Adaptive Period Embedding for Representing Oriented Objects in Aerial Images

Yixing Zhu, Jun Du, and Xueqing Wu

***Abstract*— We propose a novel method for representing oriented objects in aerial images named adaptive period embedding (APE). Although traditional object detection methods represent objects using horizontal bounding boxes, the objects in aerial images are oriented. Calculating the angle of the object is a yet challenging task. Almost all previous object detectors for aerial images directly regress the angle of objects, they use complex rules to calculate the angle, and their performance is limited by the rule design. In contrast, our method is based on the angular periodicity of oriented objects. The angle is represented by two 2-D periodic vectors the periods of which are different, so the vector is continuous as the shape changes. The label generation rule is simpler and more reasonable compared with previous methods. The proposed method is general and can be applied to other oriented detector. Besides, we propose a novel intersection over union (IoU) calculation method for long objects named length-independent IoU (LIIoU). We intercept part of the long side of the target box to get the maximum IoU between the proposed box and intercepted target box. Thereby, some long boxes will have corresponding positive samples. Our method reaches the first place of DOAI2019 competition task1 (oriented object) held in a workshop on detecting objects in aerial images in conjunction with IEEE CVPR 2019.**

***Index Terms*— Aerial images, convolutional neural networks (CNN), deep learning, intersection over union (IoU), oriented object detection.**

I. Introduction

TRADITIONAL object detection methods mainly detect objects with horizontal bounding boxes. However, objects in aerial images are oriented and cannot be effectively represented by horizontal bounding boxes. As shown in Fig. 1, detecting oriented objects with horizontal bounding boxes will contain more background and cannot accurately locate the objects. Besides, overlap calculation based on horizontal bounding box is not accurate for oriented objects, as the overlap between horizontal bounding boxes of two oriented objects may be too large; thus, nonmaximum suppression (NMS) based on horizontal bounding boxes is not suitable for oriented objects. Therefore, representing oriented objects with an oriented bounding box is necessary for object detection in aerial images. However, a regressing oriented bounding box is more challenging than regressing a horizontal bounding box. Four variables can represent a horizontal bounding box, such as x, y coordinates of top left corner and bottom right corner. However, oriented bounding box representation needs an extra variable θ to represent its angle. It is hard to directly regress θ because the angle is periodic.

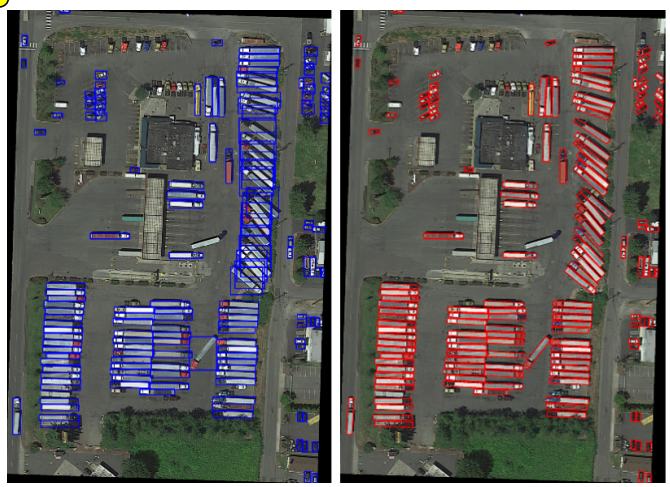


Fig. 1. (Left) Horizonal bounding boxes. (Right) Oriented bounding boxes.

Most of the previous oriented detectors [1]–[5] directly regress θ or the four vertices of the oriented bounding box, and the label is calculated by complicated rules, which is hard for the network to learn. Some methods try to design a simple label calculation rule for oriented objects. For example, Dai *et al.* [6] adopts mask region-convolutional neural networks (CNN) (R-CNN) [7] for detecting oriented text lines. Zhu *et al.* [8] regressed the outline of the objects with multiple points on sliding lines. But these methods introduced additional parameters and cannot be adopted by the region proposal networks (RPNs).

In this article, we propose a novel method for representing oriented objects. The oriented bounding box can be represented by (x, y, w, h, and θ), where x and y are the coordinates

Manuscript received July 29, 2019; revised December 23, 2019; accepted March 10, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1002202 and in part by the National Natural Science Foundation of China under Grant 61671422 and Grant U1613211. *(Corresponding author: Jun Du.)*

Yixing Zhu and Jun Du are with the National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei 230027, China (e-mail: zyxsa@mail.ustc.edu.cn; jundu@ustc.edu.cn).

Xueqing Wu was with the National Engineering Laboratory for Speech and Language Information Processing, University of Science and Technology of China, Hefei 230027, China. She is now with the Electronic Engineering and Information Science Department, University of Science and Technology of China, Hefei 230027, China (e-mail: jwuwuwu24@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2020.2981203

of the center of the bounding box, and w and h are the lengths of the sizes of long and short sides, respectively. We do not directly regress θ. Angle is different from other variables, it has periodicity, and the period of square is 90°, whereas the period of rectangle is 180°. Directly regressing the orientation parameter (θ) will lead to ambiguity. For example, a rectangle whose angle is 359°, another one's angle is 0°. Their vision is quite similar, but the labels are completely different, which is unreasonable. Considering this, θ is represented by two 2-D periodic vectors. The proposed method is different from [9], as in our method, the periods of two vectors are 90° and 180°, respectively. Finally, we calculate the angle with these vectors. Our method is versatile and can be applied to other detectors. Besides, we design a novel cascade R-CNN method for long objects such as harbors. Generally, a two-stage model proposes bounding boxes with RPN in the first stage, and the output bounding boxes of the second stage (R-CNN) are limited by RPN results. Due to the limited size of the receptive field, some long objects cannot be covered by RPN. With this in mind, we adopt a two-stage cascade R-CNN model with length-independent intersection over union (IoU) (LIIoU) to detect long objects. In the first stage, some bounding boxes which only cover part of the objects are also set to positive samples. In this way, the first R-CNN can propose longer bounding boxes. The main contributions of our article are summarized as follows.

1) We present a novel method for representing oriented bounding boxes in aerial images. We do not directly regress θ of oriented bounding boxes, but instead embed θ with vectors whose periods are different. In this way, we do not need complex rules to label the angle which avoids ambiguity.
2) We present a novel IoU calculation method named LIIoU, which is designed for long objects. The presented method makes the detector more robust to long objects.
3) The presented method achieves state of the art on DOTA and wins the first place of Challenge-2019 on Object Detection in Aerial Images task1 (oriented task) in conjunction with IEEE CVPR 2019.

