Abstract—Arbitrarily oriented object detection in remote sensing images is a challenging task. At present, most of the algorithms are dedicated to improving the detection accuracy, while ignoring the detection speed. In order to further improve the detection accuracy and provide a more efficient model for scenes that require real-time detection, we propose an improved YOLOv4-CSP network for rotating object detection in remote sensing images. There are mainly three contributions in our approach. First, we design a new bounding box regression loss function, which is distance and angle-intersection over union. This loss function is formed by adding a distance penalty term and an angle penalty term on the basis of intersection over union. It is suitable for arbitrarily oriented object detection networks. Second, we develop an adaptive angle setting method for anchors based on the k-means clustering algorithm. This method can obtain representative angles for better representing the distribution of the angle set. By assigning representative angles to all anchors for training, it is beneficial to reduce the complexity of the network to adjust anchors to ground-truth bounding boxes. Finally, we improve the YOLOv4-CSP network and make it suitable for detection scenarios based on rotated anchors by applying rotation transformations. We combine the aforementioned methods and use the final network to perform the detection task. The experimental results on three remote sensing datasets, i.e., HRSC2016, UCAS-AOD, and SSDD+, validate the effectiveness of our method. Comparison results with state-of-the-arts methods demonstrate that our method can be used to significantly improve the detection accuracy with a higher detection speed.

Index Terms—Arbitrarily oriented object detection, distance and angle-intersection over union (DAIoU), k-means, YOLOv4-CSP.

I. INTRODUCTION

With the prevalence of deep learning in the field of computer vision, many excellent object detectors have emerged. These detectors do not require the manually designed features such as those used in traditional object detectors. They use deep neural networks to automatically extract features, and these detectors significantly improve the efficiency and accuracy for object detection. There are two main categories of object detectors based on deep learning: two-stage object detectors [1], [2], [3] and single-stage object detectors [4], [5], [6], [7], [8], [9], [10], [11]. Generally, two-stage detectors have excellent detection accuracy and single-stage detectors have excellent detection speed. The representative detectors in the category of single-stage object detectors are SSD [4] and YOLO [7]. Up to now, these object detectors have been widely used in many object detection tasks.

However, the detectors mentioned previously are all based on anchors with horizontal and vertical borders or upright anchors. For object detection in remote sensing images, the scenes in remote sensing images are very complex and a large number of objects appear in different orientations as shown in Fig. 1. Object detectors based on horizontal and vertical borders have many limitations. They cannot depict the orientations and the sizes of objects in detail. In order to deal with these issues, many detectors based on rotated anchors have been developed in recent years [12], [13], [14], [15], [16]. Arbitrarily oriented object detectors have the following advantages. First, the rotated bounding boxes can separate objects from background as much as possible. Usually, many pixels in an upright bounding boxes do not belong to the objects, which is not helpful for distinguishing the target area from background, especially in a scene where the objects are densely located. Second, the upright bounding boxes often cannot depict the actual aspect ratios and sizes of objects, whereas a rotated bounding boxes can.
Third, rotated bounding boxes can be beneficial for the estimation of the orientations of objects while performing the detection task.

Currently, in the task of arbitrarily oriented object detection, the commonly used bounding box regression loss function is the smooth L1 loss function. The R3Det [15], CSL [16], and DCL [17] methods also use this approach. The use of these loss functions has achieved satisfactory results, but the loss functions based on the bounding box IoU are more suitable for the task of arbitrarily oriented object detection. They have the following advantages. First, when calculating the IoU, all shape information has been taken into account and the relationship between different parameters has been implicitly encoded. Second, IoU is scale invariant, which is suitable to resolve the scale difference and range difference between the parameters. Finally, these IoU-based loss functions do not have a boundary problem.

In previous work, the rotated bounding box IoU was not directly used as a loss function for the task of arbitrarily oriented object detection. This is because the derivative computation of the rotated bounding box IoU is not easy to implement. Recently, the authors in [27] have realized this computation.

Based on the aforementioned reasons, we design a bounding box regression loss function that can help arbitrarily oriented object detection networks converge better and faster. In the training process, the distance and the angle difference between ground-truth (GT) bounding boxes and anchors are two important factors. They will affect the value of IoU. Therefore, we add a distance penalty term and an angle penalty term on the base IoU to construct a new bounding box regression loss function, distance and angle-intersection over union (DAIoU), to obtain a better performance.

In arbitrarily oriented object detection tasks, every bounding box has an orientation. Setting suitable orientations can reduce the difficulty on network training and improve the performance of the model. For the setting on the orientations of anchors, the commonly used methods divide the whole orientation range such as [−90°, 0°] or [−180°, 0°] with a fixed interval value. There are also some recent models [13, 15], which use upright anchors for initial training. Although these methods could achieve satisfactory results, they do not make effective use of the characteristics of the datasets. Fig. 2 shows the distribution of object orientations in different datasets. From Fig. 2, we can see that for different datasets, the angular distribution varies with different samples. Therefore, it is reasonable to develop an adaptive orientation partition method for anchors based on different dataset samples. Thus, to obtain a representative partition that can better illustrate the distribution of anchor orientation for training. In this article, we introduce a k-means clustering algorithm for anchor orientation partitioning, and it can greatly reduce the difficulty of adjusting anchors to ground truth (GT) bounding boxes, thereby reducing the difficulty of training. Finally, considering the requirements on accuracy and speed in the task of arbitrarily oriented object detection, we developed our models based on the YOLOv4-CSP network and improved implementation issues such as data augmentation and bounding box regressions. By combining the improved network with the DAIoU loss function, we proposed the improved YOLOv4-CSP for arbitrarily oriented object detection.

There are mainly three contributions of our work.

1) In order to further improve the detection accuracy on arbitrarily oriented objects in remote sensing images, we propose a new bounding box regression loss function (DAIoU), which is suitable for arbitrarily oriented object detection networks.

