remote sensing

MDPI

Article

A^2S-Det: Efficiency Anchor Matching in Aerial Image Oriented Object Detection

Zhifeng Xiao [1], Kai Wang [1,*], Qiao Wan [1], Xiaowei Tan [1], Chuan Xu [2] and Fanfan Xia [1]

[1] State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; xzf@whu.edu.cn (Z.X.); wanqiao@whu.edu.cn (Q.W.); Tanxiaowei@whu.edu.cn (X.T.); xiafanfan2016@whu.edu.cn (F.X.)

[2] School of Computer Science, Hubei University of Technology, Wuhan 430064, China; chuanxu@whu.edu.cn

* Correspondence: whu-wk@whu.edu.cn

Abstract: Object detection is a challenging task in aerial images, where many objects have large aspect ratios and are densely arranged. Most anchor-based rotating detectors assign anchors for ground-truth objects by a fixed restriction of the rotation Intersection-over-Unit (IoU) between anchors and objects, which directly follow horizontal detectors. Due to many directional objects with a large aspect ratio, the object-anchor IoU is heavily influenced by the angle, which may cause few anchors assigned for some ground-truth objects. In this study, we propose an anchor selection method based on sample balance assigning anchors adaptively, which we name the Self-Adaptive Anchor Selection (A^2S-Det) method. For each ground-truth object, A^2S-Det selects a set of candidate anchors by horizontal IoU. Then, an adaptive threshold module is adopted on the set of candidate anchors, which calculates a boundary of these candidate anchors aiming to keep a balance between positive and negative anchors. In addition, we propose a coordinate regression of relative reference (CR^3) module to precisely regress the rotating bounding box. We test our method on a public aerial image dataset, and prove better performance than many other one-stage detectors and two-stage detectors, achieving the mAP of 70.64. An efficiency anchor matching method helps the detector achieve better performance for objects with large aspect ratios.

Keywords: aerial image; rotation object detection; anchor-based; anchor selection

Citation: Xiao, Z.; Wang, K.; Wan, Q.; Tan, X.; Xu, C.; Xia, F. A^2S-Det: Efficiency Anchor Matching in Aerial Image Oriented Object Detection. *Remote Sens.* **2021**, *13*, 73. https://dx.doi.org/10.3390/rs13010073

Received: 17 November 2020
Accepted: 22 December 2020
Published: 27 December 2020

1. Introduction

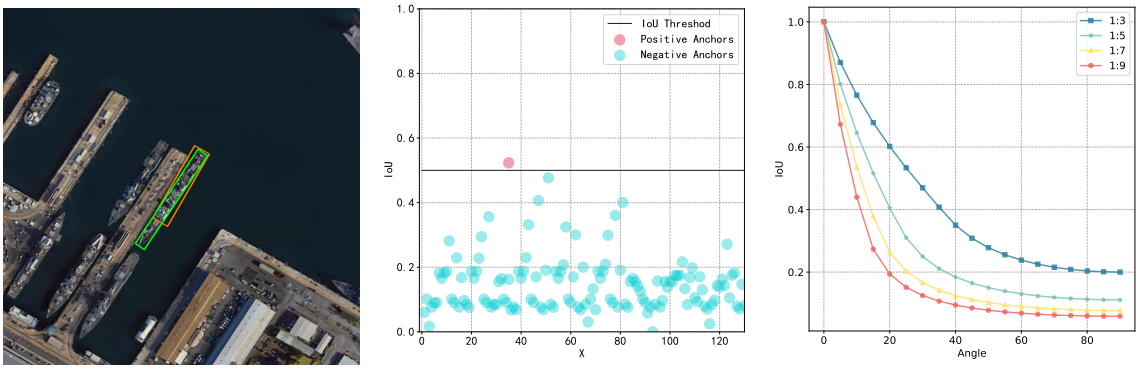
Object detection is a crucial task in aerial image information extraction. Unlike natural images, objects in aerial images may be densely arranged, in any direction and with large aspect ratios, which make it very challenging to precisely detect objects in aerial images. The scene where objects are densely arranged requires detectors to locate and identify a category precisely for each object. In addition, objects may be in any direction, which may cause missing objects in the dense scene by the postprocessing of non-maximum suppression(NMS). Objects with large aspect ratios make it hard to extract features and predict the bounding box.

For objects in any direction, many rotating detectors based on common object detectors are proposed to detect objects as rotating rectangles, which originate from text detection, such as $RRPN$ [1] and R^2CNN [2]. Furthermore, the rotating object detection partly solves the problem of missing some objects caused by NMS where objects are densely arranged in aerial images. In the common detection, detectors are divided into two-stage detectors and one-stage detectors, where it has been generally thought that two-stage detectors perform better while one-stage detectors are faster. $RRPN$ [1] and R^2CNN [2] are both two-stage detectors. Considering the large amount of aerial images, the speed of detectors are also important. Recent evidences suggest that the one-stage detector also shows great potential in aerial image rotating object detection [3]. The Anchor-based method and the anchor-free method are two main approaches to define positive samples and negative

samples. Anchor-based detectors adopt preset rectangles of different shapes in each feature point and assign these positive anchors to the corresponding ground-truth boxes by some certain rules. Anchor-free detectors define samples by points, grids, or other rules. Without preset anchors, anchor-free detectors save time in the label assignment process, but for objects densely arranged in aerial images, anchor-based detectors with dense anchors may be a good choice compared with anchor-free detectors. The characteristics of objects in aerial image may cause these challenges:

Samples are hard to be defined. Most anchor-based rotating detectors assign anchors by a restriction of the rotating IoU. As shown in Figure 1c, a small-angle deviation may lead to low IoU for objects with large aspect ratio, which may cause few anchor assigned for ground-truth objects.

Bounding boxes are hard to be regressed precisely. The sensitive rotating IoU of a large aspect ratio object means the predicted rotating bounding box must be very precise when using the rotating IoU as the evaluation method, compared with horizontal detectors.



(**a**) Object and Selected Anchors (**b**) The IoU of Object-Anchors (**c**) The Effect of Angle on IoU

Figure 1. Aspect ratio heavily affects the anchor selection process while *RetinaNet* is applied to rotating object detection. (**a**) visualizations of the ground-truth object and selected anchors; (**b**) the IoU distribution of object-anchors and anchors selection process; (**c**) the effect of angle on IoU for objects with different aspect ratios.

As shown in Figure 1a, the ship object only selects one anchor as the positive anchor by the fixed restriction of IoU, where the fixed restriction is set to 0.5. Figure 1b shows the IoU distribution of top-k anchors, which are divided into positive anchors and negative anchors by the IoU threshold. The IoU between anchors and objects with large aspect ratios is sensitive to the angle deviation. As shown in Figure 1c, the IoU between the box and rotating box with larger aspect ratios is smaller while the angle deviation is the same. In the anchor selection process, anchors are generated following certain rules, where there may be some deviation of location, box size, and angle from ground-truth objects. The difficulty of anchor selection in rotating object detection may cause inadequate training for objects with large aspect ratios.

In this paper, we discuss that the missing matching and low matching ratio between anchors and objects are two of the influencing factors for detector training, especially for objects with large aspect ratios. To solve these, we propose an anchor selection method based on sample balance to improve the anchor selection process, which consists of three modules. Firstly, candidate anchors are selected by a self-adaptive anchor selection module based on horizontal IoU, which will be divided into positive and negative anchors by the statistical threshold of rotating IoU. For the statistical threshold, a self-adaptive threshold module is devised to determine a threshold that keeps a balance between positive and negative anchors according to the rotating IoU in the set of candidate anchors. Finally, we design a coordinate regression of the relative reference module to predict the rotating bounding box precisely. In this module, there is some improvement in the coordinate regression and angle regression for rotating objects.