II. Related Work

A. Horizontal Objects Detection

Labels of traditional object detection tasks are horizontal bounding boxes. Ren *et al.* [10] presented a real-time object detection method based on RPN that shares feature maps of RPN and R-CNN and uses anchors with different sizes and aspect ratios in the RPN stage. Though Faster R-CNN shares feature maps, it still requires much computation in the fully connected (FC) layer of R-CNN. Region-based fully convolutional networks (R-FCNs) [11] present position-sensitive score maps and position-sensitive RoI pooling for saving computation in the R-CNN stage. Scale variation is always a very challenging issue in object detection; to help solve this problem, Lin *et al.* [12] presented feature pyramid networks (FPNs). The FPN generates feature maps of different scales on different layers and detects large objects on higher layers but detects small objects on lower layers; the parameters of an RPN is shared over layers. Based on the FPN, mask R-CNN [7] presents RoIAlign which calculates values in RoI features via bilinear interpolation instead of maximum pooling to avoid quantization errors and adds several convolution layers on the mask-head to generate instant segmentation maps. Liu *et al.* [13] improved the mask R-CNN by adding bottom-up path augmentation and feature fusion.

Two-stage methods require more computation than one-stage methods, so one-stage methods are more suitable for real-time object detection tasks. Single shot multibox detector (SSD) [14] generates multiple layers and then detects objects with different sizes on different layers. Deconvolutional single shot detector (DSSD) [15] upsamples feature maps and detects small objects on lower layers which improves SSD performance for small objects. Lin *et al.* [16] presents focal loss to handle the imbalance between positive and negative samples. Although anchors are widely used in object detection, many models adopt the anchor-free method. Huang *et al.* [17] did not use anchors in RPN, but used a shrunk segmentation map as the label. Redmon *et al.* [18] also used segmentation maps as ground truth. The GA-RPN [19] combines anchor-free and anchor-based ideas: the label for the first step is generated by a shrunk segmentation map, and the label for the second step is calculated based on the output anchor of the first step.

Traditional object detection in aerial images only focuses on the horizontal bounding box. Yang *et al.* [20] focused on detecting small urban elements in mobile mapping system (MMS) images. Leng *et al.* [21] tried to detect ship in single-channel synthetic aperture radar (SAR) imagery with complex signal kurtosis. Li *et al.* [22] presented local-contextual feature fusion network which is designed for remote sensing images. Its RPN includes multiangle, multiscale, and multiaspect-ratio anchors which can deal with oriented objects, but the final output bounding boxes are still horizontal. Wang *et al.* [23] presented a rotation-invariant matrix (RIM) which can get both the angular spatial information and radial spatial information. Long *et al.* [24] presented an automatic and accurate localization method for detecting objects in high-resolution remote sensing images based on Faster R-CNN. Salberg [25] presented a method to detect seals in aerial remote sensing images based on a convolutional network. Chen *et al.* [26] presented a hybrid DNN (HDNN), the last convolutional and max-pooling layers of DNN of which are divided into multiple blocks, so HDNN can generate multiscale features that improve the detector performance for small objects. Unlike the images used for general object detection, aerial images have higher resolutions. However, large models cannot be implemented due to memory limitations. Therefore, Pang *et al.* [27] proposed a self-reinforced network named remote sensing region-based CNN (R2-CNN) including Tiny-Net and intermediate global attention blocks. It adopts a lightweight residual structure, so the network can feedforward high-resolution sensing images at high speeds. Deng *et al.* [28] proposed a novel method for ship detection in SAR images. It redesigns the network structure, does not pretrain on ImageNet, and specifically designs the system for small objects such as ships.

B. Oriented Objects Detection

Oriented object detection is first presented in the field of text detection. Liao *et al.* [29] presented a novel SSD-based text detection method, which adapts the size and aspect ratio of the anchor and uses 1×5 convolutional filters for long text lines. Textboxes++ [2] are based on textboxes but directly regress the eight vertices of the oriented bounding box. Liu and Jin [30] designed rules for calculating the order of the vertices of the oriented bounding box and proposed parallel IoU computation to save time. Ma *et al.* [3] presented rotation RPNs (RRPNs) that proposed oriented bounding boxes in the RPN stage and used rotation region-of-interest (RRoI) pooling layer in the R-CNN stage. The aspect ratio of text lines varies greatly, and limited anchors cannot cover the size or aspect ratio of all objects; thus, many methods are anchor-free. Both [1] and [4] generate labels with shrunk segmentation maps and regress the vertices or angles of the bounding box on positive pixels. Lyu *et al.* [31] generated a corner map and a position-sensitive segmentation map, calculated oriented bounding boxes based on the corner map, and computed the score for each bounding box using the position-sensitive segmentation map. Zhong *et al.* [32] presented anchor-free RPN (AF-RPN) based on Faster R-CNN with the same design as the FPN [12] and the label is calculated from the shrunk segmentation map instead of the anchors. Zhang *et al.* [33] proposed an arbitrary-oriented ship detection method for ship detection from remote sensing images. Yang *et al.* [34], [35] proposed arbitrarily oriented ship detection methods based on CNN.

Horizontal bounding boxes cannot closely surround the objects in aerial images, so the academic community begins to pay attention to oriented bounding box detection in aerial images. Xia *et al.* [36] labeled a large-scale data set which contains 15 categories and 188 282 instances, each labeled with an arbitrary quadrilateral (eight vertices). A novel detector which directly regresses eight vertices based on Faster R-CNN is also presented. ICPR ODAI [37] and CVPR DOTA [38] competitions are organized based on this data set. Ding *et al.* [5] presented a two-stage R-CNN method with RoI transformer, which, in the first step, proposed horizontal bounding boxes. The first R-CNN outputs oriented bounding boxes, and the inputs of the second R-CNN are oriented bounding boxes. Li *et al.* [39] proposed a novel method named rotatable region-based residual network (R3-Net) which can detect multioriented vehicles in aerial images and videos. The rotatable RPN (R-RPN) is adopted to generate rotatable regions of interest (R-RoIs) which crops rotated rectangle areas from feature maps.

III. Method

A. Deep Learning-Based Object Detection

In recent years, deep learning-based object detection methods are widely used. In this article, our method is based on Faster R-CNN [10] including RPN and R-CNN. Its process is as follows.

1) Using a CNN such as VGG [40] or ResNet [41] to extract features of the picture. Then each pixel of features predicts whether there is a target near the current point and coordinates of the target. Finally, the network gets N candidate boxes. This process is called RPN. In the RPN stage, traditional method directly predicts θ, but our adaptive period embedding (APE) replaces θ with ($\mathbf{u}_1$ and $\mathbf{u}_2$). In this way, the label is more reasonable and there is no ambiguity in APE. The APE can be used in regressing methods such as RPN, YOLO, and SSD.
2) The N candidate boxes in the RPN stage are not accurate. Therefore, the network will classify and regress these boxes again with more powerful features. R-CNN is used in this stage to extract the features of the entire object. In the R-CNN stage, we use LIIoU for long objects, LIIoU is useful in these data sets which have many long objects. LIIoU can be used in cascade R-CNN, it can set more bounding boxes to positive ones in the R-CNN stage for long objects by replacing the IoU calculation method with LIIoU. In this way, the bounding boxes of long objects will be more accurate.