2) For the setting on anchor orientations, we use the k-means clustering algorithm to cluster the orientations of bounding boxes in the training set and obtain several orientations that can better represent the distribution of anchor orientations. It is beneficial to reduce the difficulty of adjusting anchors to GT bounding boxes and further improve the detection accuracy.

3) We developed an improved architecture for arbitrarily orientated object detection by combining the YOLOv4-CSP network with the DAIoU loss function. By taking full advantages of the YOLOv4-CSP network and the DAIoU loss function, our proposed model can significantly improve the accuracy and efficiency of the arbitrarily oriented object detection for remote sensing images.

II. RELATED WORKS

In this section, we discuss the related work on object detectors with upright anchors and arbitrarily oriented object detectors.

A. Object Detectors With Upright Anchors

Classical object detectors aim to use upright anchors for object detection. Many high-performance object detectors have been proposed. RCNN [1] is a classical algorithm used in two-stage upright object detectors. Fast RCNN [2] and Faster RCNN [3] optimize the speed and accuracy on the basis of RCNN. The main contribution of Faster RCNN is the proposal of the region proposal network (RPN), which significantly improves the efficiency of candidate regions acquisitions. SSD [4] and
YOLO [7] are representative detectors based on upright anchors in single-stage object detectors. The whole process does not need to be split, hence, the detection speed can be considerably improved. All the detectors mentioned earlier are anchor based. In recent years, many anchor-free object detectors have also been proposed. They remove the operation of setting anchors and obtain the object location by directly predicting some key points. The representative detectors include CornerNet [28], CenterNet [29], and ExtremeNet [30]. CornerNet obtains the bounding box by detecting two key points at the upper left corner and the lower right corner. CenterNet directly predicts the center point, width, and height of each object to obtain the bounding box. ExtremeNet obtains the bounding boxes by predicting four extreme points (top, left, bottom, and right) and a center point of each object. Although the detectors have been applied in many fields, it is no longer applicable for the task of object detection in remote sensing images. The orientations of objects in remote sensing images are arbitrary. The detectors based on upright anchors cannot provide accurate orientation and scale information of objects.

B. Arbitrary-Oriented Object Detectors

In recent years, many arbitrarily oriented object detectors have been developed. There are mainly two categories for these detectors: anchor-based detectors and anchor-free detectors. For anchor-based detectors, multistage detectors are mainly used, such as those in region-of-interest (RoI) transformer [18] and SCRDet [14]. In the work of the ROI transformer, a rotated RoI (RRoI) learner and a rotated position sensitive RoI alignment (RPS-RoI-Align) module are designed to improve the performance of classification and regression. In the work of SCRDet, a new network is designed for small objects detection by fusing multilayer features and sampling effective anchor points, which improves the performance of classification and regression. The methods based on multilayer features are able to detect small objects with high performance. The new network is designed for small objects detection by fusing multilayer features and sampling effective anchor points, which is beneficial for the detection of small objects.

There are also some excellent single-stage detectors. R²Det [15] is a refined single-stage detector based on rotated anchors. The design takes into consideration the advantage of the high recall rate of upright anchors and the adaptability of rotated anchors to dense scenes. CSL [16] is also an excellent single-stage detector. In order to avoid the problem of discontinuous boundaries caused by angular periodicity, it transforms the orientation prediction task from a regression problem to a classification one. CSL is designed to deal with the periodicity of orientations and increase error tolerance on adjacent orientations. This work avoids the boundary problem in the regression process on orientations. P-RSDet [25] and BAVectors [26] are the latest anchor-free detectors for arbitrarily oriented object detection tasks. P-RSDet introduces the polar coordinate system to the deep learning detector for the first time. It achieves competitive detection accuracy by using a simpler object representation model and fewer regression parameters. BAVectors extends the object detector based on upright anchors to arbitrarily oriented object detection tasks. During training, it first detects the center points of objects. Then, it regresses the perception vectors of the bounding boxes based on the center points. Finally, the rotated bounding boxes are obtained.

The methods mentioned previously mostly use the smooth L1 loss function or the IoU-Smooth L1 loss function to calculate the loss of the bounding boxes. However, they have boundary problems in the regression process for the bounding boxes and they calculate the loss of each parameter separately, which ignores the correlation between various parameters. In fact, the parameters of the bounding boxes are related. On the contrary, the loss functions constructed based on the IoU between rotated boxes do not have these problems. So, we design a new bounding box loss function based on the bounding box IoU. We combine it with the YOLOv4-CSP network for the task of arbitrarily oriented object detection in remote sensing images.

III. METHODOLOGY

Our work focuses on the design of a new bounding box regression loss function and the design of rotated anchors. In this section, we will introduce our method in details.

A. Definition of the DAIoU Loss Function

In the task of object detection based on rotated anchors, the distance and the orientation differences between the anchor and the GT bounding box are two important factors. The distance represents how far the center point of the anchor is from the center point of the GT bounding box, and the angle difference represents the degree of consistency between the orientation of the anchor and the orientation of the GT bounding box. Therefore, we add a penalty term based on distance and a penalty term based on orientation on the basis of IoU to construct a new loss function DAIoU. The distance penalty term is based on the normalized distance, which is obtained by normalizing the distance between the anchor and the GT bounding box with the diameter of their circumcircle. The orientation penalty term is expressed by the cosine of the angle between the shortest side of the anchor and the shortest side of the GT bounding box.

Fig. 3 is a schematic diagram of the circumcircle of two rectangles. The coordinate of the center point of the anchor is $C_1(X_1, Y_1)$. The coordinate of the center point of the GT
DAIoU for Two Rotated Bounding Boxes.

