target bounding box and the pre-defined anchor is greater than the threshold, it is assigned as a positive sample, and the regression targets is related to the corresponding anchor. With the help of pre-defined anchors, objects can be classified into different scales and ratios, and the networks can be optimized stably. Anchor-based methods can also be roughly divided into two categories, i.e., one-stage methods and two-stage methods.

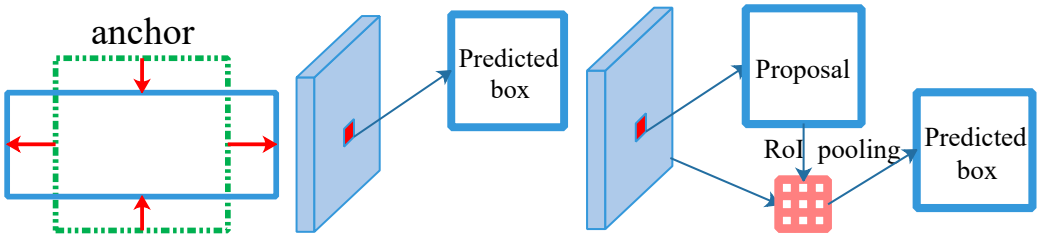


Figure 3. Illustration of Anchor-based methods. (**a**) anchor-based methods regression target box (blue box) based on the predefined anchor (green dotted box); (**b**) one-stage methods directly regress the bounding boxes based on the anchors; (**c**) two-stage methods firstly generate proposals based on anchors, then refine proposals through RoI pooling layer.

Two-stage methods usually utilize Region Proposal Networks (RPN) [13] to generate rough region proposals based on anchors at the first stage, then refines the final predicted boxes through an RoI Pooling layer which resize the cropped feature maps into the same shape. Faster R-CNN [13] is a standard and the first two-stage method. RRPN [16] adds the third dimension angle (θ) to anchor, to generate rotated 50 anchors (3 scales, 3 ratios, and 6 angles) for detecting oriented objects. RoITransformer [26] utilizes a learn-able module to transform horizontal RoIs into rotated RoIs, which can avoid a large number

of rotated anchors. However, RoITransformer still needs pre-defined horizontal anchors and complex RoI operations. RoITransformer, SCRDet [4], and Azimi et al. [53] are representative methods that mainly focus on the perspective of feature extraction. Mask OBB [6] proposes a mask-oriented bounding box representation, it adopts RoI Align [27] to extract feature maps from proposals, and then it utilizes the segmentation method, i.e., pixel-level classification for detecting oriented objects.

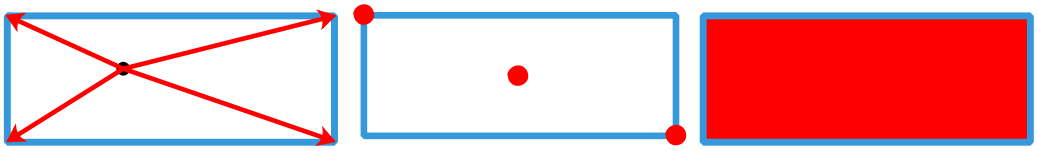
One-stage methods can be regarded as two-stage methods without RoI Pooling, they predict results directly. SSD [11], YOLO [14,28], and RetinaNet [15] are representative one-stage methods, and further improves detection speed. Especially the proposed Focal Loss in RetinaNet [15], which tries to address foreground-background class imbalance problem, and improves the performance of one-stage methods even surpassing the accuracy of two-stage. Focal Loss is still widely used in state-of-the-art detectors. TextBoxes++ [29] is designed for arbitrary-oriented scene text detection, it is modified from SSD and regresses the offsets of the rotated targets to the four points of horizontal anchors. R^3Det [30] samples features from five locations (center and four corner points) of the corresponding anchor box to balance the trade-off of accuracy and speed.

Anchors are widely utilized in many detectors, however, the design of anchors is complex especially in the rotated object detection task. Two-stage methods need complex RoI operations, which often cost a lot of time, while one-stage methods can balance the trade-off of accuracy and speed by ingenious design, such as Focal Loss.

2.2. Anchor-Free Methods

Due to the complexity of pre-defined anchors, many anchor-free methods have become popular in recent years, they can be roughly divided into three categories: directly regression methods, corner-based methods, and segmentation-based methods.

Directly regression methods usually contain two branches: classification branch and regression branch. The classification branch often adopts Fully Convolutional Networks (FCN) [31] for pixel-wise classification, while the regression branch regresses the offsets of the current position to the four points of rotated or horizontal bounding box directly without anchor references as shown in Figure 4. DenseBox [32] is the first directly regression method for face detection, and it is designed for horizontal bounding box detection. EAST [33] is a popular arbitrary-oriented scene text detection method, it adopts IoU Loss [34] for regressing boxes and regresses an angle for rotated boxes in addition. EAST also designs another regression way, i.e., regress the offset of each positive point to the four points of bounding boxes. DDR [35,36] is another arbitrary-oriented scene text detection method, which is similar to EAST. FCOS [37] proposes a centerness branch to help suppress the low-quality detected bounding boxes and improves the overall performance by a large margin. Zhou et al. [38] utilizes a point to represent one object, which predicts the center point of an object and then regress the width and height of the bounding box. FASF [39] automatically assigns objects into different levels of Feature Pyramid Networks (FPN) [40] rather than based on areas of objects.



(**a**) Directly regression methods.. (**b**) Corner-based methods. (**c**) Segmentation-based methods.

Figure 4. Illustration of Anchor-free methods. (**a**) directly regression methods usually directly regress the offset of each positive point to the four points of bounding boxes without anchor references. (**b**) Corner based methods predict the corners of bounding box (some methods contain center point), then pair the top-left point and bottom-right point to generate the bounding box. (**c**) segmentation-based methods regard detection task as segmentation task, them get the minimum bounding rectangle as the final predictions.

Corner-based methods focus on predicting the keypoints of objects. CornerNet [41], CenterNet [42], and ExtremeNet [43] try to predict some key points of objects, such as corners, centers, or extreme points of objects, and then pair them into bounding boxes for detection. There are also some detectors inspired by them for object detection in remote sensing tasks [3,44].

Segmentation-based methods are widely used in arbitrary-oriented scene text detection task. They usually adopt an FCN for foreground/background classification, and try to group target regions by other information, such as embedding. PixelLink [45] learns the eight-neighbors linkage of each pixel and then links pixels into targets through a disjoint-set data structure. TextField [46] predicts the direction field of each point in the text, each direction is the nearest boundary point points to the current text point. Tian et al. [47] utilizes embedding maps to cluster text points into text instances. PSENet [48] shrinks the original target with different scales and then expands the minimal scale kernel to the complete target gradually.