The contributions of this work are concluded as follows:

- We propose an anchor selection method combining horizontal features and rotating features. For the set of candidate anchors, a self-adaptive threshold module based on sample balance is adopted to determine a threshold, which divides these candidate anchors into positive and negative anchors. There are larger improvements in DOTA [4] compared with the baseline for objects with large aspect ratios.
- For bounding box prediction, the coordinate regression of the relative reference module can predict box more precisely and be to the benefit of more rigorous evaluations, like $AP^{IoU=0.75}$.

2. Materials and Methods

2.1. Data

DOTA [4] is a large dataset for object detection in aerial images, which contains 2806 aerial images from different sensors and with different resolutions. The image size ranges from around 800 × 800 to 4000 × 4000 pixels. This dataset consists of 15 categories, including *Plane, Ship, Bridge, Harbor, Baseball Diamond (BD), Ground Track Field (GTF), Small Vehicle(SV), Large Vehicle(LV), Tennis Court(TC), Basketball Court(BC), Storage Tank(ST), Soccer Ball Field(SBF), Roundabout (RA), Swimming Pool(SP)*, and *Helicopter (HC)*. There are 188,282 instances in this dataset, which are labeled respectively by horizontal bounding boxes and rotating bounding boxes. The dataset is officially spilled into three parts of training, validation, test. We merge these two sets of training and validation for training and test on the test dataset by the official evaluation server. In the training process, we divide these images into 600 × 600 sub-images, whose overlaps are 200 pixels. Those sub-images without any objects are discarded directly. Finally, there are 30,250 sub-images for training.

Furthermore, we adopt online data augmentation if data augmentation is needed. The data augmentation method includes random rotation and random flip, whose probabilities of occurrence are both 50%. For the random rotation, the rotation angle randomly generates from 0 to 360° by the step of 15°.

2.2. Related Work

Two-Stage Detectors. R-CNN [5] divides the detection process into the region proposal stage and regression stage, which develops a two-stage detector. Fast R-CNN is proposed to solve the problem of a heavy calculation burden by generating region proposals in the feature map. Two-Stage detectors consist mainly of region proposal network(RPN) and convolutional neural networks(CNN) from Faster C-NN [6]. The RPN module generates many proposals to distinguish foreground and background by the score, such as 0.7. Then, the CNN module randomly selects positive and negative proposals in a ratio of $(3/1)$ for training. In the inference process, the RPN module generates quantitative proposals and the CNN module predicts the categories and the bounding boxes from these proposals, which avoids lots of windows and reduces computation. Many worthy two-stage detection method are proposed from then, such as Mask R-CNN [7], FPN [8], OHEM [9], Context-Aware [10], etc.

One-Stage Detectors. Although the two-stage detectors are faster than before. The speed is still so slow to satisfy the needs of real-time detection. Different from two-stage detectors, one-stage detectors define positive and negative samples based on points in the feature map, whose inference speed is faster. Unlike two-stage detectors like YOLO [11], which compute image features for anchors of each feature point, anchors make use of the feature maps and compute class probability and bounding box based on the corresponding anchor. There is a weakness that negative samples are far more than positive samples, which makes the training process difficult. To solve this problem, RetinaNet [12] proposed the focal loss to keep a balance of loss between positive loss and negative loss, which makes it possible to train a one-stage high-detector with high performance. Many one-stage detectors are proposed today, such as FCOS [13], CenterNet [14], etc.

Rotation Detectors. Rotation target detection is derived from text detection. RRPN [1] proposed a rotation region proposal network based on Faster R-CNN [6] to detect text with direction. RRPN defines a rotation box as (x, y, w, h, θ) while Faster R-CNN [6] defines a horizontal box as (x, y, w, h). (x, y) indicates the centroid coordinates of rotation box. (w, h) indicates the width and height of rotation box. θ indicates the direction of rotation box relative to the horizontal coordinate system. R^2CNN [2] defines rotation box as $(x_1, y_1, x_2, y_2, w, h)$. (x_1, y_1) and (x_2, y_2) mean the coordinates of the first two points. (w, h) is the same as definition in RRPN [1]. In addition, R^2CNN [2] proposed a special RoI Pooling method, whose pooled size are 7×7, 3×11 and 11×3. There are many other outstanding rotation detectors applicating in text detection, such as EAST [15], DRBOX [16], TextBoxes++ [17], etc. In aerial image target detection, many high-performance detectors are proposed. RoI transfromer [18] proposed an RRoI learner to learn direction information from horizontal anchors, which reduces the amount of computation compared with the RRPN module. R3Det [3] proposed a feature refinement module to reconstruct the feature map, which achieves the purpose of feature alignment. In R3Det [3], it is proved that one-stage detectors also have huge potential in aerial image target detection. Many rotation detectors aim at how to define rotation boxes and how to define samples. For instance, Axis-learning [19] predicts the axis of rotation objects based on the idea of anchor free, which has good performance and inference fast. O2-DNet [20] define boxes as two middle lines and the intersection point of middle lines. There are many other high-performance rotation detectors, such as SCRDet [21], Gliding Vertex [22], CenterMap OBB [23], etc.

Label Assignment. ATSS [24] discusses that anchor-based method(RetinaNet [12]) perform as well as anchor free method(FCOS [13]) if the method of sample definition is similar. What affects the performance is the method to define positive and negative samples instead of how to regress the box. It matters little whether regressions by anchors or by points. Therefore, ATSS [24] proposed an adaptive training sample selection method defining samples by a dynamic threshold. FreeAnchor [25] proposed that the largest k IoU of anchors are potential positive samples. When calculating loss, each anchor has a weight which determines the regression effect. At the beginning of training, all anchors have similar weights due to a bad regression effect, but as the training goes on, some anchors regress well and weights improve. At the ending of training, there are only several anchors whose weights are far beyond other anchors. In brief, FreeAnchor [25] defines positive and negative samples by prediction, which is a special method of label assignment. MAL [26] proposed a multiple anchor learning method to assess positive anchors in the anchor bag selected by IoU. The assessment method combines the score of classification and location. PISA [27] indicates that what affects performance most is not the hard sample, but the prime sample.

2.3. Method

We adopt backbone, feature pyramid network, and detector head as basic structures. Similar to RetinaNet [12], there are several rotation anchors for each point in the feature map, which are in charge of predicting objects. For the detector head, we propose a coordinate regression module based on the relative reference module. In the training process, we propose a self-adaptive anchor selection module to define positive anchors and negative anchors, which can keep a balance between these anchors. In general, our work focuses on the training process and rotation detector head.