Input: Coordinates of predicted bounding box \( B_p \) and GT bounding box \( B_g \): \( B_p = (x_p, y_p, w_p, h_p, \theta_p) \), \( B_g = (x_g, y_g, w_g, h_g, \theta_g) \)

Output: DAIoU

1: Compute the area of \( B_p \): \( \text{Area}_p = w_p \times h_p \)
2: Compute the area of \( B_g \): \( \text{Area}_g = w_g \times h_g \)
3: Determine the vertices of the overlapping area of \( B_p \) and \( B_g \) by calculating line intersections and vector projections
4: Sort these polygon vertices in an anticlockwise order
5: Based on the sorted vertices, calculate the intersection area \( \text{Area}_{overlap} \) of \( B_p \) and \( B_g \) by using the Gaussian area formula
6: Calculate the IoU of \( B_p \) and \( B_g \): \( \text{IoU} = \frac{\text{Area}_{overlap}}{\text{Area}_p + \text{Area}_g - \text{Area}_{overlap}} \)
7: Calculate the distance \( L_{pg} \) between \( B_p \) and \( B_g \): \( L_{pg} = \sqrt{(x_p - x_g)^2 + (y_p - y_g)^2} \)
8: Obtain the vertices that can determine the minimum circumscribed circle \( P_{sc} \) of \( B_p \) and \( B_g \) by the minimum circumscribed circle algorithm
9: Calculate the diameter \( D \) of the circle \( P_{sc} \) by its vertices
10: Calculate the cosine coefficient \( \cos \alpha \) of the angle between the shortest side of \( B_p \) and the shortest side of \( B_g \)
11: Calculate the DAIoU of \( B_p \) and \( B_g \): \( a \) and \( b \) are the weights of the corresponding items: \( \text{DAIoU} = \text{IoU} - (a \times (L_{pg}^2 / D^2) + b \times (1 - \cos \alpha)) \)

Algorithm 1: DAIoU for Two Rotated Bounding Boxes.

In the back propagation process, the distance penalty term and the angle penalty term work together. Regardless of whether there is any overlap between the anchor and the GT bounding box, DAIoU always has a gradient.

The formula on DAIoU as regression loss function is shown as follows:

\[
L_{\text{DAIoU}} = 1 - \text{DAIoU} = 1 - \left( \text{IoU} - (a \times (L_{pg}^2 / D^2) + b \times (1 - \cos \alpha)) \right)
\]

We use the following five parameters to represent the rotated bounding box:

\[
(x, y, w, h, \theta)
\]

where \((x, y)\) denotes the center coordinate of the rotated bounding box and \((w, h, \theta)\) denote the width, height, and angle of the rotated bounding box, respectively. The side with an acute angle to the positive direction of the \( x \)-axis is denoted by \( w \) and the other side is denoted by \( h \). The angle between the side indicated by \( w \) and the \( x \)-axis is denoted by \( \theta \) (including the right angle), with \( \theta \in [-90^\circ, 0^\circ] \). The specific representation of the parameters on a rotated bounding box is shown in Fig. 4.

Because \( \theta \) is a new parameter, the angle regression formula needs to be included to the original bounding box regression formulas of the YOLOv4-CSP network. Equations (5)–(8) are for the bounding box regression of YOLOv4-CSP, and (9) is for the angle regression

\[
b_x = 2\sigma(t_x) - 0.5 + c_x
\]

\[
b_y = 2\sigma(t_y) - 0.5 + c_y
\]

\[
b_w = p_w(2\sigma(t_w))^2
\]

\[
b_h = p_h(2\sigma(t_h))^2
\]

\[
b_{\theta} = t_{\theta} \times 180/\pi + p_{\theta}
\]
where $t_x$, $t_y$, and $t_\theta$ denote the offset of the coordinates of the bounding box and angle. $t_w$ and $t_h$ denote the corresponding scaling ratios on width and height of the bounding box. $b_x$, $b_y$, $b_w$, $b_h$, and $b_\theta$ denote the center coordinates of the bounding box, width, height, and angle, respectively. $c_x$ and $c_y$ denote the coordinates of the upper left corner of the grid that responsible for the bounding box. $p_w$, $p_h$, and $p_\theta$ denote the width, height, and angle of the bounding box relative to the feature map, respectively.

Finally, we perform a weighted summation of the proposed bounding box loss ($L_{DAIoU}$), the confidence loss ($L_{obj}$), and the classification loss ($L_{cls}$) to obtain the final multiloss task, which is defined as follows:

$$L_{total} = \lambda_{box} \times L_{DAIoU} + \lambda_{obj} \times L_{obj} + \lambda_{cls} \times L_{cls}$$

(10)

where $\lambda_{box}$, $\lambda_{obj}$, and $\lambda_{cls}$ denote the weight of each loss. BCELoss is used to calculate the confidence loss $L_{obj}$ and the classification loss $L_{cls}$. In this article, except for the weights of the penalty terms in the DAIoU loss function, we use the default weight of the YOLOv4-CSP network for the weights for the rest of the functions.

B. Definition of the Angles of Anchors

For the setting of anchors’ angles, the commonly used method is to divide the angle interval equally, and then, assign the obtained angles to all anchors. For example, the angle interval $[-90^\circ, 0^\circ]$ is divided equally at $15^\circ$ intervals. Then, six angles of $-90^\circ$, $-75^\circ$, $-60^\circ$, $-45^\circ$, $-30^\circ$, and $-15^\circ$ are obtained. After that, assign these six angles to all anchors for network training. In any case, the aforementioned methods lay the same anchors on the feature map.

However, we count the angles of the bounding boxes in the training sets of the three datasets HRSC2016, UCAS-AOD, and SSDD+. Fig. 2 shows the distribution of all angles in these three datasets. By observing the angle distributions in the three datasets, it can be seen that the angle distributions of different datasets are quite different, and the angle distributions in different angle intervals are also very different. At this time, if we continue to use the aforementioned conventional method to set anchors’ angles, the characteristics of the dataset cannot be effectively utilized. However, if a clustering algorithm is used to cluster the angle sets of each dataset, the angles that are more in line with the distribution of the angle sets will be counted. Then, assigning the obtained angles to all anchors will get the anchors that are closer to the GT bounding boxes, which can reduce the difficulty of fine tuning the anchors to the GT bounding boxes during the training process.

