This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learning RoI Transformer for Oriented Object Detection in Aerial Images

Jian Ding, Nan Xue, Yang Long, Gui-Song Xia,* Qikai Lu
LIESMARS-CAPTAIN, Wuhan University, Wuhan, 430079, China
{jian.ding, xuenan, longyang, guisong.xia, qikai_lu}@whu.edu.cn

Abstract

Object detection in aerial images is an active yet challenging task in computer vision because of the bird's-eye view perspective, the highly complex backgrounds, and the variant appearances of objects. Especially when detecting densely packed objects in aerial images, methods relying on horizontal proposals for common object detection often introduce mismatches between the Region of Interests (RoIs) and objects. This leads to the common misalignment between the final object classification confidence and localization accuracy. In this paper, we propose a **RoI Transformer** *to address these problems. The core idea of RoI Transformer is to apply spatial transformations on RoIs and learn the transformation parameters under the supervision of oriented bounding box (OBB) annotations. RoI Transformer is with lightweight and can be easily embedded into detectors for oriented object detection. Simply apply the RoI Transformer to light-head RCNN has achieved state-of-the-art performances on two common and challenging aerial datasets,* i.e.*, DOTA and HRSC2016, with a neglectable reduction to detection speed. Our RoI Transformer exceeds the deformable Position Sensitive RoI pooling when oriented bounding-box annotations are available. Extensive experiments have also validated the flexibility and effectiveness of our RoI Transformer.*

1. Introduction

Object detection in aerial images aims at locating objects of interest (e.g., vehicles, airplanes) on the ground and identifying their categories. With more and more aerial images being available, object detection in aerial images has been a specific but active topic in computer vision [3, 29, 36, 6]. However, unlike natural images that are often taken from horizontal perspectives, aerial images are typically taken from bird's-eye view, which implies that objects in aerial images are always arbitrary oriented. Moreover, the highly complex backgrounds and variant appearances of objects further increase the difficulty of object detection in aerial images. These problems have been often approached by an *oriented and densely packed object detection* task [37, 31, 12], which is new while well-grounded and have attracted much attention in the past decade [27, 30, 26, 18, 1].

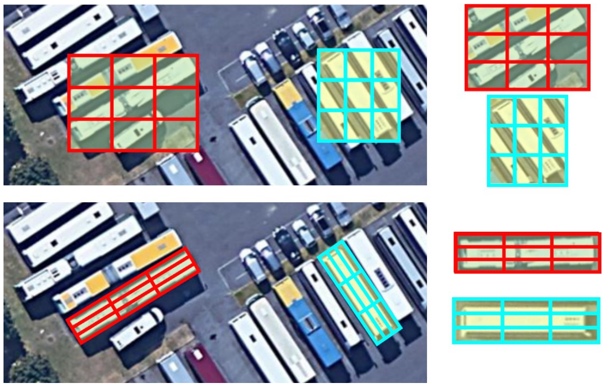


Figure 1. ***Horizontal*** **(top)** ***v.s. Rotated*** **RoI warping (bottom)** illustrated in an image with many densely packed objects. One horizontal RoI often contains several instances, which leads ambiguity to the subsequent classification and location task. By contrast, a rotated RoI warping usually provides more accurate regions for instances and enables to better extract discriminative features for object detection.

Many of recent progress on object detection in aerial images have benefited a lot from the R-CNN frameworks [9, 8, 32, 2, 29, 38, 6, 12, 16]. These methods have reported promising detection performances, by using horizontal bounding boxes as *region of interests* (RoIs) and then relying on region features for category identification [2, 29, 6]. However, as observed in [37, 28], these *horizontal RoIs* (HROIs) typically lead to misalignments between the bounding boxes and objects. For instance, as shown in Fig. 1, due to the oriented and densely-distributed properties of objects in aerial images, several instances are often crowded and contained by one HRoI. As a result, it usually turns to be difficult to train a detector for extracting object features and identifying the object's accurate localization.