2.3.1. Network Architecture

Our main network architecture uses ResNet architecture with a Feature Pyramid Network backbone to extract rich, multi-scale, directional feature information from images. As shown in Figure 2, ResNet generates C_3, C_4 and C_5, which denote P_3 to P_7 of feature pyramid network. Furthermore, P_3 to P_7 are the feature levels for prediction, whose feature map size are the down-sampling ratio of $(8, 16, 32, 64, 128)$ times of input image. In this paper, all input images are resized into 800×800. There are two subnetworks in charge

of predicting category and bounding box for each P_i, where $i = 3, 4, 5, 6, 7$. The final feature map of category prediction branch predicts K values representing K categories for each anchor in each feature point where there are A anchors in each feature point. The predicting values are transformed into the probability of each category by *Sigmoid* function. In the bounding box predicting branch, the final feature map predicts a tuple of $(\delta x, \delta y, \delta w, \delta h, \delta\theta)$ representing the deviation relative to the anchor, which needs to be decode into (x, y, w, h, θ). In addition, two subnetworks share the same parameter weights of all feature levels, which greatly reduces computation. The network architecture is almost the same as RetinaNet [12], except using rotating anchors and predicting a tuple of 5 values in the output feature map of the bounding box predicting branch. The details of implementation for those differences are represented in Sections 2.3.4 and 2.3.6.

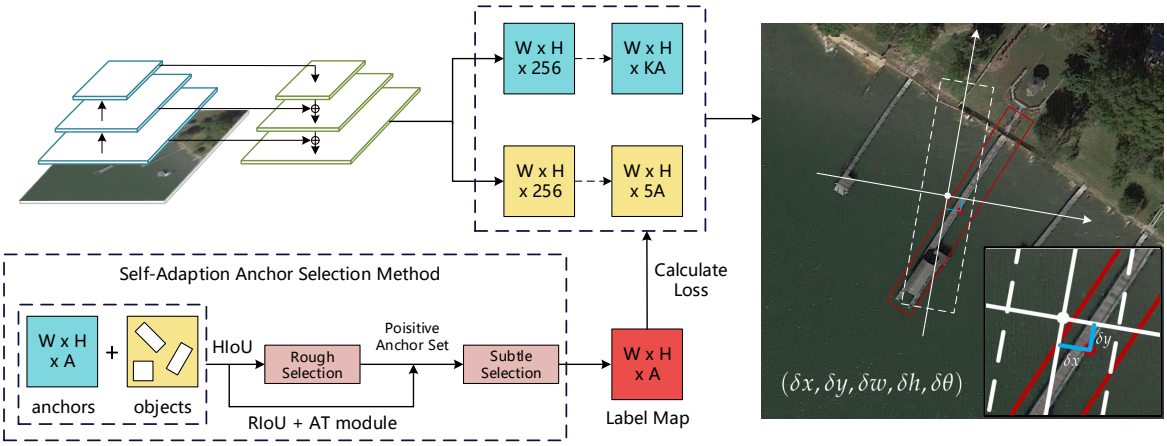


Figure 2. The main structure of our detector and training process. (1) A in this figure means there are A rotation anchors in each feature point and K means there are K categories needing to be predicted; (2) The values of points in label map mean which target this anchor matches; (3) The positive anchor set is selected by a fixed threshold of horizontal IoU(referred to as $HIoU$), which is a rough selection; (4) AT module calculates the threshold based on the distribution of the rotating IoU(referred to as $RIoU$) and the final positive anchors are selected by this threshold in the subtle selection process; (5) $(\delta x, \delta y, \delta w, \delta h, \delta\theta)$ are the deviations between anchors and objects, which need to be decoded.

2.3.2. Self-Adaptive Anchor Selection

The baseline rotating detector derives from RetinaNet and has an imbalance performance on objects with different aspect ratios. Due to the inflexible anchor selection process, objects with large aspect ratios match fewer anchors for training. We propose a self-adaptive anchor selection method which can define anchors adaptively. As shown in Figure 3, there are many anchors for the object and these anchors are disorderly. Candidate anchors are selected by horizontal IoU, which ensures that the horizontal features correspond. Then, positive anchors are selected by rotating IoU, using an adaptive threshold computed by the AT module.

In this section, the self-adaptive anchor selection method combines horizontal features and rotating features. As shown in Algorithm 1, there are the set of ground-truth boxes($\mathcal{G}$) on the image and the set of anchors($\mathcal{A}$) for all feature map. In the training process, each anchor in $\mathcal{A}$ is either assigned to one of the ground-truth boxes($\mathcal{G}$) or defined as a negative anchor. Firstly, we compute the horizontal IoU and the rotating IoU between $\mathcal{A}$ and $\mathcal{G}$, represented as HD and RD. For each anchor, the ground-truth box which has the max rotating IoU is assigned to this anchor, ensuring only one ground-truth box for an anchor. Secondly, a set of candidate anchors is selected by a conditional inequality($HD_g \geq 0.6$) for each ground-truth box(g). Thirdly, we compute a threshold T_g based on a statistical method to distinguish the candidate anchors. The function of self-adaptive computing T_g is described in Section 2.3.3 and discussed in Section 4.1. For each candidate anchor(d),

the anchor is assigned to the ground-truth box(g) and defined as a positive anchor if $RIoU(d,g) \geq T_g$. Finally, a set of positive anchors(P) is selected by this algorithm and the rest of the anchors are defined as negative anchors(N).

Algorithm 1 Self-Adaptive Anchor Selection

Require:
$\mathcal{G}$ is a set of ground-truth boxes on the image
$\mathcal{L}$ is the number of feature pyramid levels
$\mathcal{A}$ is a set of all anchor boxes

Ensure:
$\mathcal{P}$ is a set of positive samples
$\mathcal{N}$ is a set of negative samples

compute $\mathcal{RIOU}$ between $\mathcal{A}$ and G: $\mathcal{RD} = RIoU(\mathcal{A}, G)$;
build an empty set for candidate positive samples of the ground-truth G: $\mathcal{C} \leftarrow Max(RD)$;
for each ground-truth $g \in \mathcal{G}$ **do**
  compute $\mathcal{HIOU}$ between $\mathcal{C}_g$ and g: $\mathcal{HD}_g = HIoU(\mathcal{C}_g, g)$;
  remove those anchors whose $\mathcal{HIOU}$ are less than 0.6: $\mathcal{C}_g$ where $\mathcal{HD}_g \geq 0.6$;
  $\mathcal{D}_g = \mathcal{C}_g \cap \mathcal{RD}_g$;
  compute threshold $\mathcal{T}_g$ on the basis of statistical method: $\mathcal{T}_g = Fun(\mathcal{D}_g)$;
  for each candidate $d \in \mathcal{D}_g$ **do**
    if $RIoU(d,g) \geq \mathcal{T}_g$ **then**
      $\mathcal{P} = \mathcal{P} \cup d$;
    end if
  end for
end for
$\mathcal{N} = \mathcal{A} - \mathcal{P}$;
return $\mathcal{P}, \mathcal{N}$;

Moreover, we define horizontal IoU and rotating IoU as Figure 4. For the horizontal IoU, the rotating box is transformed into the horizontal box based on the vertexes of the rotating box and the horizontal IoU is computed based on the horizontal boxes. For the rotating IoU, the computing method is the same as the common IoU, using the rotating intersection area and the rotating box area.