Therefore, we choose to use the k-means algorithm to cluster the angle sets in the training sets of HRSC2016, UCAS-AOD, and SSDD+. Then, we obtain the angles that are more in line with the distribution of different angle sets. Finally, we assign these angles to all anchors for network training. The number of clusters $k$ is the number of angles for anchors. In addition to using the same number of angles as the usual method, we also use two commonly used methods, the elbow method and the silhouette coefficient method, to obtain the number of angles as a comparison.

The core metric of the elbow method is the sum of squared errors (SSE). The formula is as follows:

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

(11)

where $C_i$ is the $i$th cluster; $p$ is a sample in $C_i$; and $m_i$ is the centroid of $C_i$, and it is the mean value of all samples in $C_i$. The more compact the samples in each cluster are, the smaller the SSE is. When choosing the optimal $k$, the elbow method calculates the SSE when different $k$ is applied. As $k$ increases, the number of samples in each cluster will decrease. Each sample will be closer to the centroid of each cluster and the SSE will be smaller. The rate of the decrease of the SSE will also be slower. With $k$ increasing, when the rate of the decrease of the SSE reaches the maximum, the corresponding $k$ is the optimal value determined by the elbow method.

The core metric of the silhouette coefficient method is the silhouette coefficient, which can be used to evaluate the effect of the clustering result.

$$S = \sum_{i=1}^{n} s_i / n$$

(12)

$$s_i = (b_i - a_i) / \max(a_i, b_i)$$

(13)

$$b_i = \min(b_{i1}, b_{i2}, \ldots, b_{ik})$$

(14)

where $n$ is the number of samples; $s_i$ is the silhouette coefficient of sample $i$; $a_i$ is the average distance from the sample $i$ to other samples in the same cluster, and it represents the intracluster dissimilarity of sample $i$; $b_i$ is the minimum of the average distance from the sample $i$ to all samples in other clusters; $b_{ik}$ is the average distance from the sample $i$ to all samples in the cluster $k$, and it represents the cluster dissimilarity of the sample $i$; and $S$ is the average of the silhouette coefficient of all samples, and it is a measure of whether the clustering result is reasonable and effective. The silhouette coefficient is in the range of $[-1, 1]$. The larger the silhouette coefficient, the more reasonable it is.

IV. EXPERIMENTS

In this section, we present the performance of the proposed method on three remote sensing datasets containing vehicle objects such as ships, cars, and planes. A brief overview of these three datasets will be given as follows.

A. Datasets

1) HRSC2016 [31]: HRSC2016 is a remote sensing dataset on ships. It contains 1070 images and 2976 instances. The image sizes range from 300 × 300 to 1500 × 900. The training set, validation set, and test set include 436 images, 181 images, and 444 images, respectively.

2) UCAS-AOD [32]: UCAS-AOD is a remote sensing dataset. It contains 1510 images and two categories of targets: cars and
planes. The number of car images and plane images are 1000 and 510 respectively. Similar to the approaches used in [15] and [23], we randomly select 1110 images for training and 400 images for testing.

3) SSDD+ [33]: SSDD+ is another remote sensing dataset on ships. It contains 1160 images with different resolutions, different scales, and different sea conditions. We divide it into training set and test set according to the ratio of 8:2. Unlike optical remote sensing images, the images in SSDD+ are synthetic aperture radar (SAR) images. With SAR imaging, the superposition of various random scattering signals on object surfaces will produce speckle noise. This type of noise will spread to the entire image. It will cause considerable interference to object detectors. Hence, SSDD+ is a challenging remote sensing dataset on ships.

In addition, the size of the images used for training is 416 × 416 unless otherwise stated.

B. Parameters Setting and Implementation Details

For all experiments, we use 150 epochs. The initial learning rate is 0.01, and the decay weight of the learning rate is 0.0005. The batch size is 8. The stochastic gradient descent optimizer is used and the momentum factor is 0.937. The IoU threshold is 0.5. The operating system is Ubuntu 16.04 and the GPU is Tesla K40c.

C. Anchors Setting

Our experiments use the anchors that come with the YOLOv4-CSP network. Because our work is aimed at the rotation target detection task, the rotation angle is added to all the original anchors. We adopt the method of this article to set the anchors’ angles for network training. The specific form of the anchors is: \((w, h, \theta)\), where \(w\) is the side of the acute angle formed by the anchor and the positive direction of the x-axis, \(h\) is the other side, and \(\theta\) is the opposite number of the acute angle formed by the side \(w\) and the positive direction of the x-axis. The range of \(\theta\) is \([-90^\circ, 0^\circ]\).

D. Evaluating Metrics

We use six evaluation metrics in our experiments. They are precision (P), recall (R), average precision (AP), mean average precision (mAP), F1-Score, and detection speed (FPS). The specific expressions of each metric are as follows:

\[
P = \frac{TP}{(TP + FP)} \quad (15)
\]

\[
R = \frac{TP}{(TP + FN)} \quad (16)
\]

\[
AP = \int_0^1 P(R)dR \quad (17)
\]

\[
mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \quad (18)
\]

\[
F1\text{-Score} = \frac{2 \times (P \times R)}{(P + R)} \quad (19)
\]

where TP and FP denote the number of correct samples and wrong samples in all results, respectively. FN denotes the number of undetected samples. \(P\) refers to the proportion of correct samples in all samples detected. \(R\) refers to the proportion of correct samples detected in the total positive samples. AP is the mean value of the accuracy rate on the \(P–R\) curve. mAP is the average value of the AP of all categories. \(N\) is the number of categories. FPS is the number of images processed per second.