B. Overview

The overall pipeline of our proposed method is shown in Fig. 2. Recently, anchor-free methods [17]–[19], [32] are widely used in object detection. In this article, we also use AF-RPN. In particular, the label of RPN is not calculated based on the overlap between the anchor and ground truth; instead, the label is generated from the shrunk segmentation map of the oriented bounding box. Unlike traditional object detection tasks, the output bounding box of RPN is oriented, so a novel angle embedding method is adopted to better represent oriented bounding boxes. Segmentation maps with eight channels (x, y, w, h, angle embedding) are generated in the RPN stage. Then our model proposes oriented bounding boxes with rotated RoIAlign in the R-CNN stage, where a cascade R-CNN is used. In the first R-CNN, a novel IoU calculation method named LIIoU is adopted. To make IoU independent of the length of the target box, we intercept part of the long side of the target box to obtain the maximum IoU between the proposed box and intercepted target box. In this way, some long boxes will also have corresponding positive samples. In the second R-CNN, the traditional IoU calculation method is used. The backbone of our network is based on the FPN [12], and we augment the network in the same way as path aggregation network (PANet) [13] by adopting bottom-up path augmentation and feature fusion. Next, we will introduce each component in detail.

C. Network Design

Inspired by recent object detection works [7], [12], [13], we use FPNs as our backbone. FPN generates multiple feature maps of different sizes and detects objects of different sizes on different layers. The FPN is robust to scale variation, especially for small objects, which is suitable for this task. Besides scale variation, aspect ratio variation is another challenging problem. Most traditional object detection methods use anchors of different sizes and aspect ratios to calculate

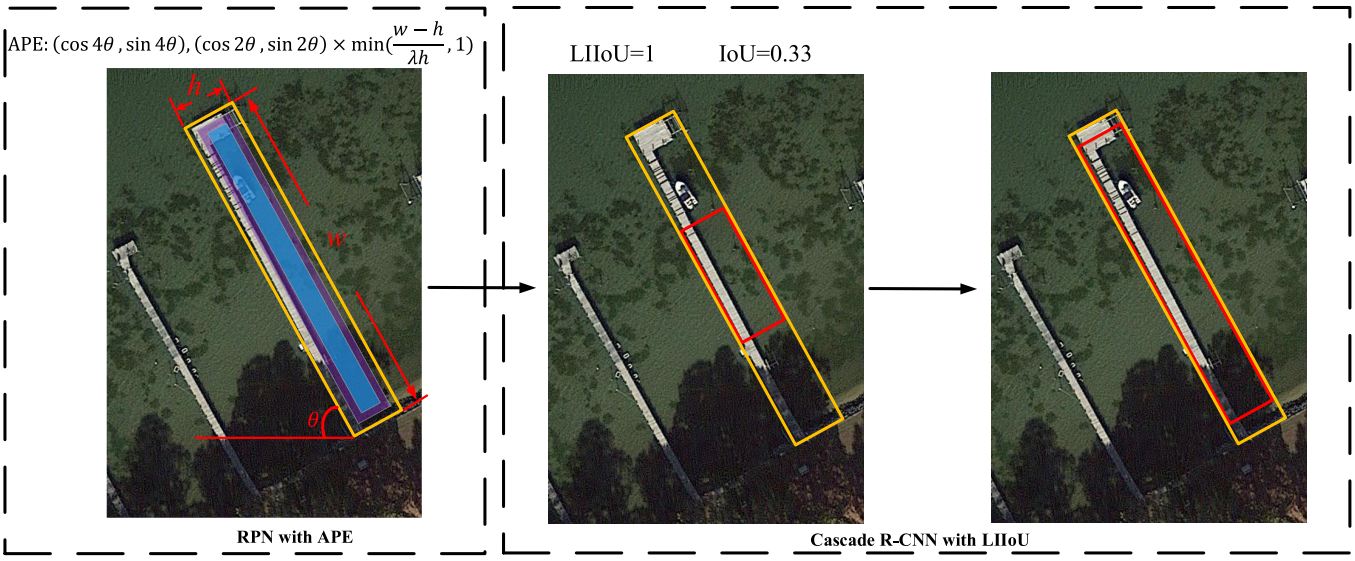


Fig. 2. Illustration of our proposed architecture. (From left to right) AF-RPN, cascade R-CNN. Yellow bounding box is ground truth, red bounding box is proposed box.

labels in the RPN stage. Thus, we have to manually set the hyperparameters of the anchors which is quite troublesome; moreover, when the aspect ratio of objects varies greatly, limited anchors cannot cover all the objects. The performance of these detectors highly relies on anchor design. Recently, many methods [1], [4], [17]–[19], [32] adopt the anchor-free strategy. In this article, we also adopt the anchor-free method and generate the label of RPN from the shrunk segmentation map. Different layers extract different features, and the detector can achieve better performance by combining these features [13]. In the R-CNN stage, we fuse features of different layers after the first FC layer with max pooling.

D. Anchor-Free Label Generation

The RPN is adopted to propose candidate bounding boxes. Most of the previous methods are based on anchors in this stage. Considering the huge difference in aspect ratio of objects, we use AF-RPN. The shrunk segmentation label is shown in Fig. 3. The shrinking method is the same as EAST [4]. In particular, r_1 is set to 0.1 and r_2 is set to 0.25. We shrink the oriented bounding box with r_2 ratio and set the pixels in the shrunk bounding box to positive samples (blue area). Next, we shrink the oriented bounding box with r_1 ratio, set the pixels in this shrunk bounding box but not in blue area to "do not care" (purple area), and set the loss weight of these pixels to 0. The FPN outputs multiscale feature maps, and we detect objects of different scales on different layers. We assign a target object the shorter side of which is h to the level p_k, and k is calculated as follows:

$$k = \lfloor k_0 + \log_2(h/128) \rfloor \tag{1}$$

where k is the layer that objects should be assigned to; k_0 is the target layer when the height h of the object is greater than 128 and less than 256, which we set to 4. The height and width are among $[0, +\infty]$, log scale transformations can turn this interval into $[-\infty, +\infty]$. The outputting of network without activation is among $[-\infty, +\infty]$. Objects of different scales share the regression and classification parameters of RPN, so the regression targets should be normalized. An oriented bounding box is labeled as