Instead of using horizontal bounding boxes, oriented bounding boxes have been employed to give more accurate locations of objects [37, 23, 28]. In order to achieve

[*] Corresponding author: guisong.xia@whu.edu.cn.

high recalls at the phase of RRoI generation, a large number of anchors are required with different angles, scales and aspect ratios. These methods have demonstrated promising potentials on detecting sparsely distributed objects [26, 43, 27, 30]. However, due to the highly diverse directions of objects in aerial images, it is often intractable to acquire accurate RRoIs to pair with all the objects in an aerial image by using RRoIs with limited directions. Consequently, the elaborate design of RRoIs with as many directions and scales as possible usually suffers from its high computational complexity at region classification and localization phases.

As the regular operations in conventional networks for object detection [8] have limited generalization to rotation and scale variations, it is required of some orientation and scale-invariant in the design of RoIs and corresponding extracted features. To this end, Spatial Transformer [14] and deformable convolution and RoI pooling [5] have been proposed to model the geometry variations. However, they are mainly designed for the general geometric deformation without using the oriented bounding box annotation. In the field of aerial images, there is only rigid deformation, and oriented bounding box annotation is available. Thus, it is natural to argue that it is important to *extract rotation-invariant region features* and to *eliminate the misalignment between region features and objects* especially for densely packed ones.

In this paper, we propose a module called RoI Transformer, targeting to achieve detection of oriented and densely-packed objects, by supervised RRoI learning and feature extraction based on position sensitive alignment through a two-stage framework [9, 8, 32, 4, 10]. It consists of two parts. The first is the *RRoI Learner*, which learns the transformation from HRoIs to RRoIs. The second is the *Rotated Position Sensitive RoI Align*, which extracts the rotation-invariant features from the RRoI for following objects classification and location regression. To further improve efficiency, we adopt a light head structure for all RoI-wise operations. We extensively test and evaluate the proposed RoI Transformer on two public datasets for object detection in aerial images *i.e.* DOTA [37] and HRSC2016 [28], and compare it with state-of-the-art approaches, such as deformable PS RoI pooling [5]. In summary, our contributions are three-fold:

- We propose a supervised rotated RoI learner, which is a learnable module that can transform Horizontal RoIs to RRoIs. This design can not only effectively alleviate the misalignment between RoIs and objects, but also avoid a large number of anchors designed for oriented object detection.
- We design a Rotated Position Sensitive RoI Alignment module for spatially invariant feature extraction, which can effectively boost the object classification and location regression. The module is a crucial design when using the light-head RoI-wise operation, which grantees efficiency and low complexity.
- We achieve state-of-the-art performance on several public large-scale datasets for oriented object detection in aerial images. Experiments also show that the proposed RoI Transformer can be easily embedded in different backbones with significant detection performance improvements.

2. Related Work

2.1. Oriented Bounding Box Regression

Detecting oriented objects is an extension of general horizontal object detection. The task is to locate and classify an object with orientation information, which is mainly tackled with methods based on region proposals. The HRoI based methods [15, 37] usually use a normal RoI Warping to extract feature from a HRoI, and regress position offsets relative to the ground truths. The HRoI based method exists a problem of misalignment between region feature and instance. The RRoI based methods [30, 26] usually use a Rotated RoI Warping to extract feature from a RRoI, and regress position offsets relative to the RRoI, which can avoid the problem of misalignment in a certain. However, the RRoI based method involves generating a lot of rotated proposals. The [26] adopted the method in [27] for rotated proposals. The SRBBS [27] is hard to be embedded in the neural network, which would cost extra time for rotated proposal generation. The [30, 43, 41, 1] used a design of rotated anchor in RPN [32]. However, the design is still time-consuming due to the dramatic increase in the number of anchors ($num\_scales \times num\_aspect\_ratios \times num\_angles$). For example, $3 \times 5 \times 6 = 90$ anchors at a location. A large amount of anchors increases the computation of parameters in the network, while also degrades the efficiency of matching between proposals and ground truths at the same time. Furthermore, directly matching between oriented bounding boxes (OBBs) is harder than that between horizontal bounding boxes(HBBs) because of the existence of plenty of redundant rotated anchors. Therefore, in the design of rotated anchors, both the [30, 24] used a relaxed matching strategy. There are some anchors that do not achieve an IoU above 0.5 with any ground truth, but they are assigned to be True Positive samples, which can still cause the problem of misalignment. In this work, we still use horizontal anchors. The difference is that when the HRoIs are generated, we transform them into RRoIs by a light fully connected layer. Based on this strategy, it is unnecessary to increase the number of anchors. And a lot of precisely RRoIs can be acquired, which will boost the matching process. So we directly use the IoU between OBBs as a matching criterion, which can effectively avoid the problem of misalignment.