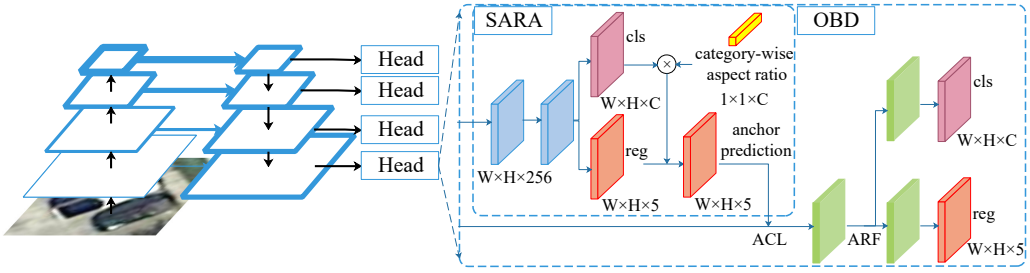


Figure 5. The pipeline of proposed method, which contains two parts: Self-Adaptive Aspect Ratio Anchor Mechanism (SARA) and Oriented Box Decoder (OBD). The SARA can be regarded as high-quality anchor generator, which products anchor prediction map for the following OBD, details are illustrated in Figure 7. The OBD adopts Alignment Convolution Layer (ACL) [17] and Active Rotating Filters (ARF) [25] to align the feature maps and encode the orientation information, respectively.

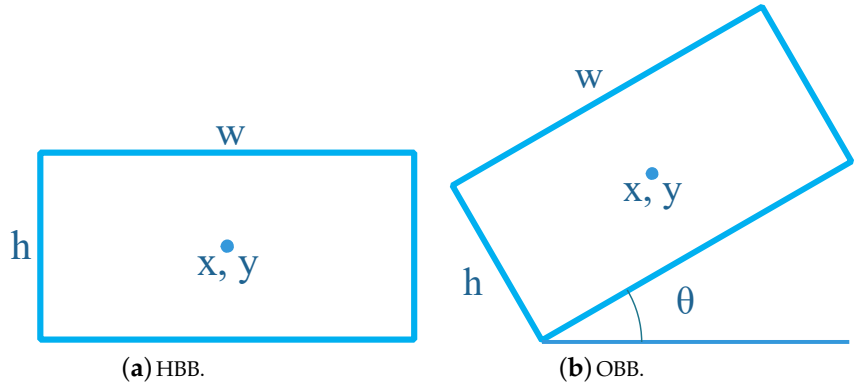
3. Our Method

3.1. Overall Structure

As shown in Figure 5, our method contains two parts: Self-Adaptive Aspect Ratio Anchor Mechanism (SARA) and Oriented Box Decoder (OBD). The SARA can be regarded as a high-quality anchor generator, which products anchor prediction map for the following OBD. The OBD adopts Alignment Convolution Layer (ACL) [17] and Active Rotating Filters (ARF) [25] to align the feature maps and encode the orientation information, respectively. In this way, the feature maps can benefit from the oriented bounding box regression task.

3.2. HBB and OBB

Most of the existing methods for object detection are designed for horizontal bounding boxes (HBB). However, we focus on the oriented bounding boxes (OBB) task of DOTA in this work. As shown in Figure 6a, HBB contains four dimensions (x, y, w, h), which means the center point position (x, y), the width, and the height, respectively. Figure 6b shows that OBB contains five dimensions (x, y, w, h, θ), which means the center point position (x, y), the width (**the longest side**), the height (**the shortest side**), and θ denote the angle of OBB. Here, $\theta \in [-\frac{\pi}{4}, \frac{3\pi}{4}]$. HBB is a particular case of OBB, i.e., $\theta = 0$.



(**a**) HBB. (**b**) OBB.

Figure 6. Horizontal bounding boxes (HBB) and oriented bounding boxes (OBB). (**a**) HBB is widely utilized in object detection task, it contains four dimensions (x, y, w, h), which means the center point position (x, y), the width, and the height, respectively. (**b**) OBB contains five dimensions (x, y, w, h, θ), which means the center point position (x, y), the width (the longest side), the height (the shortest side), and θ denote the angle of OBB. HBB is a particular case of OBB, i.e., $\theta = 0$.

3.3. Self-Adaptive Aspect Ratio Anchor Mechanism

The Self-Adaptive Aspect Ratio Anchor Mechanism (SARA) can be regarded as a high-quality anchor generator. As shown in Figure 7, the inputs of SARA are two branches: classification ($W \times H \times C$) and regression ($W \times H \times 5$). W and H represent the width and height of the feature maps, C means there are C categories. The outputs of SARA are predicted initiatory oriented boxes. In SARA, each category has a self-adaptive aspect ratio value (i.e., category-wise aspect ratios), these category-wise aspect ratios are unsupervised updated with the gradient of the following anchor prediction. Notably, the predicted category-wise aspect ratios of SARA only contribute to the regression reference rather generates real anchors for matching samples. The output anchor predictions for the following OBB are also for regression reference only.

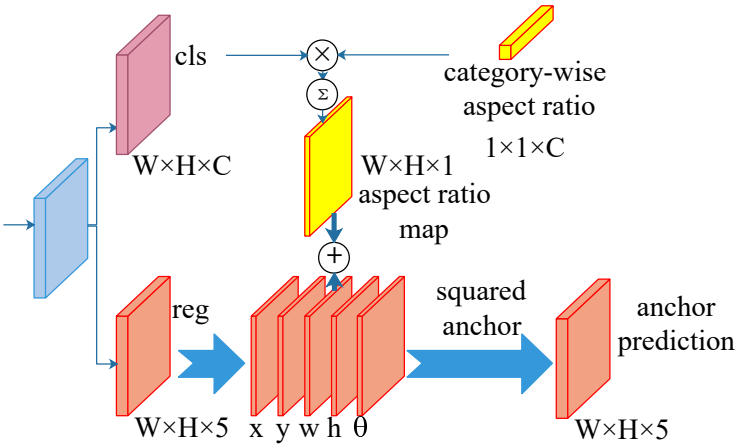


Figure 7. Category-wise aspect ratios are self-adaptive aspect ratio for each category. The category-wise aspect ratios firstly multiply by the classification probabilities, then the weighted sum of them generate the aspect ratio map. The w (width) channel adds the values of aspect ratio map to generate the w_{ar} in Equation (2). After converting with the simple squared anchor, the final anchor prediction is generated.