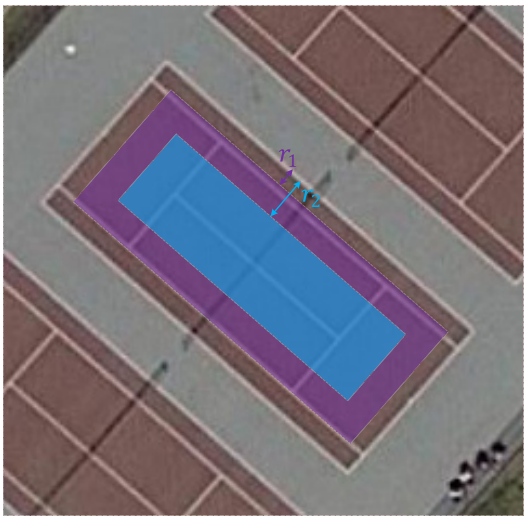


Fig. 3. Shrunk segmentation label of anchor-free method. Purple area is the ignored area which is shrunk with r_1 ratio, blue area is the positive area which is shrunk with r_2 ratio.

$$(x_c, y_c, w, h, \theta) \tag{2}$$

where (x_c, y_c) are the coordinates of the center point, w and h are the lengths of the long side and short side, respectively, and θ is the angle of the long side. The pixel on the kth layer is labeled as x_k, y_k. First, we normalize the target bounding box with the stride of the kth layer

$$x_c' = \frac{x_c}{s_k}, \quad y_c' = \frac{y_c}{s_k}, \quad w' = \frac{w}{s_k}, \quad h' = \frac{h}{s_k}, \quad \theta' = \theta \tag{3}$$

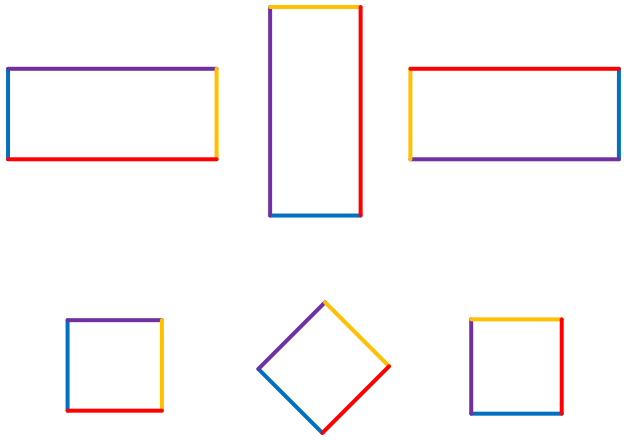


Fig. 4. Period of oriented bounding box. (Top) Rectangle the period of which is 180°. (Bottom) Square the period of which is 90°. (The four sides of each bounding box are in different colors for better visualization.)

where s_k is the stride of the kth layer calculated as

$$s_k = 2 \times 2^k. \tag{4}$$

The regression targets are calculated as follows:

$$t_{x_c} = \frac{x_c' - x_k}{N}, \quad t_{y_c} = \frac{y_c' - y_k}{N}$$
$$t_w = \log \frac{w'}{N}, \quad t_h = \log \frac{h'}{N} \tag{5}$$

where N is a constant and is set to 6 as default.

E. Adaptive Period Embedding

A horizontal bounding box can be easily represented by four variables (x, y, w, h). But we need an extra variable θ to represent an oriented bounding box. The primary challenge of oriented bounding box detection is to regress the angle of objects. The property of θ is different from other variables, such as (x, y, w, h), as θ is a periodic variable. As shown in Fig. 4, if the length and width of the rectangle are equal, the rectangle is a square, and the period of θ is 90°. Otherwise, the period of θ is 180°. In neural networks, the periodicity cannot be represented by one variable. Though [9], [42], [43] all use 2-D periodic vector $(\cos\theta, \sin\theta)$ for representing the angle, they do not adapt the period of vector. The proposed APE uses two 2-D vectors to represent the angle. The first vector has a period of 90° and can be formulated as

$$\mathbf{u}_1 = (\cos 4\theta, \sin 4\theta) \tag{6}$$

where θ is the angle of long side of the rectangle. The period of the second vector is 180°. It is calculated as follows:

$$\mathbf{v} = (\cos 2\theta, \sin\ 2\theta) \tag{7}$$
$$\mathbf{u}_2 = \mathbf{v} \times \min\left(\frac{(w-h)}{\lambda h}, 1\right) \tag{8}$$

where λ is set to 0.5, w is the long side of the rectangle, and h is the short side. Each component of $\mathbf{u}_1, \mathbf{u}_2$ is in $[-1, 1]$, so we use sigmoid as activation, and then multiply them by 2 and subtract 1. Smooth $L1$ loss [44] is used in all regression tasks of this article which can be formulated as

$$\text{smooth}_{L_1}(z, z^*) = \begin{cases} 0.5\,(z - z^*)^2, & \text{if } |z - z^*| < 1 \\ |z - z^*| - 0.5, & \text{otherwise.} \end{cases} \tag{9}$$

The final outputs of the neural network are $(x, y, w, h, \mathbf{u}_1, \mathbf{u}_2)$. Next, we calculate the angle of long side of the rectangle based on $(\mathbf{u}_1, \mathbf{u}_2)$. First, θ_{90° the period of which is 90° and can be calculated as

$$\theta_{90^\circ} = \frac{\text{atan2}\,(\mathbf{u}_1)}{4} \tag{10}$$

where atan2 function calculates one unique arctangent value from a 2-D vector. The θ of rectangle's long side may be θ_{90° or $\theta_{90^\circ} + 90^\circ$. The θ_{180° the period of which is 180° can be calculated as

$$\theta_{180^\circ} = \frac{\text{atan2}\,(\mathbf{u}_2)}{2}. \tag{11}$$

Then we calculate the distance between θ_{90° and θ_{180°

$$\text{dis} = |\,(2\theta_{90^\circ} - 2\theta_{180^\circ} + 180^\circ) \mod 360^\circ - 180^\circ|. \tag{12}$$

Therefore, the final θ is calculated as

$$\theta = \begin{cases} \theta_{90^\circ}, & \text{dis} < 90^\circ \\ \theta_{90^\circ} + 90^\circ, & \text{otherwise.} \end{cases} \tag{13}$$

If they are all squares (the gap between long and short sides is zero), only using $\mathbf{u}_1$ can represent its angle. If they are all rectangles (the gap between long and short sides is large), only using $\mathbf{u}_2$ can represent its angle. But in DOTA, the gap between long and short sides may be large or zero. So, both of them are needed. θ_{90° encodes angles into vectors with a 90° period and θ_{180° encodes angles into vectors with a 180° period. The angle calculated by θ_{90° may be the angle of the long or short side. When the sizes of long and short sides are equal, there is no additional information required. But if long and short sides are not equal, other information to represent the angle of the long side is needed. $\mathbf{u}_2 = \mathbf{v} \times \min((w-h)/\lambda h, 1)$ which means the angle of long side. We do not directly regress $\mathbf{v}$ because there is ambiguity for $\mathbf{v}$. For example, when height and width are equal, the angle of the long side is ambiguous, the $\mathbf{v}$ will produce mutation. But $\mathbf{u}_2$ continuously changes as long or short sides change. When the sizes of long and short sides are equal, $\mathbf{u}_2$ is zero, the changing is also continuous. We use sigmoid as activation, accordingly, we set the maximum of $\mathbf{u}_2$ to 1. The distance between θ_{90° and θ_{180° is $\text{dis} = |(2\theta_{90^\circ} - 2\theta_{180^\circ} + 180^\circ) \mod 360^\circ - 180^\circ|$. θ_{90° maybe the angle of the long or short side, the final angle of the long side may be θ_{90° or $\theta_{90^\circ} + 90^\circ$. We find the closer angle among θ_{90°, $\theta_{90^\circ} + 90^\circ$, and θ_{180°. The distance between θ_{90° and θ_{180° is the distance of two periodic vectors which is the closer rotated angle from θ_{90° to θ_{180° clockwise or counterclockwise.