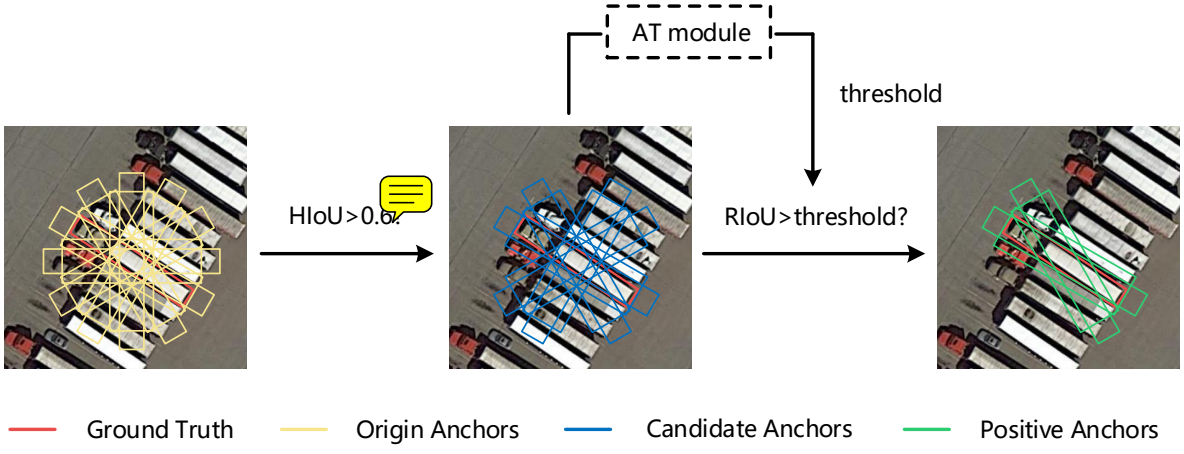
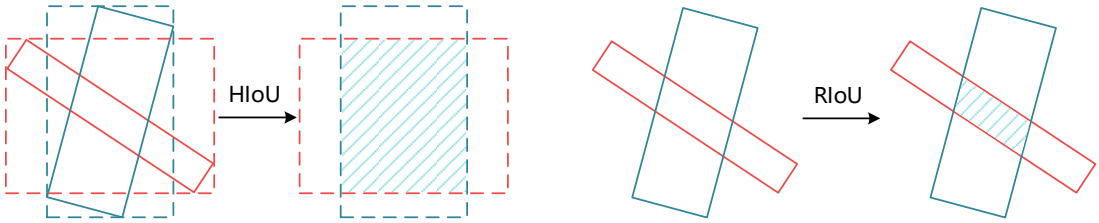


Figure 3. The visualization of the anchor selection process. $HIoU$ and $RIoU$ are described in Figure 4. AT module is described in Section 2.3.3.



(**a**) Horizontal IoU (**b**) Rotating IoU

Figure 4. For the degree of overlapping between rotating anchors and objects, there are two evaluation methods. (**a**) the visualization of horizontal IoU; (**b**) the visualization of rotating IoU.

2.3.3. Self-Adaptive Threshold Based on Sample Balance

In Section 2.3.2, we discuss the algorithm flow of the self-adaptive anchor selection method. There is a key problem, which is how to compute the threshold(T_g) adaptively. Functions based on statistical methods are reasonable. Statistical data varies from different samples. Mean value and standard deviation are common statistical parameters, whose combination is a common method to describe the normal distribution. In this section, we discuss how to describe the distribution of positive and negative samples correctly. In general, anchors are divided into positive anchors and negative anchors by IoU, and they can be formulated as:

$$A_i \; is \begin{cases} positive^+ & T_g \leq RIoU(A_g, g), \; and \; A_i \in C_g \\ negative^- & others. \end{cases} \tag{1}$$

Here, $RIoU(A_g, g)$ means the rotating IoU between the anchor(A_i) and the object(g). T_g is a key parameter to divide positive anchors and negative anchors. $(Mean + Std)$ may be an effective method to divide anchors while IoU between anchors and objects is a normal distribution. For those objects with large aspect ratios, the distribution of IoU may be random. The adaptive anchor selection method aims at finding a rotating IoU boundary to divide these anchors into two sets and keeping a balance between positive anchors and negative anchors. The algorithm can be described as:

$$arg \; min \; F(T_g) \tag{2}$$

$$s.t. \quad \begin{cases} F(T_g) = |Std(C_1) - Std(C_2)|, \\ c \leq T_g, \; and \; c \in C_1 \; , \\ c > T_g, \; and \; c \in C_2 \; , \\ C_1 \cup C_2 = C_g. \end{cases} \tag{3}$$

The Formula (2) and (3) describes this problem as an optimization problem. The key problem is how to define sample balance and solve T_g. The standard deviation reflects the dispersion degree of data. In this formula, the standard deviation is adopted to describe the stability of the rotating IoU. The optimization goal is to minimize $|Std(C_1) - Std(C_2)|$, which represents the balance degree between these two sets(C_1, C_2). The solving process of T_g will be complicated if a precise solution is needed. Considering both speed and effectiveness, this part adopts an estimation method to compute a rough T_g:

$$T_g \in (0, step, 1), \; and \; step = 0.01. \tag{4}$$

Then, the algorithm can be simplified as:

$$arg\ min\ F(T_g) \tag{5}$$

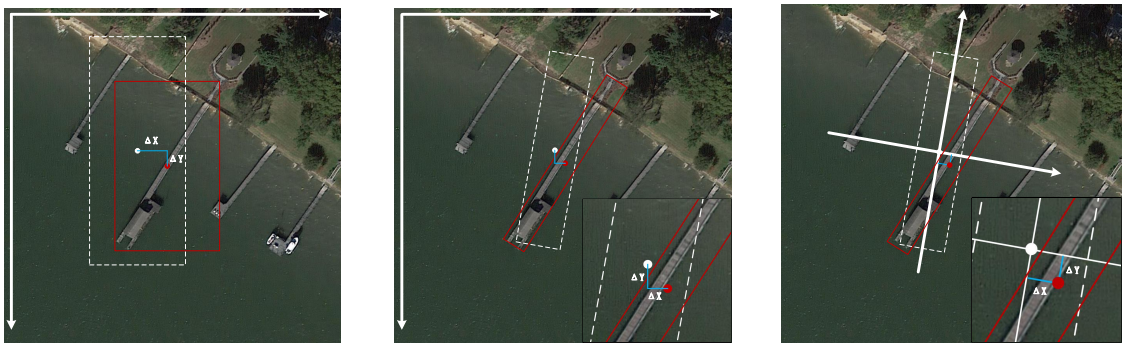
$$s.t. \quad \begin{cases} F(T_g) = |Std(C_1) - Std(C_2)|, \\ c \leq T_g,\ and\ c \in C_1\ , \\ c > T_g,\ and\ c \in C_2\ , \\ C_1 \cup C_2 = C_g, \\ T_g \in (0, step, 1),\ and\ step = 0.01. \end{cases} \tag{6}$$

2.3.4. Coordinate Regression of Relative Reference

In common detector, the box regression method is shown as (a) in Figure 5. This box regression method of the horizontal detector is used in most rotating detectors, which is shown as (b) in Figure 5. The box encoding method can be described as:

$$\begin{aligned} t_x &= \frac{x - x_a}{w_a} & t_y &= \frac{y - y_a}{y_a} \\ t_w &= \ln(\frac{w - w_a}{w_a}) & t_h &= \ln(\frac{h - h_a}{h_a}) \\ t_\theta &= \frac{\theta - \theta_a}{180}\pi \end{aligned} \tag{7}$$

(x, y, w, h, θ) represents bounding box of the object while $(x_a, y_a, w_a, h_a, \theta_a)$ represents bounding box of the anchor. For objects, (x, y) is the centroid coordinates of bounding box. (w, h) represents width and height. θ represents the rotation angle of bounding box. $(t_x, t_y, t_w, t_h, t_\theta)$ is values which we want to regress precisely, representing the offsets relative to the corresponding anchor.