In addition, in our experimental results, “07” indicates that the calculation of mAP uses the VOC 2007 evaluation formula, and “12” indicates that the calculation of mAP uses the VOC 2012 evaluation formula. By default, the calculation of mAP uses the VOC 2012 evaluation formula.

E. Angles Analysis

Based on the three remote sensing datasets, HRSC2016, UCAS-AOD, and SSDD+, we analyze how to select more suitable angles to assign to all anchors for network training. First, the angles of the labels in the training set are proposed to form an angle set. Second, the optimal \(k\) is obtained by applying the silhouette coefficient method and the elbow method. Third, the angle set is clustered to obtain the corresponding number of angles. Finally, the obtained angles are assigned to all anchors.

There are four methods for setting the angles.

1) Even division (ED) method. We divide the angle range \([-90^\circ, 0^\circ]\) into six equal parts and obtain six angles, which are \(-90^\circ, -75^\circ, -60^\circ, -45^\circ, -30^\circ, and -15^\circ\).

2) Fixed clusters (FC) method. Corresponding to the number of angles in the previous method, we cluster the angle samples to obtain six angles.

3) Elbow (E) method. We cluster the angle samples according to the optimal \(k\) to obtain the corresponding number of angles. The \(k\) is obtained by using the elbow method.

4) Silhouette coefficient (SC) method. According to the optimal \(k\) obtained by the silhouette coefficient method, we cluster the angle samples to obtain the corresponding number of angles.

Table I shows the mAP and F1-Score obtained by applying the aforementioned four anchor angle setting methods on the three remote sensing datasets of HRSC2016, UCAS-AOD, and SSDD+. The data format in the table is: “mAP/F1–Score (number of angles)” Table II shows the precision and recall obtained by applying the four methods of setting the anchors’ angles on the HRSC2016 dataset. The parameters \(a\) and \(b\) in the DAIoU loss function are both set to 1 by default in the experiments of this section.

By observing the experimental results in Table I, we can see that when the number of angles is the same, the model obtained by using the clustering algorithm to set anchors’ angles for network training has better performance, and the mAP and F1-Score are the best. When the elbow method and silhouette coefficient method are applied, the mAP and F1-Score are further improved on the HRSC2016 and UCAS-AOD datasets, and slightly decreased on the SSDD+ datasets.

When the elbow method and the silhouette coefficient method are used to set anchors’ angles, the performance of the obtained
As can be seen from Table II, changing the number of anchors has different effects on the precision and recall of the model. When the number of anchors decreases, the precision of the model increases, but the recall decreases. When the number of anchors increases, the recall of the model increases, but the precision decreases. The comprehensive performance of a model should not only focus on one of these indicators, so we use the harmonic mean of precision and recall, namely F1-Score, to characterize the comprehensive performance of the model. By observing the results in Table II, we can see that the comprehensive performance of the model is better when the silhouette coefficient method is used to set the anchors’ angles.

Fig. 5 and Fig. 6 show the mAP curves and F1-Score curves, which are obtained by applying the even division method and the silhouette coefficient method based on the HRSC2016 dataset to set the anchors’ angles for network training. By observing the trend of the curves, we can see that the performance of the model obtained by applying the silhouette coefficient method to set the anchors’ angles for network training is better.
is relatively even, the performance of the model by applying the elbow method and the silhouette coefficient method to set anchors’ angles will be further improved. We integrate this clustering operation of selecting anchors into our approach, which can be applied in different scenarios to obtain more suitable anchors for training.

### F. Weights Analysis

In this section, we conduct experiments on parameter sensitivity and analyze the impact of each penalty item with different weights on the performance of the model. Because the penalty items are added on the basis of IoU, IoU is still the most important indicator. Therefore, we analyze the weight of each penalty item in the range of [0,1]. In the display of the experimental results, DAIoU_D (Distance) indicates that the loss function only includes the distance penalty item, and DAIoU_DA (Distance and Angle) indicates that the loss function includes the distance penalty item and the angle penalty item.

First, we add the distance penalty term separately on the basis of IoU, and observe the experimental results when the parameter $a$ takes different values. Table III shows the experimental results obtained on the HRSC2016, UCAS-AOD, and SSDD+ datasets. Fig. 7 shows the $mAP$ curves obtained by applying IoU and DAIoU_D on the HRSC2016 dataset. From the experimental results and the $mAP$ curves, we can see that the application effect of DAIoU_D is better, and the detection accuracy of the model is higher. Moreover, as the network converges later, the detection accuracy of the obtained model is always optimal. By further observing the experimental results in Table III, we can see that on different datasets, with the change of the weight of the distance penalty item, the performance of the model is always maintained at a relatively consistent level. The distance penalty item is not sensitive to the size of the weight. In subsequent experiments, the parameter $a$ is set to 1 by default.

### TABLE III

| Dataset      | $a$  | 0   | 0.05 | 0.1  | 0.5  | 1   |
|--------------|-----|-----|------|------|------|-----|
| HRSC2016     | 96.32 | 97.08 | 97.11 | 97.06 | 97.08 |
| UCAS-AOD     | 97.43 | 97.53 | 97.54 | 97.51 | 97.52 |
| SSDD+        | 95.23 | 96.00 | 96.03 | 96.03 | 96.06 |

The bold entities indicate the best result.

### TABLE IV

| Dataset      | $b$  | 0   | 0.05 | 0.1  | 0.5  | 1   |
|--------------|-----|-----|------|------|------|-----|
| HRSC2016     | 97.08 | 97.34 | 97.39 | 97.46 | 97.43 |
| UCAS-AOD     | 97.52 | 97.54 | 97.62 | 97.20 | 97.02 |
| SSDD+        | 96.06 | 96.25 | 96.16 | 96.01 | 95.63 |

The bold entities indicate the best result.