The predicted classification probabilities ($W \times H \times C$) can be regarded as attention weights for the category-wise aspect ratios. For each position in classification, it has C channels to represent the probabilities of the C categories. The category-wise aspect ratios multiply by the classification probabilities, then the weighted sum of them generate the aspect ratio of a given sample. The predicted aspect ratios of all positions ($W \times H \times 1$)

compose the aspect ratio map. The regression maps contains 5 channels, i.e., (x, y, w, h, θ), the w (width) channel add the aspect ratio map then generates the regression results benefited aspect ratio. After converting with the simple squared anchor (related to the strides of feature maps in Feature Pyramid Networks), the final oriented boxes of SARA is predicted. The details of one position are mathematically described as following:

$$AR = (\sum_{i=1}^{C} P_i^{cls} \cdot C_{ar}) / (\sum_{i=1}^{C} P_i^{cls}), \tag{1}$$

$$w_{ar} = w + AR, \tag{2}$$

where AR denotes predicted aspect ratio of current position; P_i^{cls} means the probability of i-th category; C_{ar} represents the category-wise aspect ratios; w denotes the predicted width; and w_{ar} means sum of predicted width and predicted aspect ratio.

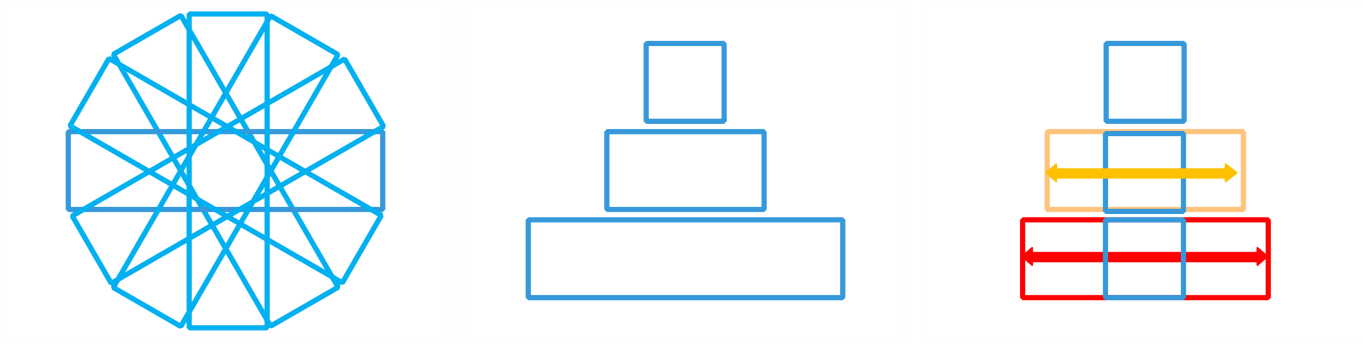
Following previous works for OBB (such as S^2A-Net [17]), the regression targets of SARA and the following OBB is defined as

$$\begin{aligned} \Delta x &= R(\theta) \frac{(x_g - x_a) R(\theta)}{w_a}, \\ \Delta y &= R(\theta) \frac{(y_g - y_a) R(\theta)}{h_a}, \\ \Delta w &= log(w_g) - log(w_a), \\ \Delta h &= log(h_g) - log(h_a), \\ \Delta \theta &= \frac{\theta_g - \theta_a + k\pi}{\pi}, \end{aligned} \tag{3}$$

where Δx, x_g, and x_a represent the regression target x, the ground-truth x, and the anchor center x, respectively (likewise for y, w, h, θ), and $R(\theta)$ denotes the transformational relation between θ and width/height. Combining the Equation (2), the origin task of baseline if to regress the w to the target Δw. When we add our SARA, the task will change to regress the w_{ar} (i.e., $w + AR$) to the target Δw.

Compare with Other Anchor Mechanism

As shown in Figure 8, different from complex pre-defined anchors, i.e., rotated anchors (RRPN [16]) or anchors with several pre-defined ratios (Faster R-CNN [13]), our SARA only utilize one simple squared anchor. The scale $(4 \times S)$ of the squared anchor we utilized is related to the strides (S) of feature maps in Feature Pyramid Networks. As described in Section 3.3, our SARA utilizes the category-wise self-adaptive aspect ratio to generate anchor references with various aspect ratios.



(**a**) Rotated anchors. (**b**) Anchors with predefined ratios. (**c**) Proposed Self-Adaptive Aspect Ratio Anchor.

Figure 8. Comparison with other anchors. (**a**) rotated anchors such as RRPN [16]; (**b**) Anchors with several predefined aspect ratios such as Faster R-CNN [13]; (**c**) proposed Self-Adaptive Aspect Ratio Anchor (SARA), SARA utilize one simple squared anchor and category-wise self-adaptive aspect ratio to generate anchor references with various aspect ratios.

3.4. Oriented Box Decoder

The predicted oriented bounding boxes by SARA can be regarded as predicted high-quality anchors (anchor prediction in Figure 7), we utilize an Oriented Box Decoder (OBD) to refine the oriented bounding boxes based on these predicted anchors. Inspired by S^2A-Net [17], we adopt Alignment Convolution Layer (ACL) [17] and Active Rotating Filters (ARF) [25] to align the feature maps and encode the orientation information, respectively. In this way, the feature maps can benefit from the oriented bounding box regression task. The regression targets of OBB are the same as SARA in Equation (3).

3.4.1. Feature Alignment

Feature Alignment is important for oriented object detection, we utilize the AlignConv [17] (shown in Figure 9c) to adaptively align the feature maps based on the predicted oriented bounding boxes. AlignConv can alleviate the misalignment between axis-aligned feature maps and oriented objects in a fully convolutional way. Alignment Convolution Layer (ACL) [17] is a layer adopt AlignConv to extract aligned features by the predicted oriented bounding boxes (anchor prediction in Figure 7).

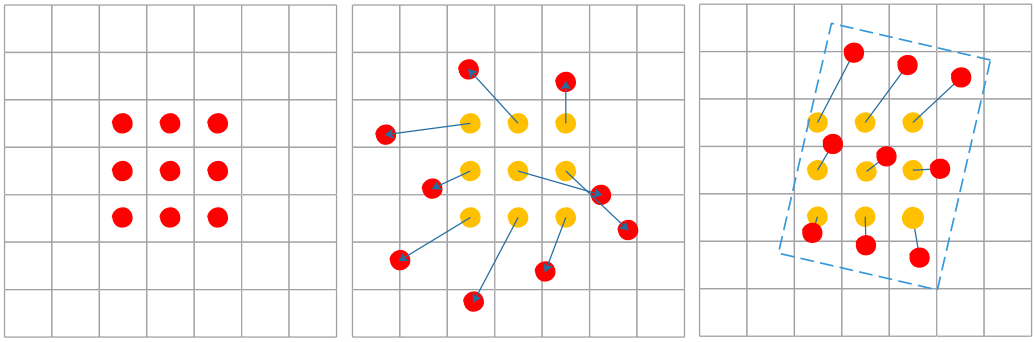


Figure 9. Examples of the sampling locations in different methods with 3×3 convolution kernel. (**a**) Standard 2D convolution; (**b**) the sampling locations of Deformable Convolution [49] are predicted for each point; (**c**) the sampling locations of AlignConv [17] are the corresponding oriented bounding box.