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

Figure 5. (**a**) the encode method of common detection; (**b**) the encode method of rotation detection which is similar to horizontal detection; (**c**) the encode method we propose based on relative reference.

The edges of the box are parallel to the axies of the image in horizontal detection while there is an angle between the rotating box and axies of the image in rotating detection. Therefore, the coordinate regression method of horizontal detection can not well describe the relation between $(\delta x, \delta y)$ and rotating IoU in rotating detection. To solve this problem, we propose a coordinate regression method based on relative reference. The new coordinate encoding method is shown as:

$$\begin{aligned} \delta x &= (x - x_a)cos(\theta_a) - (y - y_a)sin(\theta_a),\ \ t'_x = \frac{\delta x}{w_a} \\ \delta y &= (y - y_a)cos(\theta_a) + (x - x_a)sin(\theta_a),\ \ t'_y = \frac{\delta y}{h_a} \end{aligned} \tag{8}$$

We establish the coordinate system which takes (x_a, y_a) as the origin. The X-axis and the Y-axis are separately parallel to the width and height of the anchor. The new coordinate system and $(\delta x, \delta y)$ are shown in Figure 5c. In the inference process, the corresponding coordinate decoding method can be described as:

$$\begin{aligned} \delta x &= reg_x\ w_a\ cos(\theta_a) + reg_y\ h_a\ sin(\theta_a), \quad pred_x = \delta x + x_a \\ \delta y &= reg_y\ h_a\ cos(\theta_a) - reg_x\ h_a\ sin(\theta_a), \quad pred_y = \delta y + y_a \end{aligned} \tag{9}$$

(reg_x, reg_y) represents the coordinate for network output, which needs to be decoded to $(pred_x, pred_y)$. The angle (θ) of the rotating box also needs to be predicted. The angle is usually defined in $[-90, 90]$, which may cause ambiguity in the border. For instance, the $\delta\theta$ between $-89°$ and $89°$ should be $2°$ instead of $178°$. The new theta encoding method is shown as:

$$t'_\theta = \begin{cases} (\theta - \theta_a) + 180, & (\theta - \theta_a) \in (-180°, -90°) \\ (\theta - \theta_a) + 180, & (\theta - \theta_a) \in (90°, 180°) \\ \theta - \theta_a, & (\theta - \theta_a) \in (-90°, 90°) \end{cases} \tag{10}$$

Combined with the formula of (3), (4) and (6), the box regression method based on relative reference can be described as:

$$\begin{aligned} t_x &= t'_x & t_y &= t'_y \\ t_w &= \ln(\frac{w - w_a}{w_a}) & t_h &= \ln(\frac{h - h_a}{h_a}) \\ t_\theta &= \frac{t'_\theta}{180}\pi \end{aligned} \tag{11}$$

2.3.5. Loss

The loss function consists of classification loss and regression loss. Classification loss calculates loss of all anchors, including positive anchors and negative anchors. Regression loss calculates loss only for those positive anchors. It can be formulated as:

$$L = \frac{\lambda_1}{N_{pos}} \cdot L_{reg}\{positive\} + \frac{\lambda_2}{N_{pos}} \cdot L_{cls} \tag{12}$$

L_{cls} and L_{reg} are classification loss and regression loss. The classification loss is $focalloss$ while the regression loss is Smooth L1 loss. (λ_1, λ_2) represents the weight of L_{cls} and L_{reg}, which are hyperparameters. N_{pos} is the number of positive anchors. $(t_x, t_y, t_w, t_h, t_\theta)$ is the input parameter of Smooth L1, which can be formulated as:

$$SmoothL1\{positive\} = \begin{cases} \sum 0.5 * (\Delta_i - \Delta_i^{pred})^2 & |\Delta_i - \Delta_i^{pred}| \le \beta,\ i = t_x, t_y, t_w, t_h, t_\theta \\ \sum(|\Delta_i - \Delta_i^{pred}| - 0.5 * \beta) & others,\ i = t_x, t_y, t_w, t_h, t_\theta. \end{cases} \tag{13}$$

2.3.6. Implementation Details

The code of the proposed method is implemented with PyTorch [28] and based on RetinaNet [12]. For some rotating modules, we refer to RRPN [1]. In this paper, we adopt ResNet-50 and ResNet-101 as the backbone network with the initialization of the pre-trained model. There are two Nvidia GeForce RTX 2080 Ti GPUs with 11G memory for experiments. We train the model for 24 epoch, about 90 k iterations on DOTA. Stochastic gradient descent(SGD) is adopted to train the model. The learning rate is initialized at 0.01 and decays to 10% of the current learning rate at the 60 k and 82.5 k learning rate decay steps. The weight decay and momentum are 0.001 and 0.9. In the anchor generation process, the aspect ratios are set to $(1/1, 3/1, 5/1)$. The anchor angles are set to $(60°, 30°, 0, -30°, -60°, -90°)$. The anchor scales are set to (0.2, 4), which means the anchor

scales are ($0.2^{1/4}$, $0.2^{2/4}$, $0.2^{3/4}$, $0.2^{4/4}$). There are 72 anchors for each feature point and 960k anchors totally. Loss parameters are the same as RetinaNet [12], including focal loss and smooth L1 loss. In the inference and evaluation stage, the prediction is judged to be correct if the confidence score is greater than 0.1. Furthermore, the threshold of Non-Maximum Suppression(NMS) is set to 0.15 for each category.

3. Results

A^2S-Det is compared with other rotating detectors in Table 1 under the evaluation method of $AP^{IoU=0.5}$. In the inference process, we divide test images into 600 with the overlap of 200, referring to R3Det [3]. A^2S-Det has better performance than most detectors in Table 1, including one-stage detectors [3,12,19,20,29–31] and two-stage detectors [1,2,6,18,32]. On *Ship, Small Vehicle(SV), Large Vehicle(LV), Basketball Court(BC), Storage Tank(ST), Swimming Pool(SP)*, our method achieves the best performance. For categories with large aspect ratios, A^2S-Det has a distinct advantage compared with other detectors where the self-adaptive anchor selection method improves the anchor selection process. The visualizations of predictions are shown in Figure 6.

Our method aims to improve the performance for objects with large aspect ratios but does not achieve the best performance on *Bridge* and *Harbor*. IENet [31] achieves the best performance on *Bridge* and R3Det [3] achieves the best performance on *Harbor*. The mAP of A^2S-Det is very close to the mAP of R3Det [3] on *Harbor*, which respectively are 65.29% and 65.44%. For *Bridge*, the mAP of A^2S-Det is lower than several state-of-art detectors(IENet [31], O^2-DNet [20], and R3Det [3]). From another perspective, there are 3.01% increase compared with the baseline(RetinaNet-R [12]) under no data augmentation on *Bridge*, as shown in Figure 2.

As shown in Figure 2, A^2S-Det has worse performance than the baseline on several categories, such as Baseball Diamond(BD), Basketball Court(BC), and Helicopter (HC). As opposed to objects with large aspect ratios, the aspect ratios of these objects are close to 1. Under these circumstances, A^2S-Det may define many anchors with high IoU into negative anchors, which are not conducive to the training process, but there are few impacts for *Plane* and *Storage Tank*, whose aspect ratios are also close to 1. Taken as a whole, there is a large number of objects for *Plane* and *Storage Tank*, which can make up the deficiency of the anchor selection process.

Table 1. Comparison of our method with other detectors performance ($AP^{IoU=0.5}$) on DOTA.