F. Length-Independent IoU

IoU is the evaluation protocol of object detection; the more accurate the regression, the better the performance. But the receptive field of a neural network is limited and thus cannot cover some long objects. The detector proposes candidate

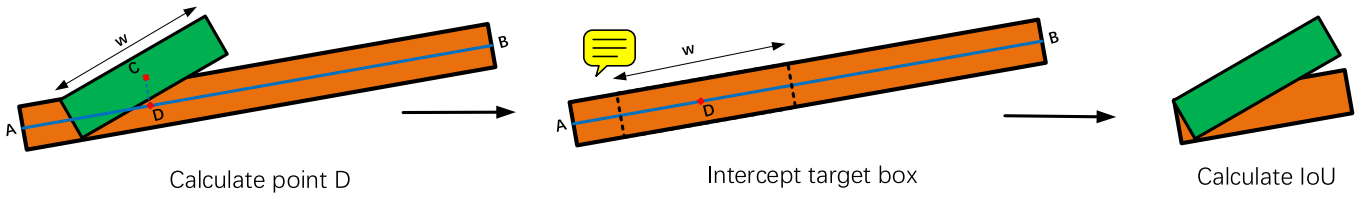


Fig. 5. Details of LIIoU calculation, green bounding box is proposed bounding box, orange bounding box is target box.

bounding boxes in the RPN stage and then classifies and regresses these boxes again. The result of R-CNN highly relies on the output bounding boxes of the RPN. In the R-CNN stage, only the proposed bounding boxes whose IoU is higher than 0.5 is set to positive samples. Some target objects that are not well regressed in the RPN cannot be detected in the R-CNN. One idea is multiple regression [45] in the R-CNN stage, but if there are no positive proposed bounding boxes in the first R-CNN, the improvement is limited. Considering this, we propose a novel IoU calculation method named LIIoU. We intercept part of the target box along its long side and make the length of the intercepted box the same as the proposed box. The presented method is inspired by Seglink [46], but in our method, the aspect ratio of the proposed bounding box is arbitrary. As shown in Fig. 2, the traditional IoU is about 0.3, but our proposed LIIoU is nearly 1. The details of the LIIoU calculation are illustrated in Fig. 5, where AB is the centerline of the target box, and point C is the center of the proposed box. First, we find the perpendicular of AB through point C and label the intersection of the perpendicular and AB as point D. Next, we intercept a rectangle from the target bounding box as follows: if the length of the target box is smaller than the proposed box, we do not intercept; otherwise, the center of the intercepted rectangle is point D and the length is the same with the proposed box (green box). Finally, we calculate IoU between the intercepted target box and the proposed box. The procedure is summarized in Algorithm 1. In this way, more bounding boxes will be utilized to regress targets in R-CNN which can improve the overall quality of the bounding boxes.

In the first stage, we set more bounding boxes to the positive ones for long targets. In this way, the network can better regress long samples. But in the last R-CNN, the network will predict the final score of each bounding box, we need to lower the score of the box with poor quality and the evaluation is IoU. Accordingly, we use IoU in the last R-CNN.

Algorithm 1 LIIoU Calculation

Input: $pbbox(x^p, y^p, w^p, h^p, \theta^p)$, $gbbox(x^g, y^g, w^g, h^g, \theta^g)$
$pbbox$ - proposed bounding box
$gbbox$ - ground truth bounding box

Output: LIIoU

if $w^p >= w^g$ **then**
  $x'^g = x^g$; $y'^g = y^g$; $w'^g = w^g$; $h'^g = h^g$; $\theta'^g = \theta^g$
else
  $\mathbf{A}_x = x^g - \cos(\theta^g) \times \frac{w^g}{2}$
  $\mathbf{A}_y = y^g - \sin(\theta^g) \times \frac{w^g}{2}$
  $\mathbf{B}_x = x^g + \cos(\theta^g) \times \frac{w^g}{2}$
  $\mathbf{B}_y = y^g + \sin(\theta^g) \times \frac{w^g}{2}$
  $\mathbf{C}_x = x^p$; $\mathbf{C}_y = y^p$
  $z = \frac{(\mathbf{C}-\mathbf{A})\cdot(\mathbf{B}-\mathbf{A})}{||(\mathbf{B}-\mathbf{A})||}$
  $w_1 = z - \frac{w^p}{2}$; $w_2 = z + \frac{w^p}{2}$
  if $w_1 <= 0$ **then**
    $w_1 = 0$; $w_2 = w^p$
  else if $w_2 >= w^g$ **then**
    $w_2 = w^g$; $w_1 = w^g - w^p$
  end if
  $x'^g = A_x + \cos(\theta) \times \frac{w_2+w_1}{2}$; $y'^g = A_y + \sin(\theta) \times \frac{w_2+w_1}{2}$
  $w'^g = w_2 - w_1$; $h'^g = h^g$; $\theta'^g = \theta^g$
end if
calulate overlaps between $(x^p, y^p, w^p, h^p, \theta^p)$ and $(x'^g, y'^g, w'^g, h'^g, \theta'^g)$

G. Cascade R-CNN

As shown in Fig. 2, two R-CNNs are used after RPN. In the first R-CNN, we only refine the center, height, and width of the oriented bounding box without regressing the vertices of the target box. This is because the output of the first R-CNN is the input of the second R-CNN, and rotated RoIAlign can only handle oriented rectangle but not quadrangle. In the second R-CNN, we regress the vertices of the target box. Our method is different from rotated cascade R-CNN [8], we directly regress oriented bounding boxes in the RPN. The two R-CNNs are both rotated, cascade R-CNN is adopted for improving the model's performance on long objects.