3.4.2. Feature Orientation Information

Since we need to detect oriented bounding boxes (OBB) of objects, we adopt Active Rotating Filters (ARF) [25] to encode the orientation information, which can benefit the oriented bounding box regression task. An ARF is defined as a $W \times W \times N$ filter that rotates N-1 times for convolution to generate N orientation feature maps. With an ARF (F), for an input feature map X, the k-th orientation output feature map $Y^{(k)}$ is defined as

$$Y^{(k)} = \sum_{n=0}^{N-1} F_{\theta_k}^{(n)} * X^{(n)}, \theta_k = k\frac{2\pi}{N}, k = 0, ..., N-1, \tag{4}$$

where F_{θ_k} is the clockwise θ_k-rotated version of F, and $F_{\theta_k}^{(n)}$ and $X^{(n)}$ are the n-th orientation channel of F_{θ_k} and X, respectively.

4. Loss Function

The loss function of our method contains two parts: loss of SARA and loss of OBD. The loss function of SARA and OBD are the same, the both utilize Focal loss [15] (L_c) for classification and Smooth-L1 loss (L_r) for regression.

$$L_{c\_r} = \frac{1}{N}\left(\sum_{i \in \Omega_c} L_c(F_i^c, G_i^c) + \sum_{i \in \Omega_r} L_r(F_i^r, G_i^r)\right) \tag{5}$$

$$L = L_{SARA} + L_{OBD}, \tag{6}$$

where Ω_c and Ω_r denote all the samples in feature maps for classification and all the positive samples for regression, respectively; F_i^c and G_i^c mean the i-th predicted classification result and ground-truth, respectively; F_i^r and G_i^r represent the i-th predicted regression result and ground-truth, respectively; L_{c_r} denote the sum of classification and regression loss; and L_{SARA} and L_{OBD} represent loss for SARA and OBD, respectively. They both use L_{c_r}.

5. Experiments

5.1. Dataset Description

DOTA [50] is one of the largest remote sensing datasets for object detection which contains 15 categories: plane (PL), baseball diamond (BD), bridge (BR), ground field track (GTF), small vehicle (SV), large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), and helicopter (HC). It contains 2806 images collected from different sensors and platforms, has been divided into a training set (1411 images), validation set (458 images), and testing set (937 images). These images are with size ranges from 800×800 to 4000×4000. There are 188,282 annotated object instances, which are labeled by arbitrary quadrilaterals, and it contains two detection tasks: horizontal bounding boxes (HBB) and oriented bounding boxes (OBB). We focus on OBB task in this paper.

Following most of the other state-of-the-art methods, such as Reference [6,17,23], our method is trained on the training and validation set, and evaluated on the testing set. In the training phase, we randomly flip and rotate the images as data augmentation, we also crop a series of 1024×1024 patches from original images with a stride of 824 following Reference [17,50]. In the testing phase, we crop testing images into 1024×1024 patches with a stride of 512. And for multi-scale testing, we resize original images into three scales (i.e., 0.5, 1.0, and 1.5).

5.2. Implementation Details

5.2.1. Training and Inference

We utilize ResNet-101/ResNet-50 [51] with Feature Pyramid Networks (FPN) [40] as our backbone. Following Reference [17], we adopt one simple squared anchor for each level of FPN, and the scale ($4 \times S$) of the squared anchor is related to the strides (S, i.e., 32, 64, 128, 256, 512) in FPN. Our method is trained with a total batch size of 16 for 12 epochs on the DOTA dataset, using 4 GTX 1080Ti GUPs. The learning rate, momentum, and weight decay are 001, 0.9, and 0.001, respectively. At the testing phase, NMS is utilized as the post-processing step.

5.2.2. Evaluation Indicators

AP is a popular evaluation indicators, which is the average precision of the target in the range of recall=$[0, 1]$, and is generally the area under precision-recall curve (PRC). PRC can be calculated by recall and precision. Recall (R) and precision (P) are defined as

$$R = \frac{TP}{TP + FN}, \tag{7}$$

$$P = \frac{TP}{TP + FP}, \tag{8}$$

$$AP = \int_0^1 P(R)dR, \tag{9}$$

where true positive (TP), false positive (FP), and false-negative (FN) represent the number of correctly detected targets, the number of incorrectly detected targets, and the number of non-detected targets, respectively. mAP is adopted for multi-class evaluation, which is the average value of AP values for all classes,

$$mAP = \frac{\sum_{N}^{i=1} AP_i}{N}, \tag{10}$$

where N denotes the number of class. The larger the mAP value, the better the object detection performance. The evaluation of mAP on DOTA is reported by submitting the results to the official DOTA evaluation server.

5.3. Ablation Study

To verify the effectiveness of the proposed SARA, we compare our SARA and baseline in this Section. Our SARA can be Plug-and-Play for methods with anchor-free or simple squared anchor without aspect ratio. Thus, we adopt S^2A-Net [17] as the baseline in our work, which achieves state-of-the-art and only adopts one simple squared anchor.

Baseline: The baseline network is S^2A-Net [17] which is a state-of-the-art method for oriented object detection on DOTA dataset. In Table 2, † means S^2A-Net [17] does not report the performance, and we re-evaluate and re-inference them based on the provided models and code https://github.com/csuhan/s2anet (accessed on 26 March 2021).

Table 2. Ablation study. † means Baseline [17] does not report the performance, we re-evaluate and re-inference them base on the provided model of the Baseline[17]. R-101 represents ResNet-101 with FPN (so does R-50). The short names of the categories are plane (PL), baseball diamond (BD), bridge (BR), ground field track (GTF), small vehicle (SV), large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP) and helicopter (HC), respectively. Note that the FPS is a relative FPS calculated by the overall inference time and the number of chip images.