	Methods	Backbone	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP
two-stage detectors	FR-O [6]	ResNet-101	79.09	69.12	17.17	63.49	34.20	37.16	36.20	89.19	69.60	58.96	49.4	52.52	46.69	44.80	46.30	52.93
	R-DFPN [32]	ResNet-101	80.92	65.82	33.77	58.94	55.77	50.94	54.78	90.33	66.34	68.66	48.73	51.76	55.10	51.32	35.88	57.94
	R^2CNN [2]	VGG16	80.94	65.67	35.34	67.44	59.92	50.91	55.81	90.67	66.92	72.39	55.06	52.23	55.14	53.35	48.22	60.67
	RRPN [1]	VGG16	88.52	71.20	31.66	59.30	51.85	56.19	57.25	90.81	72.84	67.38	56.69	52.84	53.08	51.94	53.58	61.01
	RoI-Transformer [18]	ResNet-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
one-stage detectors	SSD [29]	SSD	39.83	9.09	0.64	13.18	0.26	0.39	1.11	16.24	27.57	9.23	27.16	9.09	3.03	1.05	1.01	10.59
	YOLOv2 [30]	DarkNet-19	39.57	20.29	36.58	23.42	8.85	2.09	4.82	44.34	38.35	34.65	16.02	37.62	47.23	25.5	7.45	21.39
	IENet [31]	ResNet-101	57.14	80.20	64.54	39.82	32.07	49.71	65.01	52.58	81.45	44.66	78.51	46.54	56.73	64.40	64.24	57.14
	Axis-Learning [19]	ResNet-101	79.53	77.15	38.59	61.15	67.53	70.49	76.30	89.66	79.07	83.53	47.27	61.01	56.28	66.06	36.05	65.98
	RetinaNet-R [12]	ResNet-50	88.03	71.04	40.27	50.70	71.33	72.99	84.83	90.76	77.12	82.95	38.38	58.91	55.08	67.41	54.29	66.94
	O^2-DNet [20]	104-Hourglass	89.31	82.14	47.33	61.21	71.32	74.03	78.62	90.76	82.23	81.36	60.93	60.17	58.21	66.98	61.03	71.04
	R3Det [3]	ResNet-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
proposed	A^2S-Det	ResNet-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	ResNet-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

Figure 6. The visualizations of predictions using A^2S-Det on the DOTA dataset.

4. Discussion

4.1. Effectiveness of Self-Adaptive Anchor Selection

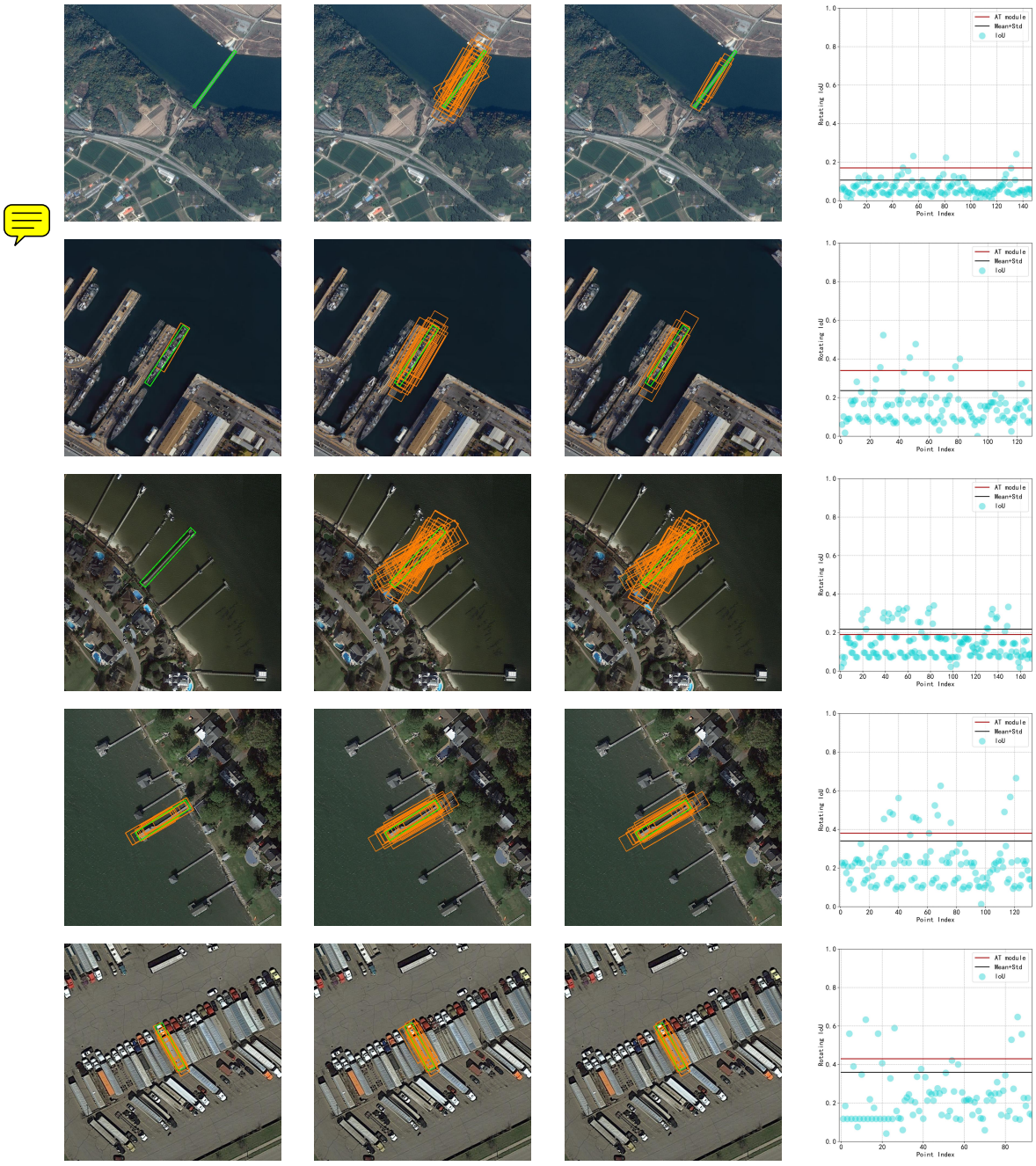
In Section 1, we discuss that there are some weaknesses in rotating detection when using the traditional anchor selection method. The self-adaptive anchor selection method proposed in the Section 2.3.2 is effective, especially for objects with large aspect ratios. As shown in Table 2, A^2S-Det with AT module has 1.12% improvements on the evaluation method of $mAP^{IoU=0.5}$, compared with the baseline *RetianNet-R* [12]. For objects with large aspect ratios, there are 2.46% increase for the *bridge*, 3.06% increase for the SV, 0.25% increase for the LV and 4.73% increase for the *harbor*. Moreover, the AT module plays an important role in A^2S-Det. In this paper, A^2S-Det without AT module adopts (Mean+Std) to define positive anchors and negative anchors. From the Table 2, A^2S-Det with AT module has 0.31% improvements, compared with A^2S-Det without AT module. For those objects with large aspect ratios, there are 1.89% increase for the *bridge*, 0.3% increase for the LV and 3.95% increase for the *harbor*. The A^2S-Det and AT module are very effective in terms of both objects with large aspect ratios and the whole result.