Rotated RoIAlign is adopted, so the ground truth is calculated in a rotated coordinate system. Following [8], if the center of a rotated RoIAlign is (x_c^p, y_c^p) and the angle is θ^p, the affine transformation can be represented by an affine matrix

$$\begin{aligned}\boldsymbol{M} &= \begin{bmatrix}1 & 0 & x_c^p\\ 0 & 1 & y_c^p\\ 0 & 0 & 1\end{bmatrix} * \begin{bmatrix}\cos\theta^p & \sin\theta^p & 0\\ -\sin\theta^p & \cos\theta^p & 0\\ 0 & 0 & 1\end{bmatrix} * \begin{bmatrix}1 & 0 & -x_c^p\\ 0 & 1 & -y_c^p\\ 0 & 0 & 1\end{bmatrix}\\ &= \begin{bmatrix}\cos\theta^p & \sin\theta^p & (1-\cos\theta^p)\,x_c^p - y_c^p * \sin\theta^p\\ -\sin\theta^p & \cos\theta^p & (1-\cos\theta^p)\,y_c^p + x_c^p * \sin\theta^p\\ 0 & 0 & 1\end{bmatrix}\end{aligned} \tag{14}$$

$$\begin{pmatrix}x\\ y\\ 1\end{pmatrix} = \boldsymbol{M}\begin{pmatrix}x'\\ y'\\ 1\end{pmatrix}. \tag{15}$$

We set the coordinate system to rotated coordinate system with (14) and (15). The final ground truth in the rotated coordinate system is (x, y), and (x', y') is the coordinates in the original coordinate system. In the first R-CNN, the regression targets are $(t_{x_c}^{\text{rcnn1}}, t_{y_c}^{\text{rcnn1}}, t_w^{\text{rcnn1}}, t_h^{\text{rcnn1}})$ which can be formulated as

$$t_{x_c}^{\text{rcnn1}} = \frac{x_c - x_c^p}{w^p}; \quad t_{y_c}^{\text{rcnn1}} = \frac{y_c - y_c^p}{h^p} \tag{16}$$

$$t_w^{\text{rcnn1}} = \log\left(\frac{w}{w^p}\right); \quad t_h^{\text{rcnn1}} = \log\left(\frac{h}{h^p}\right). \tag{17}$$

In the second R-CNN, the regression targets are $(t_{x_i}^{\text{rcnn2}}, t_{y_i}^{\text{rcnn2}})$, $i = 1, 2, 3, 4$ which can be formulated as

$$t_{x_i}^{\text{rcnn2}} = \frac{x_i - x_c^p}{w^p}; \quad t_{y_i}^{\text{rcnn2}} = \frac{y_i - y_c^p}{h^p}, \ i = 1, 2, 3, 4 \tag{18}$$

where (x_c, y_c, w, h) are the center, width, and height of the ground truth, (x_i, y_i) is the vertex of the ground truth bounding box, and (x_c^p, y_c^p, w^p, h^p) are the center, width, and height of the proposed bounding box.

IV. Experiment

A. Data Sets

DOTA [36] is a large data set that contains 2806 aerial images from different sensors and platforms. The size of the image varies greatly, ranging from about 800 × 800 to 4000 × 4000 pixels, so it is necessary to crop the images and detect the objects in the cropped images. As the instances in aerial images, such as cars, ships, and bridges, are oriented, each instance is labeled by an arbitrary (eight d.o.f.) quadrilateral. For the oriented object detection task, the output bounding boxes are quadrilateral to evaluate the performance of our detector on the quadrilateral, we use the evaluation system provided along with this data set. There are two versions of the DOTA data set, DOTA-v1.0 and DOTA-v1.5; DOTA-v1.5 fixes some errors and is provided for DOAI2019 competition [38]. We use DOTA-v1.5 for this competition, but in the following experiments, we use DOTA-v1.0 for a fair comparison.

B. Implementation Details

The backbone of our detector is ResNet-50 [41] pretrained on ImageNet [47]. The number of FPN channels is set to 256. In the R-CNN stage, two FC layers are used, the channel of which is set to 1024. Feature fusion is applied after the first FC layer along with maxpooling. Batchnorm is not used in this article. Our network is trained with SGD, where the batch size of images is 1 and the initial learning rate is set to 0.00125, model is trained with about 161 790 iterations, learning rate is then divided by 10 at $2/3$ and $8/9$ of the entire training. Weight decay is set to 0.0001. Due to the limited memory, we crop images to 1024 × 1024 with the stride of 256 for training and testing. We merge all predicted bounding boxes with DOTA's code[1] in testing. The model is trained and tested on a single scale. Data augmentation is used for better performance; in particular, we randomly rotate images with angle among 0, $\pi/2$, π, and $3\pi/2$, and class balance resampling is adopted to solve class imbalance problem. In default, we train our model with the training set and evaluate it on the validation set and testing set.

[1]https://github.com/CAPTAIN-WHU/DOTA_devkit

TABLE I
EXPERIMENT OF APE ON DOTA VALIDATION SET IN RPN STAGE (IN %)

Methods	w/o APE	w/ APE
AP	70.16	72.20

TABLE II
ABLATION EXPERIMENTS OF LIIOU AND IOU ON DOTA VALIDATION SET (IN %)

Methods	Faster R-CNN	Cascade R-CNN	Cascade R-CNN+LIIoU
mAP	71.40	72.76	**73.88**

C. Ablation Study

In order to evaluate the effect of each component, we conduct ablation experiments on the validation set of DOTA. The model is not modified except for the component being tested.

1) Effect of APE: We need to propose oriented bounding boxes in the RPN stage, but it is challenging to effectively represent an oriented bounding box. Most of the previous methods [3]–[5] that directly regress the angle do not notice the periodicity of the angle. When the angle is too diverse, the performance of the system will drop significantly. To evaluate whether the proposed APE can well handle the diversity of the angles, we conduct ablation experiments: one model directly regresses the angle of the long side of the target box, whereas the other regresses APE vectors. We evaluate the quality of the proposed oriented bounding box in the RPN stage. The network only classifies objects into two classes (positive sample and negative sample) in the RPN stage. We use average precision (AP) as our evaluation protocol. AP is average precision which calculates the AP at different recall rates, it is widely used in object detection evaluation. And mAP is the mean of AP on all classes. As shown in Table I, RPN achieves much better performance with APE. We show the comparison in Fig. 6, where we can see that RPN outputs a more accurate angle with APE compared with directly regressing the angle.

2) LIIoU Versus IoU: To evaluate the efficiency of LIIoU, we conduct a control experiment. Faster R-CNN means there is only one R-CNN. When Cascade R-CNN is adopted, two R-CNNs are used. In the first model, we calculate the overlap between oriented bounding boxes with traditional IoU in both two R-CNNs. In the second model, the overlap between oriented bounding boxes is calculated with LIIoU in the first R-CNN and with traditional IoU in the last R-CNN, and the threshold is set to 0.5. Results are shown in Table II, where we can see that cascade R-CNN gains much better performance with LIIoU. We show their comparison in Fig. 7, where we can find that LIIoU can improve the quality of the proposed bounding boxes and the recall rate. Regardless of the aspect

Fig. 6. Comparison of RPN with APE and without APE. (Top) Without APE. (Bottom) With APE.