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP	FPS
Baseline †	R-50	88.94	83.77	57.49	72.62	80.14	81.76	88.72	90.83	85.36	86.98	64.33	68.75	78.10	73.45	69.18	78.02	11.6
Baseline+SARA	R-50	89.13	85.81	54.88	73.39	80.31	81.84	89.07	90.78	87.45	87.02	65.02	66.52	78.38	80.08	70.13	**78.65**	10.9
Baseline [17]	R-101	88.70	81.41	54.28	69.75	78.04	80.54	88.04	90.69	84.75	86.22	65.03	65.81	76.16	73.37	58.86	76.11	12.7
Baseline †	R-101	88.96	84.65	56.96	73.21	80.06	81.71	88.71	90.78	84.80	86.13	62.39	70.44	78.58	73.96	63.77	77.67	9.3
Baseline+SARA	R-101	89.20	84.60	55.94	73.71	79.77	82.03	88.99	90.75	85.70	87.25	62.36	66.61	78.94	75.57	67.60	**77.93**	8.8

Effect of proposed SARA: Representative results are shown in Figure 10. We can figure out that our SARA performs better than baseline, especially in objects with big aspect ratios. We train and test our method on the DOTA dataset with both ResNet-50/ResNet-101 as backbone, and our SARA can outperforms baseline in both of these backbone with small degradation in speed (FPS). Quantitative results are listed in Table 2. When we adopt ResNet-50 as backbone, our method (baseline+SARA) improves 0.63% in terms of mAP (78.65% versus 78.02%) compared with the baseline †. Specifically, our method achieves improvement in many categories: 89.13% versus 88.94% for PL, **85.81% versus 83.77% for BD**, 73.39% versus 72.62% for GTF, 80.31% versus 80.14% for SV, 81.84% versus 81.76% for LV, 89.01% versus 88.72% for SH, **87.45% versus 85.36% for BC**, 87.02% versus 86.98% for ST, 65.52% versus 64.33% for SBF, 78.38% versus 78.10% for HA, **80.08% versus 73.45% for SP**, and 70.13% versus 69.18% for HC. As for ResNet-101 as backbone, our method outperforms 1.82% in terms of mAP (77.93% versus 76.11%) compared with the baseline [17] (84.60% versus 81.41% for BD, 73.71% versus 69.75% for GTF, 75.57% versus 73.37% for SP, and 67.60% versus 58.86% for HC). We also compare our method with baseline †, i.e., 77.93% versus 77.67%, which improves 0.26% in terms of mAP.

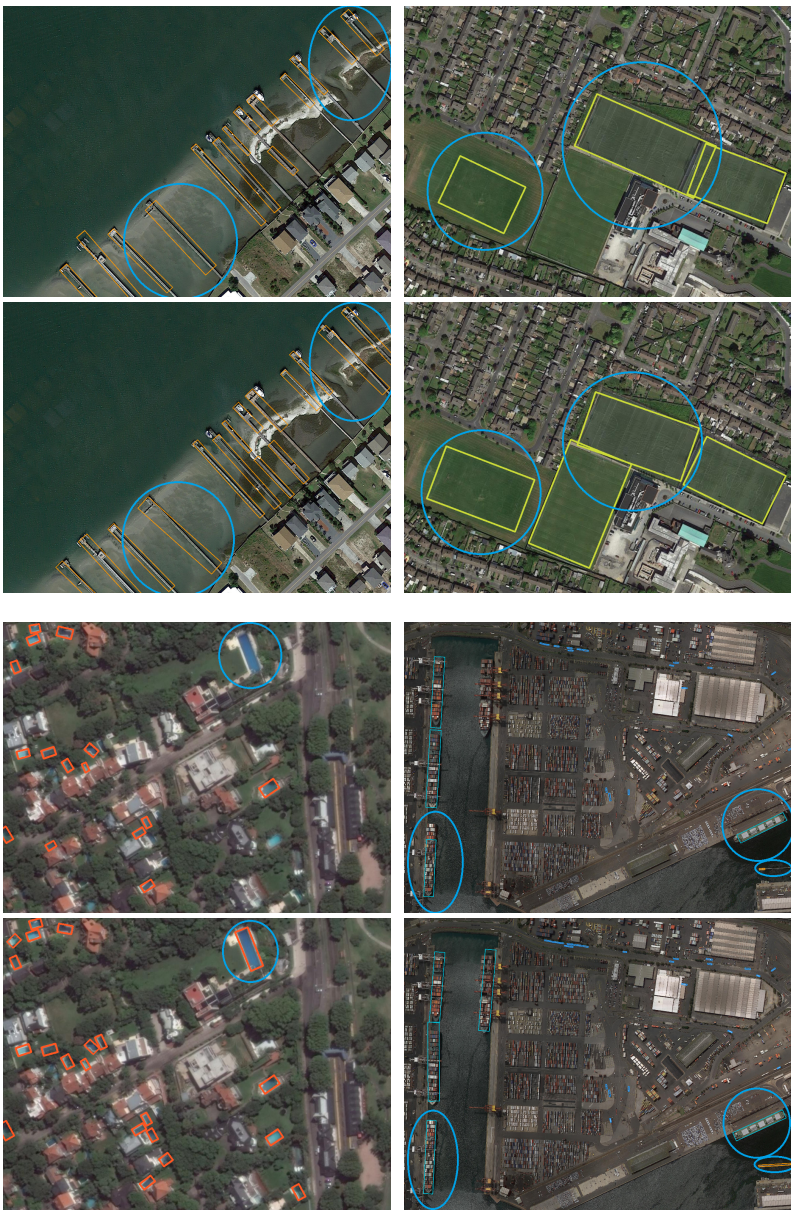


Figure 10. Representative results of ablation study. The first row of each subfigure shows detected results of baseline, and the second row shows results of our SARA. Blue ellipses hight the significant differences. Notably, we only show the detect boxes whose score is higher than a threshold of 0.5.

5.4. Comparisons with the State-of-the-Arts

Representative results are shown in Figure 11. Table 3 shows a comparison of our method (SARA) with state-of-the-art methods, including one-stage methods (RetinaNet [15], A^2S-Det [23], DRN [52], R^3Det [30], and S^2A-Net [17]) and two-stage methods (FR-O [50], Azimi et al. [53], RoITransformer [26], CAD-Net [54], SCRDet [4], GLS-Net [5], Xu et al. [55], Mask OBB [6], and F^3-Net [1]). Our SARA surpassed all the state-of-the-art methods and is only a slightly slower than S^2A-Net (10.9 FPS vs 11.6 FPS on Titan 1080 Ti GPU), demonstrating the effectiveness of proposed SARA in oriented object detection in remote sensing images. As listed in Table 3, our SARA achieves the best performance in many categories: 82.29% in GTF, 80.49% in SV, 83.54% in LV, 89.37% in SH, 88.07% in ST, 78.94% in HA, 80.98% in SP, and 70.13% in HC. The results show that our method performs better with ResNet-50 than ResNet-101, which is similar to S^2A-Net [17]. Compare with state-of-the-art methods, for single scale testing, our method achieves 78.65% in terms of mAP, outperforms F3-Net [1] (76.02%, two-stage) and S^2A-Net (76.11%, one-stage). While, for multi-scale

testing, our method achieves 79.91% in terms of mAP and outperforms the second method S^2A-Net (79.42%).