Due to the randomness in anchor generation, defining anchors by a fixed restriction of rotating IoU may lead to few positive anchor in the training process. In Figure 7, we compare the differences of anchor visualization of these three anchor selection methods. The origin anchor selection method in (a) defines positive samples by a fixed IoU threshold. For objects with large aspect ratios, this anchor selection method may cause matching no anchor, such as bridge and harbor. Compared with the origin anchor selection, A^2S-Det which is a flexible anchor selection method performs better in the anchor selection process, especially for objects with large aspect ratios. In A^2S-Det, a set of candidate anchors selected by horizontal IoU can avoid matching no anchor in some special situation. A^2S-Det without AT module use the function of $(mean + std)$ as the threshold to distinguish samples, which is an empirical value. The AT module can adaptively find a boundary between the set of positive samples and the set of negative samples. As shown in Figure 7, selected positive anchors in (c) seem more regular than positive anchors in (b), which is more obvious in images with bridge and harbor. From the distribution of IoU in (d), the AT module can divide the candidate anchors into positive anchors and negatives anchors by the characteristics of IoU distribution, instead of an empirical value.

Table 2. Ablation study ($AP^{IoU=0.5}$) of each module in our proposed method on DOTA.

Methods	DataAug	AT	CR^3	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP
RetinaNet-R [12]	-	-	-	88.03	71.04	40.27	50.70	71.33	72.99	84.83	90.76	77.12	82.95	38.38	58.91	55.08	67.41	54.29	66.94
A^2S-Det	-	-	-	89.05	69.67	40.84	56.88	75.95	72.94	84.13	90.66	73.39	82.15	41.90	62.12	55.86	70.17	50.60	67.75
	-	✓	-	89.09	68.68	42.73	56.94	74.39	73.24	84.68	90.74	73.80	82.95	44.03	59.63	59.81	69.64	50.50	68.06
	-	✓	✓	89.08	70.53	43.28	55.89	74.20	73.54	84.92	90.48	74.42	84.05	42.70	62.63	60.31	68.38	52.36	68.45
	✓	✓	✓	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

✓means this method or module is adpoted.



(**a**) Origin Anchor Selection (**b**) A^2S-Det without AT (**c**) A^2S-Det with AT (**d**) The distribution of IoU

Figure 7. Images of the bridge, harbor, large vehicle and ship from train dataset with their ground truth and positive anchors. (**a**) visualizations of positive anchors when using origin anchor selection; (**b**) visualizations of positive anchors in A^2S-Det without AT; (**c**) visualizations of positive anchors in A^2S-Det with AT; and (**d**) the anchor selection process using two threshold defining methods of $(Mean + Std)$ and AT.

4.2. Effectiveness of CR^3 Module

CR^3 module proposed in Section 2.3.4 solves the problem of inaccurate regression. As shown in Table 2, CR^3 module has a positive influence. In the benchmark of A^2S-Det without CR^3 module, there is a total 0.39% increase if CR^3 module is applied to A^2S-Det. For those objects with large aspect ratios, there is an obvious increase, especially for *Bridge*,

LV, *Ship*, and *Harbor*. What is more, there are 0.55% increase for *bridge*, 0.3% increase for *LV*, and 0.5% increase for *harbor*. Just from the view of *Average Precision*(AP), CR^3 module may not have a big advantage while the increase of AP is not obvious. For evaluating if a target is detected correctly, the rotating IoU threshold is set to 0.5 in Evaluation-Server [4]. The object is considered to be detected correctly if the rotating IoU between the predicted bounding box and the ground-truth box is greater than the threshold. As shown in Figure 8, the bounding box of A^2S-Det with CR^3 module regress better than the bounding box of A^2S-Det without CR^3 module. Due to the rotating IoU threshold of 0.5, most predicted boxes are judged as correct while there is a little deviation in the regression of bounding box, such as results in Figure 8. Whether regressing precisely has a greater influence on objects with larger aspect ratios(*bridge*, *harbor*, *LV*).

(**a**) Results without CR^3 module (**b**) Results with CR^3 module

Figure 8. The visualizations of whether the CR^3 module is adopted. (**a**) visualizations of predictions without the CR^3 module; (**b**) visualizations of predictions with the CR^3 module.

The official evaluation server only support $AP^{IoU=0.5}$. To verify the reasoning, we train the model on training dataset and test on validation dataset, whose performance evaluation method uses $AP^{IoU=0.75}$ and AP^*. AP^* means we test the model from $AP^{IoU=0.5}$ to $AP^{IoU=0.95}$, where the IoU step is 0.05, and calculate the average value as AP^*. As shown in Table 3, A^2S-Det without CR^3 performs better than A^2S-Det with CR^3 for the *bridge* and the *harbor* on $AP^{IoU=0.5}$, but on $AP^{IoU=0.75}$, there are 1.52% increase for the *bridge* and 1.01% increase for the *harbor* on A^2S-Det with CR^3. On the stricter evaluation method(like $AP^{IoU=0.75}$), the impact of CR^3 module is more obvious, such as the *LV*, the *BC* and the *SBF*.

Table 3. Comparison of CR^3 module effect with different evaluation methods ($AP^{IoU=0.5}$, $AP^{IoU=0.75}$, and AP^*) on DOTA.

Methods	Evaluation	CR^3	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP
A^2S-Det	$AP^{IoU=0.5}$	-	89.01	58.36	31.82	44.35	53.50	71.34	86.40	90.37	54.91	85.27	32.97	58.13	51.17	52.14	39.59	59.95
		✓	89.12	59.64	29.59	45.10	58.08	72.52	86.13	90.23	56.67	84.98	34.76	56.08	50.82	51.37	36.52	60.11
A^2S-Det	$AP^{IoU=0.75}$	-	51.23	9.76	3.03	20.93	19.41	31.30	34.12	80.53	30.73	43.24	5.57	22.73	6.94	6.36	9.70	25.04
		✓	54.18	8.65	4.55	23.49	19.21	34.85	34.82	80.36	37.20	40.31	9.04	22.55	7.95	4.32	8.77	26.02
A^2S-Det	AP^*	-	50.72	24.47	10.69	21.98	27.20	37.31	42.51	70.42	29.82	46.95	11.28	29.26	17.83	17.73	18.06	30.42
		✓	53.89	25.13	10.32	22.67	26.76	38.55	42.65	70.19	34.54	46.63	13.21	29.73	17.33	16.96	15.22	30.92

✓means this method or module is adpoted.

4.3. Advantages and Limitations

The anchor-based rotating detectors are very dependent on anchor parameters and the positive-negative threshold, which are hard to be adjusted. The anchor parameters affect the anchor generation process, indirectly affecting the anchor selection process, such as aspect ratios, angels, and scales. Moreover, the anchor selection process is directly affected by the positive-negative threshold. As is discussed in Section 1, there are some situations where some objects with large aspect ratios match few anchors for training, leading to inadequate training and bad performance. A^2S-Det combines horizontal features and rotating features and selects anchors entirely based on the distribution of the rotating IoU. Our method solves the problem of the missing matching and low matching ratio between anchors and objects, especially for objects with large aspect ratios. As the visualizations of the anchor selection process in Figure 7 shows, A^2S-Det is suitable for both general situations and extreme situations. For predicting rotating bounding boxes, the CR^3 module helps regress the rotating bounding box precisely. As shown in Table 1, our method has better performance than most rotating detectors, showing great potential on those objects with large aspect ratios. The Table 2 shows there are big improvements when the modules proposed in Sections 2.3.2, 2.3.3 and 2.3.4 are applied in the baseline(RetinaNet-R [12]).