TABLE III

RESULTS ON DOTA TESTING SET (IN %). * INDICATES VALIDATION SET IS ALSO USED FOR TRAINING, OTHERWISE ONLY TRAINING SET IS USED FOR TRAINING

Method	Ours	Ours *	FR-O [36]	RoI Transformer * [5]	RRPN [3]	R2CNN [48]	R-DFPN [35]	Yang *et al.* [34]
Plane	89.67	89.96	79.09	88.64	80.94	88.52	80.92	81.25
BD	76.77	83.62	69.12	78.52	65.75	71.2	65.82	71.41
Bridge	51.28	53.42	17.17	43.44	35.34	31.66	33.77	36.53
GTF	71.65	76.03	63.49	75.92	67.44	59.3	58.94	67.4
SV	73.11	74.01	34.2	68.81	59.92	51.85	55.77	61.16
LV	77.18	77.16	37.16	73.68	50.91	56.19	50.94	50.91
Ship	79.54	79.45	36.2	83.59	55.81	57.25	54.78	56.6
TC	90.79	90.83	89.19	90.74	90.67	90.81	90.33	90.67
BC	79.01	87.15	69.6	77.27	66.92	72.84	66.34	68.09
ST	84.54	84.51	58.96	81.46	72.39	67.38	68.66	72.39
SBF	66.51	67.72	49.4	58.39	55.06	56.69	48.73	55.06
RA	64.71	60.33	52.52	53.54	52.23	52.84	51.76	55.60
Harbor	73.97	74.61	57.79	62.83	55.14	53.08	55.10	62.44
SP	67.73	71.84	44.8	58.93	53.35	51.94	51.32	53.35
HC	58.40	65.55	46.3	47.67	48.22	53.58	35.88	51.47
mAP	73.66	**75.75**	52.93	69.56	61.01	60.67	57.94	62.2
FPS	4.0	4.0	-	5.9	-	-	-	-

ratio and size, nearly every object has positive samples with LIIoU, so the detector can handle objects with large aspect ratios and lengths well.

D. Comparing With Other State-of-the-Art Methods

We compare our method with other state-of-the-art methods. The results are shown in Table III, results of RRPN, R2CNN, R-DFPN, and Yang *et al.* [34] are from [5]. Our model is trained and tested with the single-scale setting. When our model is only trained with the training set ex validation set, our method significantly outperforms other methods, and if the validation set is also used for training, our model achieves better performance. The detection results are shown in Fig. 8. The angle, size, aspect ratio of objects in aerial images vary greatly, but our proposed method can well handle these challenging conditions. Our model is also efficient that can handle four images whose resolution is 1024/sec.

E. Limitations

We show some failure cases of our method in Fig. 9. When the boundary line of the court is not obvious, our method cannot predict accurate bounding boxes. When the scale of the object is too small, the detector misses these objects. When the shape of the objects is quite special and the shape does not exist in the training set, our method will produce the wrong

TABLE IV
TASK1—ORIENTED LEADERBOARD ON DOAI2019 (IN %)

Team Name	USTC-NELSLIP	pca_lab	czh	AICyber	CSULQQ	peijin
Plane	89.2	88.2	89.0	88.4	87.8	80.9
BD	85.3	86.4	83.2	85.4	83.6	83.6
Bridge	57.3	59.4	54.5	56.7	56.7	55.1
GTF	80.9	80.0	73.8	74.4	74.4	70.7
SV	73.9	68.1	72.6	63.9	63.2	59.9
LV	81.3	75.6	80.3	72.7	71.0	76.4
Ship	89.5	87.2	89.3	87.9	87.8	88.3
TC	90.8	90.9	90.8	90.9	90.8	90.9
BC	85.9	85.3	84.4	86.3	84.6	79.2
ST	85.6	84.1	85.0	85.0	84.0	78.3
SBF	69.5	73.8	68.7	68.9	67.8	59.1
RA	76.7	77.5	75.3	76.0	75.5	74.8
Harbor	76.3	76.4	74.2	74.1	67.4	74.1
SP	76.0	73.7	74.4	72.9	71.2	74.9
HC	77.8	69.5	73.4	73.4	68.8	59.8
CC	57.3	49.6	42.1	37.9	22.5	39.5
mAP	78.3	76.6	75.7	74.7	72.3	71.6

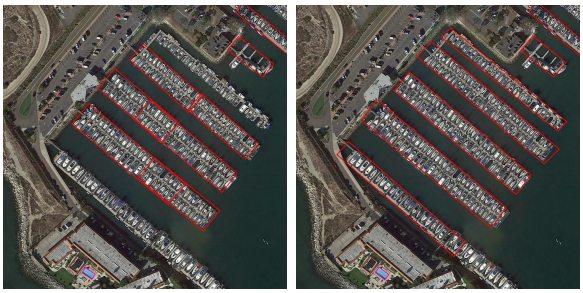


Fig. 7. Comparison of LIIoU and IoU. (From left to right) Calculate overlaps with IoU in the first R-CNN, calculate overlaps with LIIoU in the first R-CNN (overlaps are both calculated with IoU in the last R-CNN).

results. In the future, we will explore a solution for these hard cases.

F. DOAI2019 Competition

DOAI2019 competition [38] is held in the workshop on detecting objects in aerial images in conjunction with IEEE CVPR 2019. The competition is more difficult and requires detecting all objects including samples labeled as difficult. Based on our proposed methods including APE and LIIoU, we adopt class balance resampling, image rotation, multi-scale training and testing and model assembling for better performance. Three models are used whose backbone is ResNeXt-101(32 $\times$ 4) [49]. Finally, we combine the training set with the validation set for training. The results of the competition are shown in Table IV. Our method wins the first place on the oriented task, with a gain of about 1.7% over the most competing competitor.

V. CONCLUSION AND FUTURE WORK

Detecting oriented objects in aerial images is a challenging task. In this article, we make full use of the periodicity of the angle. A novel method named APE is proposed which can well-regress oriented bounding boxes in aerial images. The vector with the adaptive period can learn the periodicity

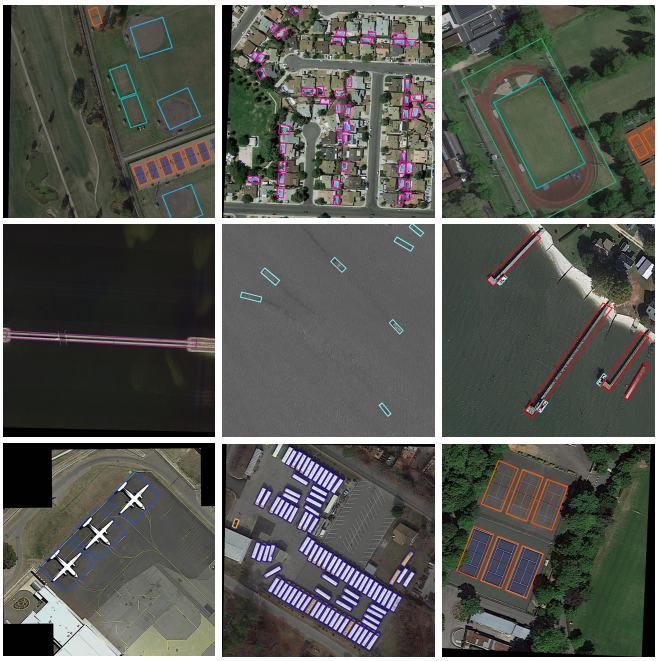


Fig. 8. Some results of our method on DOTA. The image's size is 1024 $\times$ 1024.