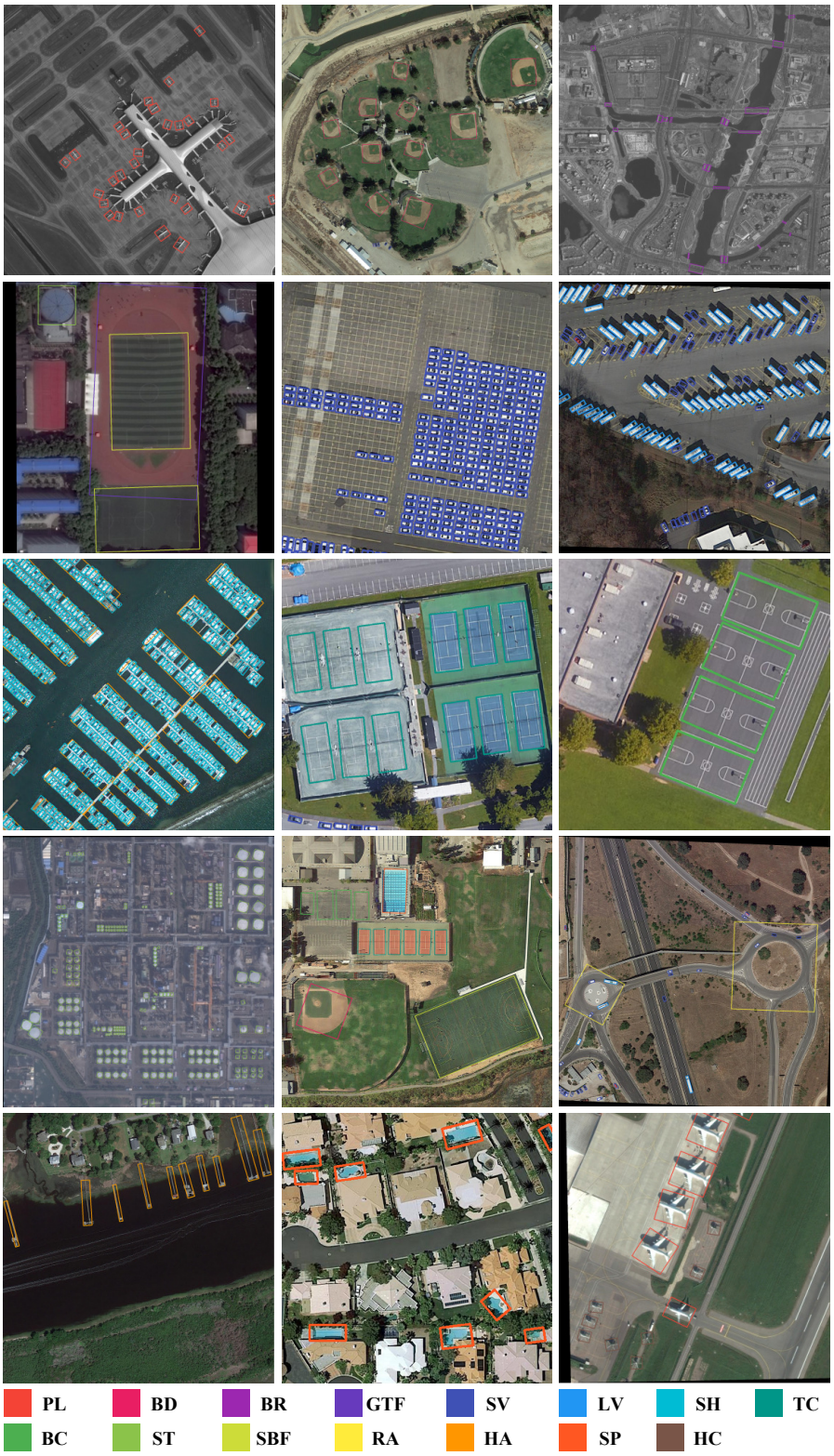


Figure 11. Representative results on DOTA.

Table 3. Comparisons with state-of-the-art methods (both two-stage and one-stage) on DOTA OBB task. R-50, R-101, R-152, and H-104 represents ResNet-50, ResNet-101, ResNet-152, and Hourglass-104, respectively. * means multi-scale testing. Note that the FPS is a relative FPS calculated by the overall inference time and the number of chip images. For FPS, † denotes tested on Titan Xp GPU, ‡ means on V100 GPU, and § represents on Titan 1080 Ti.