### TABLE V

| Dataset      | IoU | DAIoU_D | DAIoU_DA |
|--------------|-----|---------|----------|
| HRSC2016     | 96.32 | 97.08 | 97.46    |
| UCAS-AOD     | 97.43 | 97.52 | 97.62    |
| SSDD+        | 95.23 | 96.06 | 96.25    |

The bold entities indicate the best result.

### TABLE VI

| Model                  | mAP (07) | mAP (12) |
|------------------------|----------|----------|
| YOLOv4-CSP+DIoU        | 89.62    | 96.37    |
| YOLOv4-CSP+IoU         | 89.73    | 96.32    |
| YOLOv4-CSP+GlIoU       | 89.76    | 95.45    |
| YOLOv4-CSP+DAIoU_D     | 89.96    | 97.08    |
| YOLOv4-CSP+DAIoU_DA    | 90.29    | 97.46    |

The bold entities indicate the best result.
Then, we continue to add the angle penalty term and observe the impact of the weight changes on the performance of the model. Table IV shows the experimental results obtained on HRSC2016, UCAS-AOD, and SSDD+ datasets. Fig. 8 shows the mAP curves obtained by applying DAIoU_D and DAIoU_DA on HRSC2016 dataset. From the experimental results and the mAP curves, we can see that the application effect of DAIoU_DA is better, and the detection accuracy of the model is higher. Moreover, as the model converges later, the detection accuracy of the obtained model is always optimal. When the angle penalty item takes different weights, the performance of the model is different. For the HRSC2016 dataset, the angle penalty term when the model is optimal has a larger weight. This is because most ships are larger in size and have larger aspect ratios. At this time, the angle difference has a greater influence on the IoU. Assigning a larger weight to the angle penalty term can make the model to focus on the angle difference, thereby improving the model’s performance.

To sum up, the distance penalty term is not sensitive to the weight, and the angle penalty term is relatively sensitive to the weight. This is because with the different remote sensing scenes, the distribution of the objects’ angles is very different, and the change of the weight of the angle penalty term will affect the trend of model training. However, the distance penalty is a generally applicable penalty with consistent performance across different datasets.

Table V shows the experimental results before and after adding two penalty terms. From the experimental results in the table, we can see that the loss function after adding the distance penalty item and the angle penalty item in turn can further improve the performance of the model.

### Table VII

**Performance Evaluation on HRSC2016**

| Method       | Backbone            | Image size | mAP (07) | mAP (12) |
|--------------|---------------------|------------|----------|----------|
| RetinaNet-H  | ResNet101           | 800×800    | 82.89    | 89.27    |
| ROI-Transformer | ResNet101      | 512×800    | 86.20    | —        |
| Gliding Vertex | ResNet101       | —          | 88.20    | —        |
| BBAVectors  | ResNet101           | 608×608    | 88.60    | —        |
| RetinaNet-R  | ResNet101           | 800×800    | 89.18    | 95.21    |
| R²Det [15]   | ResNet101           | 800×800    | 89.26    | 96.01    |
| OPLD [20]    | ResNet50            | 1024×1333  | 88.44    | —        |
| DCL [17]     | ResNet101           | —          | 89.46    | 96.41    |
| CSL [16]     | ResNet101           | —          | 89.62    | 96.10    |
| OSSDet [21]  | ResNet101           | 512×800    | 89.91    | —        |
| DAL [13]     | ResNet101           | 800×800    | 89.77    | —        |
| YOLOv4-CSP_D | Modified CSPDarknet53 | 416×416   | 89.96    | 97.08    |
| YOLOv4-CSP_DA| Modified CSPDarknet53 | 416×416   | 90.29    | 97.46    |
| YOLOv4-CSP_DA| Modified CSPDarknet53 | 608×608   | 89.96    | 97.39    |
| YOLOv4-CSP_DA| Modified CSPDarknet53 | 800×800   | 89.86    | 97.17    |

The bold entities indicate the best result.

### Table VIII

**Performance Evaluation on UCAS-AOD**

| Method       | Plane | Car | mAP (12) |
|--------------|-------|-----|----------|
| S³ARN [22]   | 97.60 | 92.20 | 94.90    |
| RetinaNet-H  | 97.34 | 93.60 | 95.47    |
| ICN [23]     | —     | —    | 95.67    |
| FADet [24]   | 98.69 | 92.72 | 95.71    |
| R³Det [15]   | 98.20 | 94.14 | 96.17    |
| YOLOv4-CSP_D | 98.71 | 96.33 | 97.52    |
| YOLOv4-CSP_DA| 98.69 | 96.55 | 97.62    |

The bold entities indicate the best result.

### G. Comparison Experiments

We extend the two bounding box loss functions of GIoU [34] and DIoU [35] to the task of arbitrarily oriented object detection. When calculating GIoU and DIoU, the bounding boxes take the smallest bounding boxes of anchors and GTs. Table VI shows the experimental results obtained by using GIoU, DIoU, and DAIoU as the bounding box loss function. By observing the experimental results, it can be found that our proposed loss function achieves the best performance with the highest mAP.

In order to further verify the effectiveness of our method, we make comparisons with the state-of-the-art methods. Tables VI–VIII show the experimental results of our method and other methods on the HRSC2016, UCAS-AOD, and SSDD+ datasets.
1) Results on HRSC2016: The HRSC2016 dataset contains a large number of rotated ships with large aspect ratios and most of the ships are very large. It is a challenging remote sensing ships dataset. Our method has excellent performance on HRSC2016. The experimental results are shown in Table VII. When the size of the input images is $416 \times 416$, the mAP is 90.29%. Compared with the state-of-the-arts methods, such as R$^3$Det, CSL, and DAL, the detection performance is significantly improved. Fig. 9 shows some detection results of our method on HRSC2016. From Fig. 9, we can see that our method has excellent performance with various scenes.