Combining horizontal features with rotating features and searching for an appropriate threshold based on the balance of samples provide a possible direction to improve the anchor matching process. For objects with large ratio aspects, horizontal features benefit the category classification while rotating features benefit the box regression, but it takes up a lot of computation to calculate both the rotating IoU and the horizontal IoU, which may be simplified in future work. In this paper, we explore how to define positive anchors and negative anchors by the distribution of anchors. The anchor matching process is considered as an optimization problem, whose goal is to keep a balance between positive anchors and negative anchors. To reduce the amount of calculation and achieve end-to-end training, we simplify the goal function and the solving process. As shown in Tables 1 and 2 , considering the anchor matching process as an optimization problem shows great potential.

There are also some limitations of this method. Firstly, the angle deviation has a little impact for objects whose aspect ratios are close to 1, such as $Plane$, BD, ST, BC, RA, and HC. Therefore, the self-adaptive anchor selection method based on the distribution of the rotating IoU is not suitable for all categories. In Table 2, there are 0.51% decreases for BD, 2.7% decreases for BC and 1.93% decreases for HC on $AP^{IoU=0.5}$ when A^2S-Det is compared with the baseline. Benefiting from a large number of objects, the AP of A^2S-Det does not decline on $Plane$ and ST. This limitation may lead to low AP for the object whose height is close to width if there is a small number of objects. Further, this method increases the training time. Both the rotating IoU and the horizontal IoU are needed in A^2S-Det, and the threshold solving process in the AT module costs a lot of time. The training time of A^2S-Det nearly doubles the baseline. The inference process of A^2S-Det is almost unaffected, whose inference time is very close to the baseline. There are some inadequacies of the proposed method on mAP compared with some state-of-art rotating detectors. The IoU can not describe the relationship between positive anchors and objects well. We will aim to adopt a better way to describe it and improve this method in the future.

5. Conclusions

We have presented a self-adaptive anchor selection method based on the one-stage detector. Aiming at objects with large aspect ratios, three modules are proposed in this paper, which are the self-adaptive anchor selection module, the AT module, and the CR^3 module. A^2S-Det improves the prediction performance of objects with large aspect ratios by improving the anchor selection process. The CR^3 helps regress the rotating bounding box more precisely. Further, we design several experiments on DOTA [4] dataset and prove that these modules are effective in aerial image object detection. Our approach has better performance than many other state-of-art rotating detectors according to $mAP^{IoU=0.5}$, achieving the mAP of 70.64. Compared with the baseline detector, there is an averagely

1.51% increase when these three modules are applied. For objects with large ratio aspects, the increases of mAP range from 0.09% to 5.23%. The results prove that an efficient anchor matching method helps the detector learn better feature information and achieve better performance for objects with large ratio aspects. In future work, we will aim to improve this method and explore more potential methods of label assignment to improve the detection performance in aerial images.

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

Funding: This research received no external funding.

Data Availability Statement: The source code used in this study is available in [https://github.com/RSIA-LIESMARS-WHU/A2S-DET].

Acknowledgments: The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

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

Abbreviations

The following abbreviations are used in this manuscript:

A^2S-Det	self-adaptive anchor selection
CR^3	coordinate regression of relative reference
AT	self-adaptive threshold
IoU	intersection over unit
NMS	non-maximum suppression
BD	baseball diamond
GTF	ground track field
SV	small vehicle
LV	large vehicle
TC	tennis court
BC	basketball court
ST	storage tank
SBF	soccer ball field
RA	roundabout
SP	swimming pool
HC	helicopter
Std	standard deviation
AP	average precision

References

1. Ma, J.; Shao, W.; Ye, H.; Wang, L.; Wang, H. Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE Trans. Multimed.* **2017**, *20*, 3111–3122.
2. Jiang, Y.; Zhu, X.; Wang, X.; Yang, S.; Li, W.; Wang, H.; Fu, P.; Luo, Z. R2cnn: rotational region cnn for orientation robust scene text detection. *arXiv* **2017**, arXiv:1706.09579.
3. Yang, X.; Liu, Q.; Yan, J.; Li, A.; Zhang, Z.; Yu, G. R3det: Refined single-stage detector with feature refinement for rotating object. *arXiv* **2019**, arXiv:1908.05612.
4. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3974–3983.
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation; In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.

6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:cs.CV/1506.01497.
7. Kaiming, H.; Georgia, G.; Piotr, D.; Ross, G. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
8. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944
9. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 761–769.
10. Gong, Y.; Xiao, Z.; Tan, X.; Sui, H.; Xu, C.; Duan, H.; Li, D. Context-Aware Convolutional Neural Network for Object Detection in VHR Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 34–44. [CrossRef]
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
12. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2999–3007
13. 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), Seoul, Korea, 27 October–2 November 2019; pp. 9626–9635.
14. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
15. 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 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2642–2651.
16. Liu, L.; Pan, Z.; Lei, B. Learning a rotation invariant detector with rotatable bounding box. *arXiv* **2017**, arXiv:1711.09405.
17. Liao, M.; Shi, B.; Bai, X.; Wang, X.; Liu, W. Textboxes: A fast text detector with a single deep neural network. *arXiv* **2016**, arXiv:1611.06779
18. Ding, J.; Xue, N.; Long, Y.; Xia, G.S.; Lu, Q. Learning roi transformer for detecting oriented objects in aerial images. *arXiv* **2018**, arXiv:1812.00155.
19. 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]
20. Wei, H.; Zhang, Y.; Chang, Z.; Li, H.; Wang, H.; Sun, X. Oriented Objects as pairs of Middle Lines. *arXiv* **2020**, arXiv:cs.CV/1912.10694.
21. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Xian, S.; Fu, K. SCRDet: Towards More Robust Detection for Small, Cluttered and Rotated Objects. *arXiv* **2018**, arXiv:cs.CV/1811.07126.
22. Xu, Y.; Fu, M.; Wang, Q.; Wang, Y.; Bai, X. Gliding vertex on the horizontal bounding box for multi-oriented object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [CrossRef] [PubMed]
23. Wang, J.; Yang, W.; Li, H.; Zhang, H.; Xia, G. Learning Center Probability Map for Detecting Objects in Aerial Images. *IEEE Trans. Geosci. Remote Sens.* **2020**, 1–17. [CrossRef]
24. 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* **2019**, arXiv:cs.CV/1912.02424.
25. Zhang, X.; Wan, F.; Liu, C.; Ji, R.; Ye, Q. FreeAnchor: Learning to Match Anchors for Visual Object Detection. *arXiv* **2019**, arXiv:cs.CV/1909.02466.
26. Ke, W.; Zhang, T.; Huang, Z.; Ye, Q.; Liu, J.; Huang, D. Multiple Anchor Learning for Visual Object Detection. *arXiv* **2019**, arXiv:cs.CV/1912.02252.
27. Cao, Y.; Chen, K.; Loy, C.C.; Lin, D. Prime Sample Attention in Object Detection. *arXiv* **2019**, arXiv:cs.CV/1904.04821.
28. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8026–8037.
29. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
30. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *arXiv* **2016**, arXiv:1612.08242
31. Lin, Y.; Feng, P.; Guan, J. IENet: Interacting Embranchment One Stage Anchor Free Detector for Orientation Aerial Object Detection. *arXiv* **2019**, arXiv:cs.CV/1912.00969.
32. 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]