of the angle, which cannot be implemented with the 1-D vector. The proposed method can be applied to both one-stage methods such as RPN and two-stage methods, and we believe other detectors can also directly adopt the APE module. Besides, we propose a novel LIIoU. LIIoU sets more proposed bounding boxes to positive samples especially for long objects which can improve the quality of R-CNN regression. Our ablation study proves that each proposed module is effective. Based on our method, we won the first place on the oriented task of DOAI2019. In the future, we will explore a more efficient and accurate detector for detecting oriented objects in aerial images.

Fig. 9. Some failure cases of our method on DOTA. The size of the image is 1024 × 1024.

REFERENCES

[1] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 745–753.

[2] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[3] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[4] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5551–5560.

[5] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning RoI transformer for oriented object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2849–2858.

[6] Y. Dai *et al.*, "Fused text segmentation networks for multi-oriented scene text detection," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 3604–3609.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2961–2969.

[8] Y. Zhu, C. Ma, and J. Du, "Rotated cascade R-CNN: A shape robust detector with coordinate regression," *Pattern Recognit.*, vol. 96, Dec. 2019, Art. no. 106964.

[9] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 19–35.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[11] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[13] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[14] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[15] 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

[16] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2980–2988.

[17] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "DenseBox: Unifying landmark localization with end to end object detection," 2015, *arXiv:1509.04874*. [Online]. Available: http://arxiv.org/abs/1509.04874

[18] 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. (CVPR)*, Jun. 2016, pp. 779–788.

[19] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2965–2974.

[20] Z. Yang, Y. Liu, L. Liu, X. Tang, J. Xie, and X. Gao, "Detecting small objects in urban settings using SlimNet model," *IEEE Trans. Geosci. Remote Sens.*, to be published.

[21] X. Leng, K. Ji, S. Zhou, and X. Xing, "Ship detection based on complex signal kurtosis in single-channel SAR imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6447–6461, Sep. 2019.

[22] K. Li, G. Cheng, S. Bu, and X. You, "Rotation-insensitive and context-augmented object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2337–2348, Apr. 2018.

[23] G. Wang, X. Wang, B. Fan, and C. Pan, "Feature extraction by rotation-invariant matrix representation for object detection in aerial image," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 6, pp. 851–855, Jun. 2017.

[24] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, "Accurate object localization in remote sensing images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2486–2498, May 2017.

[25] A.-B. Salberg, "Detection of seals in remote sensing images using features extracted from deep convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 1893–1896.

[26] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle detection in satellite images by hybrid deep convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1797–1801, Oct. 2014.

[27] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng, "R^2-CNN: Fast tiny object detection in large-scale remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5512–5524, Aug. 2019.

[28] Z. Deng, H. Sun, S. Zhou, and J. Zhao, "Learning deep ship detector in SAR images from scratch," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4021–4039, Jun. 2019.

[29] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A fast text detector with a single deep neural network," in *Proc. AAAI*, 2017, pp. 4161–4167.

[30] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3454–3461.

[31] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7553–7563.

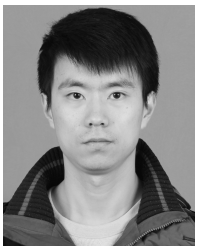
[32] Z. Zhong, L. Sun, and Q. Huo, "An anchor-free region proposal network for faster R-CNN based text detection approaches," 2018, *arXiv:1804.09003*. [Online]. Available: http://arxiv.org/abs/1804.09003

[33] Z. Zhang, W. Guo, S. Zhu, and W. Yu, "Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 11, pp. 1745–1749, Nov. 2018.

[34] X. Yang, H. Sun, X. Sun, M. Yan, Z. Guo, and K. Fu, "Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network," *IEEE Access*, vol. 6, pp. 50839–50849, 2018.

[35] X. Yang *et al.*, "Automatic ship detection in remote sensing images from Google Earth of complex scenes based on multiscale rotation dense

feature pyramid networks," *Remote Sens.*, vol. 10, no. 1, p. 132, 2018.
[36] G.-S. Xia *et al.*, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.
[37] *ICPR-ODAI*. [Online]. Available: https://captain-whu.github.io/ODAI/
[38] *CVPR-DOTA*. [Online]. Available: https://captain-whu.github.io/DOAI2019/challenge.html
[39] Q. Li, L. Mou, Q. Xu, Y. Zhang, and X. X. Zhu, "R^2-net: A deep network for multioriented vehicle detection in aerial images and videos," *IEEE Trans. Geosci. Remote Sens.*, to be published.
[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[42] Y. Zhu and J. Du, "TextMountain: Accurate scene text detection via instance segmentation," 2018, *arXiv:1811.12786*. [Online]. Available: http://arxiv.org/abs/1811.12786
[43] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "TextField: Learning a deep direction field for irregular scene text detection," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5566–5579, Nov. 2019.
[44] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[45] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.
[46] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2550–2558.
[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[48] Y. Jiang *et al.*, "R^2CNN: Rotational region CNN for orientation robust scene text detection," 2017, *arXiv:1706.09579*. [Online]. Available: http://arxiv.org/abs/1706.09579
[49] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.



Yixing Zhu received the B.Eng. degree from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2017, where he is currently pursuing the master's degree.

His current research interests include deep learning, optical character recognition (OCR), and object detection in aerial images.

Jun Du received the B.Eng. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2004 and 2009, respectively.

From 2004 to 2009, he was with the iFlytek Speech Laboratory, USTC. During the above period, he worked as an Intern twice for nine months at Microsoft Research Asia (MSRA), Beijing, China, where he was an Associate Researcher from July 2010 to January 2013, working on handwriting recognition, optical character recognition (OCR), and speech recognition. In 2007, he also worked as a Research Assistant for six months with the Department of Computer Science, The University of Hong Kong, Hong Kong. From July 2009 to June 2010, he worked at iFlytek Research on speech recognition. Since February 2013, he has been with the National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP), USTC.

Xueqing Wu is currently pursuing the bachelor's degree with the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China.

From June 2018 to September 2018, she was with the National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP), USTC, working on object detection. She is also a Research Intern at Microsoft Research Asia (MSRA), Beijing, China, working on neural machine translation.