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP	FPS
two-stage :																		
FR-O [50]	R-101	79.42	77.13	17.70	64.05	35.30	38.02	37.16	89.41	69.64	59.28	50.30	52.91	47.89	47.40	46.30	54.13	-
Azimi et al. [53]	R-101	81.36	74.30	47.70	70.32	64.89	67.82	69.98	90.76	79.06	78.20	53.64	62.90	67.02	64.17	50.23	68.16	-
RoITransformer * [26]	R-101	88.64	78.52	43.44	75.92	68.81	73.68	83.59	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	69.56	5.9 †
CAD-Net [54]	R-101	87.80	82.40	49.40	73.50	71.10	63.50	76.60	**90.90**	79.20	73.30	48.40	60.90	62.00	67.00	62.20	69.90	-
SCRDet [4]	R-101	**89.98**	80.65	52.09	68.36	68.36	60.32	72.41	90.85	**87.94**	86.86	65.02	66.68	66.25	68.24	65.21	72.61	-
GLS-Net [5]	R-101	88.65	77.40	51.20	71.03	73.30	72.16	84.68	90.87	80.43	85.38	58.33	62.27	67.58	70.69	60.42	72.96	-
Xu et al. [55]	R-101	89.64	85.00	52.26	77.34	73.01	73.14	86.82	90.74	79.02	86.81	59.55	**70.91**	72.94	70.86	57.32	75.02	10.0 †
Mask OBB [6]	R-50	89.61	85.09	51.85	72.90	75.28	73.23	85.57	90.37	82.08	85.05	55.73	68.39	71.61	69.87	66.33	74.86	-
Mask OBB [6]	R-101	89.56	**85.95**	54.21	72.90	76.52	74.16	85.63	89.85	83.81	86.48	54.89	69.64	73.94	69.06	63.32	75.33	-
F^3-Net [1]	R-152	88.89	78.48	54.62	74.43	72.80	77.52	87.54	90.78	87.64	85.63	63.80	64.53	78.06	72.36	63.19	76.02	-
one-stage :																		
RetinaNet [15]	R-101	88.82	81.74	44.44	65.72	67.11	55.82	72.77	90.55	82.83	76.30	54.19	63.64	63.71	69.73	53.37	68.72	12.7 ‡
A^2S-Det [23]	R-50	89.45	78.52	42.78	53.93	76.37	74.62	86.03	90.68	83.35	83.55	48.58	60.51	63.46	71.33	53.10	70.42	-
A^2S-Det [23]	R-101	89.59	77.89	46.37	56.47	75.86	74.83	86.07	90.58	81.09	83.71	50.21	60.94	65.29	69.77	50.93	70.64	-
DRN [52]	H-104	88.91	80.22	43.52	63.35	73.48	70.69	84.94	90.14	83.85	84.11	50.12	58.41	67.62	68.60	52.50	70.70	-
DRN * [52]	H-104	89.71	82.34	47.22	64.10	76.22	74.43	85.84	90.57	86.18	84.89	57.65	61.93	69.30	69.63	58.48	73.23	-
R^3Det [30]	R-101	89.54	81.99	48.46	62.52	70.48	74.29	77.54	90.80	81.39	83.54	61.97	59.82	65.44	67.46	60.05	71.69	-
R^3Det [30]	R-152	89.49	81.17	50.53	66.10	70.92	78.66	78.21	90.81	85.26	84.23	61.81	63.77	68.16	69.83	67.17	73.74	-
S^2A-Net [17]	R-101	88.70	81.41	54.28	69.75	78.04	80.54	88.04	90.69	84.75	86.22	65.03	65.81	76.16	73.37	58.86	76.11	12.7 ‡/9.3 §
S^2A-Net * [17]	R-101	89.28	84.11	56.95	79.21	80.18	82.93	89.21	90.86	84.66	87.61	**71.66**	68.23	78.58	78.20	65.55	79.15	12.7 ‡/9.3 §
S^2A-Net * [17]	R-50	88.89	83.60	**57.74**	81.95	79.94	83.19	89.11	90.78	84.87	87.81	70.30	68.25	78.30	77.01	69.58	79.42	16.0 ‡/11.6 §
ours :																		-
SARA (Ours)	R-101	89.20	84.60	55.94	73.71	79.77	82.03	88.99	90.75	85.70	87.25	62.36	66.61	**78.94**	75.57	67.60	77.93	8.8 §
SARA (Ours) *	R-101	89.24	82.81	57.44	81.21	80.23	**83.54**	89.29	90.75	85.55	**88.07**	69.70	66.11	78.92	75.53	68.62	79.13	8.8 §
SARA (Ours)	R-50	89.13	85.81	54.88	73.39	80.31	81.84	89.07	90.78	87.45	87.02	65.02	66.52	78.38	80.08	**70.13**	78.65	10.9 §
SARA (Ours) *	R-50	89.40	84.29	56.72	**82.29**	**80.49**	83.01	**89.37**	90.67	86.20	87.44	71.34	69.06	78.49	**80.98**	68.98	**79.91**	10.9 §

5.5. Failure Cases

Figure 12 shows some typical failure predictions of our method. Figure 12 (a) shows black small vehicles may be missing since they are similar to the black background. Some small and large vehicles in Figure 12 (b) are also missing due to the low resolution and blur.

(**a**)Missing black small vehicles. (**b**)Missing vehicles due to low resolution and blur.

Figure 12. Some typical failure predictions of our method. Blue circles hight the missing detected objects. Notably, we only show the detect boxes whose score is higher than a threshold of 0.5.

6. Conclusions

In this paper, we propose a novel Self-Adaptive Aspect Ratio Anchor (SARA) for the aspect ratio variations of objects in remote sensing images, which can self-adaptively learn the appropriate aspect ratio for each category and benefit the regression. With the help of SARA, we can only utilize a simple squared anchor (related to the strides of feature maps in Feature Pyramid Networks) to regress objects in variation aspect ratio. We also utilize an Oriented Box Decoder (OBD) to align the feature maps and encode the orientation information of oriented objects. Our method achieves mAP value of 79.91% on the DOTA dataset, which has achieved state-of-the-art.

Author Contributions: Conceptualization, J.-B.H. and X.Z.; methodology, J.-B.H.; software, J.-B.H.; validation, J.-B.H., X.Z., and X.-C.Y.; investigation, J.-B.H.; resources, X.Z. and X.-C.Y.; writing—original draft preparation, J.-B.H.; writing—review and editing, X.Z. and X.-C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Key R&D Program of China (2019YFB1405900), and the Fundamental Research Funds for the Central Universities and USTB-NTUT Joint Research Program

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This author would like to thank the providers of the awesome remote sensing images DOTA dataset. The author would like to express their appreciation to the developers of pytorch and S^2A-Net for their open source code.

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

Abbreviations

The following abbreviations are used in this manuscript:

SARA	Self-Adoptive Aspect Ratio Anchor
OBD	Oriented Box Decoder
mAP	mean Average Precision
HBB	Horizontal bounding boxes
OBB	Oriented bounding boxes
IoU	Intersection-over-union
RoI	Region of Interest