2) Results on UCAS-AOD: The UCAS-AOD dataset contains two categories of targets: cars and planes. Most of the objects in UCAS-AOD are closely arranged and extremely small, especially for cars. It can be seen from Table VIII that our result is the best and the mAP is 97.20%. Moreover, by observing the test results on different categories in UCAS-AOD, we can see that our method performs best on both categories of cars and planes. The detection performance on small cars has been significantly improved compared with other methods. It shows that our method is also robust to small objects. Fig. 10 shows some example results of our method on UCAS-AOD.
Fig. 11. Visual comparison of SSDD+. (a) Densely distributed small ships on the sea. (b) Densely distributed small ships near the coast. (c) Scene of small and medium ships near the coast. (1)–(6) represent the ground-truth and the experimental results of RetinaNet-H, CSL, R3Det, BBAVectors, improved YOLOv4-CSP, respectively.

The experimental results show that our method has excellent performance on scenes with densely arranged objects and small objects.

3) Results on SSDD+: Most of the ships in the SSDD+ dataset are extremely small, and there are abundant coherent speckle noise in SAR images. Hence, SSDD+ is a challenging remote sensing dataset with ships. Table IX shows the experimental results of our method and the state-of-the-arts methods on SSDD+. From the table, we can see that our result is the best and the mAP reaches 97.51%. Compared with other models, our result has been significantly improved, and the mAP has been improved by 5% to 6%. It can be seen that our method has excellent performance on the ship detection task on SAR images. Fig. 11 shows the partial detection results of our method and current state-of-the-art models on the SSDD+ dataset. Fig. 12 additionally shows some example results of our method on
Through the comparison of various aspects, it can be seen that our method has excellent performance in the task of ship detection in SAR images.

The aforementioned experimental results validate that our proposed method has superior performance on the HRSC2016, UCAS-AOD, and SSDD+ datasets.

### H. Detection Speed

Another contribution of our approach is to introduce the YOLOv4-CSP network into the task of arbitrarily oriented object detection and use it as the base network. We improve some contents of the YOLOv4-CSP network to make it suitable for the task of arbitrarily oriented object detection. Then, we combine the improved network with our proposed method. The final experimental results show that the model obtained by training the combined network not only significantly improves the detection accuracy of arbitrarily oriented objects, but also improves the efficiency on object detection.

In the test environment based on Tesla P100, we compare the detection speed of our method with the state-of-the-arts methods on SSDD+. The results are shown in Table X. When the input image size is $800 \times 800$, the mAP from the YOLOv4-CSP_DA model reaches 97.51% and the detection speed reaches 18.2 fps. The detection accuracy and detection speed are both significantly improved comparing with other models. When the input image size is $416 \times 416$, the mAP from the YOLOv4-CSP_DA model reaches 96.25%, which is still much better than other models. What is more prominent is that the detection speed reaches 58.1 fps. Fig. 13 shows the detection accuracy and detection speed of each model, which clearly illustrate the effectiveness and efficiency of our proposed method.

It can be seen from these results that by combining the improved YOLOv4-CSP, the proposed bounding box regression loss function, and the improved anchors, the final model improves the accuracy of the arbitrarily oriented object detection with a high computational efficiency.
In this article, we propose a method for detecting arbitrarily oriented objects in remote sensing images. Considering to further improve the performance while maintaining a high detection speed, we choose the YOLOv4-CSP network as our base network and combine our proposed bounding box regression loss function with improved anchors. We demonstrate the effectiveness of our works based on three datasets. The final experimental results show that our method not only improves the detection accuracy on remote sensing images, but also has an excellent detection speed. Our work is of considerable value on applications that require high detection accuracy and high detection speed for vehicles. In the future, we will try to carry out more intensive experiments for the other classes and datasets, such as the large-scale dataset DOTA, which contains more complex images and object classes.

**REFERENCES**

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.

[2] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 1440–1448.

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards realtime object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

[4] W. Liu et al., “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 21–37.

[5] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, “DSSD: Deconvolutional single shot detector,” 2017, arXiv:1701.06659. [Online]. Available: http://arxiv.org/abs/1701.06659

[6] Z. Li and F. Zhou, “FSSD: Feature fusion single shot multibox detector,” 2017, arXiv:1712.00960. [Online]. Available: http://arxiv.org/abs/1712.00960

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 779–788.

[8] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 7263–7271.

[9] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, arXiv: 2004.10934. [Online]. Available: https://arxiv.org/abs/2004.10934

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “Scaled-YOLOv4: Scaling cross stage partial network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 13029–13038.

[12] R. Nabati and H. Qi, “RRPN: Radar region proposal network for object detection in autonomous vehicles,” in Proc. IEEE Int. Conf. Image Process., 2019, pp. 3093–3097.

[13] Q. Ming et al., “Dynamic anchor learning for arbitrary-oriented object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2355–2363.

[14] X. Yang et al., “SCRDet: Towards more robust detection for small, cluttered and rotated objects,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2019, pp. 8232–8241.

[15] X. Yang, J. C. Yan, Z. T. Feng, and T. He, “R3Det: Refined single-stage detector with feature refinement for rotating object,” in Proc. Assoc. Adv. Artif. Intell., 2021, pp. 3163–3171.

[16] X. Yang, J. C. Yan, and T. He, “Arbitrary-oriented object detection with circular smooth label,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 677–694.

[17] X. Yang, L. Hou, Y. Zhou, W. Wang, and J. Yan, “Dense label encoding for boundary discontinuity free rotation detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jul. 2021, pp. 15819–15829.

[18] J. Ding, N. Xue, Y. Long, G. Xia, and Q. Lu, “Learning RoI transformer for detecting oriented objects in aerial images,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 2849–2858.