References

1. Ye, X.; Xiong, F.; Lu, J.; Zhou, J.; Qian, Y. F3-Net: Feature Fusion and Filtration Network for Object Detection in Optical Remote Sensing Images. *Remote Sens.* **2020**, *12*, 4027. [CrossRef]
2. Qiu, H.; Li, H.; Wu, Q.; Meng, F.; Ngan, K.N.; Shi, H. A^2RMNet: Adaptively Aspect Ratio Multi-Scale Network for Object Detection in Remote Sensing Images. *Remote Sens.* **2019**, *11*, 1594. [CrossRef]
3. 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]
4. 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 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea, 27 October–2 November 2019; pp. 8231–8240.
5. Li, C.; Luo, B.; Hong, H.; Su, X.; Wang, Y.; Liu, J.; Wang, C.; Zhang, J.; Wei, L. Object Detection Based on Global-Local Saliency Constraint in Aerial Images. *Remote Sens.* **2020**, *12*, 1435. [CrossRef]
6. Wang, J.; Ding, J.; Guo, H.; Cheng, W.; Pan, T.; Yang, W. Mask OBB: A Semantic Attention-Based Mask Oriented Bounding Box Representation for Multi-Category Object Detection in Aerial Images. *Remote Sens.* **2019**, *11*, 2930. [CrossRef]
7. Zhang, X.; Zhu, K.; Chen, G.; Tan, X.; Zhang, L.; Dai, F.; Liao, P.; Gong, Y. Geospatial Object Detection on High Resolution Remote Sensing Imagery Based on Double Multi-Scale Feature Pyramid Network. *Remote Sens.* **2019**, *11*, 755. [CrossRef]
8. Liu, Z.; Wang, H.; Weng, L.; Yang, Y. Ship Rotated Bounding Box Space for Ship Extraction From High-Resolution Optical Satellite Images With Complex Backgrounds. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1074–1078. [CrossRef]
9. Liu, Z.; Hu, J.; Weng, L.; Yang, Y. Rotated region based CNN for ship detection. In Proceedings of the 2017 IEEE International Conference on Image Processing, ICIP 2017, Beijing, China, 17–20 September 2017; pp. 900–904.
10. Kwan, C.; Chou, B.; Yang, J.; Rangamani, A.; Tran, T.D.; Zhang, J.; Etienne-Cummings, R. Deep Learning-Based Target Tracking and Classification for Low Quality Videos Using Coded Aperture Cameras. *Sensors* **2019**, *19*, 3702. [CrossRef]
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision - ECCV 2016—14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
12. Nguyen, P.H.; Arsalan, M.; Koo, J.H.; Naqvi, R.A.; Truong, N.Q.; Park, K.R. LightDenseYOLO: A Fast and Accurate Marker Tracker for Autonomous UAV Landing by Visible Light Camera Sensor on Drone. *Sensors* **2018**, *18*, 1703. [CrossRef]
13. Ren, S.; He, K.; Girshick, R.B.; 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]
14. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
15. Lin, T.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 2999–3007.
16. 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]
17. Han, J.; Ding, J.; Li, J.; Xia, G. Align Deep Features for Oriented Object Detection. *arXiv* **2020**, arXiv:2008.09397.
18. Tang, T.; Liu, Y.; Zheng, Y.; Zhu, X.; Zhao, Y. Rotating Objects Detection in Aerial Images via Attention Denoising and Angle Loss Refining. *DEStech Trans. Comput. Sci. Eng.* **2020**.
19. Yang, X.; Yan, J.; Yang, X.; Tang, J.; Liao, W.; He, T. SCRDet++: Detecting Small, Cluttered and Rotated Objects via Instance-Level Feature Denoising and Rotation Loss Smoothing. *arXiv* **2020**, arXiv:2004.13316.
20. Li, C.; Xu, C.; Cui, Z.; Wang, D.; Jie, Z.; Zhang, T.; Yang, J. Learning Object-Wise Semantic Representation for Detection in Remote Sensing Imagery. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 20–27.
21. Fu, K.; Chang, Z.; Zhang, Y.; Xu, G.; Zhang, K.; Sun, X. Rotation-aware and multi-scale convolutional neural network for object detection in remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2020**, *161*, 294–308. [CrossRef]
22. Li, Y.; Huang, Q.; Pei, X.; Jiao, L.; Shang, R. RADet: Refine Feature Pyramid Network and Multi-Layer Attention Network for Arbitrary-Oriented Object Detection of Remote Sensing Images. *Remote Sens.* **2020**, *12*, 389. [CrossRef]
23. Xiao, Z.; Wang, K.; Wan, Q.; Tan, X.; Xu, C.; Xia, F. A2S-Det: Efficiency Anchor Matching in Aerial Image Oriented Object Detection. *Remote Sens.* **2021**, *13*, 73. [CrossRef]

24. Ming, Q.; Zhou, Z.; Miao, L.; Zhang, H.; Li, L. Dynamic Anchor Learning for Arbitrary-Oriented Object Detection. *arXiv* **2020**, arXiv:2012.04150.
25. Zhou, Y.; Ye, Q.; Qiu, Q.; Jiao, J. Oriented Response Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 4961–4970.
26. 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, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 2849–2858.
27. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
28. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
29. Liao, M.; Shi, B.; Bai, X. Textboxes++: A single-shot oriented scene text detector. *IEEE Trans. Image Process.* **2018**, *27*, 3676–3690. [CrossRef] [PubMed]
30. Yang, X.; Liu, Q.; Yan, J.; Li, A. R3Det: Refined Single-Stage Detector with Feature Refinement for Rotating Object. *arXiv* **2019**, arXiv:1908.05612.
31. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
32. Huang, L.; Yang, Y.; Deng, Y.; Yu, Y. DenseBox: Unifying Landmark Localization with End to End Object Detection. *arXiv* **2015**, arXiv:1509.04874.
33. Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; Liang, J. EAST: An Efficient and Accurate Scene Text Detector. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 2642–2651.
34. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T.S. UnitBox: An Advanced Object Detection Network. In Proceedings of the 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
35. He, W.; Zhang, X.Y.; Yin, F.; Liu, C.L. Deep Direct Regression for Multi-oriented Scene Text Detection. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 745–753.
36. He, W.; Zhang, X.Y.; Yin, F.; Liu, C.L. Multi-oriented and multi-lingual scene text detection with direct regression. *IEEE Trans. Image Process.* **2018**, *27*, 5406–5419. [CrossRef] [PubMed]
37. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea, 27 October–2 November 2019; pp. 9626–9635.
38. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
39. Zhu, C.; He, Y.; Savvides, M. Feature Selective Anchor-Free Module for Single-Shot Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 840–849.
40. Lin, T.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
41. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. In Proceedings of the Computer Vision-ECCV 2018—15th European Conference, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11218, pp. 765–781.
42. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea, 27 October 27–2 November 2019; pp. 6568–6577.
43. Zhou, X.; Zhuo, J.; Krähenbühl, P. Bottom-Up Object Detection by Grouping Extreme and Center Points. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 850–859.
44. Wei, H.; Zhou, L.; Zhang, Y.; Li, H.; Guo, R.; Wang, H. Oriented Objects as pairs of Middle Lines. *arXiv* **2019**, arXiv:1912.10694.
45. Deng, D.; Liu, H.; Li, X.; Cai, D. PixelLink: Detecting Scene Text via Instance Segmentation; In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 6773–6780.
46. Xu, Y.; Wang, Y.; Zhou, W.; Wang, Y.; Yang, Z.; Bai, X. TextField: Learning a Deep Direction Field for Irregular Scene Text Detection. *IEEE Trans. Image Process.* **2019**, *28*, 5566–5579. [CrossRef] [PubMed]
47. Tian, Z.; Shu, M.; Lyu, P.; Li, R.; Zhou, C.; Shen, X.; Jia, J. Learning Shape-Aware Embedding for Scene Text Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 4234–4243.
48. Wang, W.; Xie, E.; Li, X.; Hou, W.; Lu, T.; Yu, G.; Shao, S. Shape Robust Text Detection with Progressive Scale Expansion Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 16–20 June 2019; pp. 9336–9345.
49. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 764–773.

50. Xia, G.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.J.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3974–3983.
51. 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 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
52. Pan, X.; Ren, Y.; Sheng, K.; Dong, W.; Yuan, H.; Guo, X.; Ma, C.; Xu, C. Dynamic Refinement Network for Oriented and Densely Packed Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; pp. 11204–11213.
53. 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 Computer Vision—ACCV 2018—14th Asian Conference on Computer Vision, Perth, Australia, 2–6 December 2018; Volume 11363, pp. 150–165.
54. 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]
55. 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. *arXiv* **2019**, arXiv:1911.09358.