[19] Y. Xu et al., “Gliding vertex on the horizontal bounding box for multi-oriented object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 4, pp. 1452–1459, Apr. 2021.

[20] Q. Song, F. Yang, L. Yang, C. Liu, M. Hu, and L. Xia, “Learning point-guided localization for detection in remote sensing images,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 14, pp. 1084–1094, 2021.

[21] S.-B. Chen, B.-M. Dai, J. Tang, B. Luo, W.-Q. Wang, and K. Lv, “A refined single-stage detector with feature enhancement and alignment for oriented objects,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 14, pp. 8898–8908, 2021.

[22] B. Bao, X. Zhong, R. Zhu, X. Zhang, Z. Li, and M. Li, “Single shot anchor refinement network for oriented object detection in optical remote sensing imagery,” IEEE Access, vol. 7, pp. 87150–87161, 2019.

[23] S. M. Azimi et al., “Towards multi-class object detection in unconstrained remote sensing imagery,” in Proc. Assoc. Adv. Artif. Intell., 2018, pp. 150–165.

[24] C. Li, C. Xu, Z. Cui, D. Wang, T. Zhang, and J. Yang, “Feature-attention object detection in remote sensing imagery,” in Proc. IEEE Int. Conf. Inf. Process., 2019, pp. 3886–3890.

[25] L. Zhou, H. Wei, H. Li, W. Zhao, Y. Zhang, and Y. Zhang, “Arbitrary-oriented object detection in remote sensing images based on polar coordinates,” IEEE Access, vol. 8, pp. 223373–223384, 2020.

[26] J. Yi, P. Wu, B. Liu, Q. Huang, H. Qu, and D. Metaxas, “Oriented object detection in aerial images with box boundary-aware vectors,” in Proc. IEEE/CVF Winter Conf. Comput. Vis., 2021, pp. 2150–2159.

[27] D. Zhou et al., “IoU loss for 2D/3D object detection,” in Proc. 4th Int. Conf. 3D Vis., 2019, pp. 85–94.

[28] H. Law and J. Deng, “CornerNet: Detecting objects as paired keypoints,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 734–750.

[29] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint triplets for object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 6569–6578.

[30] X. Zhou, J. Zhuo, and P. Krahmenbuhl, “Bottom-up object detection by grouping extreme and center points,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 850–859.

[31] Z. Liu et al., “A high resolution optical satellite image dataset for ship recognition and some new baseline,” in Proc. Int. Conf. Pattern Recognit. Appl. Methods, 2017, pp. 324–331.

[32] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, “Orientation robust object detection in aerial images using deep convolutional neural network,” in Proc. IEEE Int. Conf. Inf. Process., 2015, pp. 3735–3739.

[33] J. Li, C. Qu, and J. Shao, “Ship detection in SAR images based on an improved faster R-CNN,” in Proc. SAR Big Data Era, Models, Methods Appl., 2017, pp. 1–6.

[34] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 658–666.

[35] Z. Zheng et al., “Distance-IoU loss: Faster and better learning for bounding box regression,” in Proc. Assoc. Adv. Artif. Intell., 2020, pp. 12993–13000.
Ziyang Zheng was born in 1996. He received the B.E. degree in computer science and technology from the Hebei University of Engineering, Handan, China, in 2015. His main research interests include deep learning and computer vision.

Ting Zhang was born in 1986. She received the Ph.D. degree in computer science and technology from the Beijing University of Technology (BJUT), Beijing, China, in 2018. From 2018 to 2020, she was a Postdoctoral Researcher with the BJUT, where she is currently a Lecturer. Her current research interests include pattern recognition, deep learning, and natural language processing. Dr. Zhang is a Member of China Computer Federation (CCF).

Zhaoying Liu was born in 1986. She received the Ph.D. degree in pattern recognition and intelligent system from Beihang University, Beijing, China, in 2015. From 2015 to 2017, she was a Postdoctoral Researcher with the Beijing University of Technology, where she is currently an Associate Professor with the Faculty of Information Technology. Her main research interests include image processing, pattern recognition, and deep learning. Dr. Liu is a Member of China Computer Federation (CCF) and Chinese Association for Artificial Intelligence (CAAI).

Bo Liu received the M.S. degree in theoretical physics and the Ph.D. degree in computer science, both from the Beijing University of Technology, Beijing, China, in 2000 and 2005, respectively. He is currently an Associate Professor with the Faculty of Information Technology, Beijing University of Technology. His current research interests include machine learning and computer vision.

Yujiian Li received the B.S. degree in applied mathematics from the Department of Applied Mathematics, Huazhong University of Science and Technology, Wuhan, China, in 1990, the M.S. degree in computer science and technology from the Institute of Mathematics, Chinese Academy of Sciences, Beijing, China, in 1993, and the Ph.D. degree in artificial intelligent from the Institute of Semiconductors, Chinese Academy of Sciences, in 1999. From 2001 to 2019, he worked as a Professor with the College of Computer Science and Technology, Beijing University of Technology. From 1993 to 1996, he was with the Institute of Biophysics of the Chinese Academy of Sciences. From 1999 to 2001, he was a Postdoctoral Fellow with the Beijing University of Posts and Telecommunications. He is currently a Professor with the Institute of Artificial Intelligence, Guilin University of Electronic Science and Technology, Guilin, China. His current research interests include image processing, pattern analysis, deep learning, and machine intelligence.

Changming Sun received the Ph.D. degree in computer vision from the Imperial College London, London, U.K., in 1992. He then joined CSIRO, Sydney, Australia, where he is currently the Principal Research Scientist carrying out research and working on applied projects. He is also a Conjoint Professor with the School of Computer Science and Engineering, University of New South Wales. His current research interests include computer vision, image analysis, and pattern recognition. Dr. Sun has served on the program/organizing committees of various international conferences. He is an Associate Editor for the EURASIP Journal on Image and Video Processing.