2.2. Spatial-invariant Feature Extraction

CNN has the property of translation-invariant while showing poor performance on rotation and scale variations. For image feature extraction, the Spatial Transformer [14] and deformable convolution [5] are proposed to model arbitrary deformation. They are learned from the target tasks without extra supervision. For region feature extraction, the deformable RoI pooling [5] is proposed, which is achieved by offset learning for sampling grid of RoI pooling. It can better model the deformation at instance level compared to regular RoI warping [8, 10, 4]. The STN and deformable modules are widely used for recognition in the field of scene text and aerial images [40, 33, 19, 34, 39]. As for object detection in aerial images, there are more rotation and scale variations, but hardly nonrigid deformation. Therefore, our RoI Transformer only models the rigid spatial transformation, which is learned in the format of $(d_x, d_y, d_w, d_h, d_\theta)$. However, different from deformable RoI pooling, our RoI Transformer learns the offset with the supervision of ground truth. And the RRoIs can also be used for further rotated bounding box regression, which can also contribute to the object localization performance.

2.3. Light RoI-wise Operations

The roi-wise operation is the bottleneck of efficiency on two-stage algorithms because the computations are not shared. The Light-head R-CNN [17] is proposed to address this problem by using a larger separable convolution to get a thin feature. It also employs the PS RoI pooling [4] to reduce the dimensionality of feature maps further. A single fully connected layer is applied on the pooled features with the dimensionality of 10, which can significantly improve the speed of two-stage algorithms. In aerial images, there exist scenes where the number of instances is large. For example, there may be over 800 instances on a single 1024×1024 image. Our approach is similar to Deformable RoI pooling [5] where the RoI-wise operations are conducted twice. The light-head design is also employed for efficiency guarantee.

3. RoI Transformer

In this section, we present details of our proposed *RoI Transformer*, which contains two parts, *RRoI Learner* and *RRoI Warping*. The RRoI Learner is a PS RoI Align followed by a fully connected layer with the dimension of 5, which regress the offsets of Rotated Ground Truths (RGTs) relative to HRoIs. The RRoI Warping warp the rotated region features to maintain the rotation invariance. Both of these two layers are differentiable for the end-to-end training. The architecture is shown in Fig.2.

3.1. RRoI Learner

The RRoI learner aims at learning rotated RoIs (RRoIs) from the feature map of horizontal RoIs (HRoIs). Suppose we have got n HRoIs denoted by $\{\mathcal{H}_i\}$ with the format of (x, y, w, h) for predicted 2D locations, width and height of a HRoI, the corresponding feature maps can be denoted as $\{\mathcal{F}_i\}$. Since every HRoI is the external rectangle of a RRoI in ideal scenarios, we are trying to infer the geometry of RRoIs from every feature map $\mathcal{F}_i$ by using the fully connected layers. We first give the regression targets of offsets relative to general RRoIs as

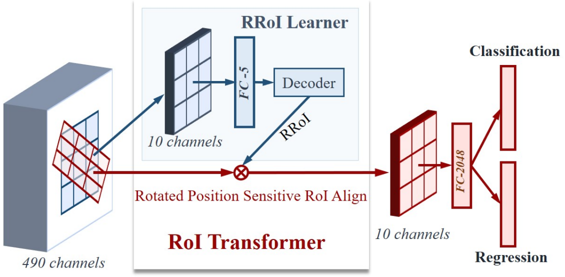


Figure 2. **The architecture of RoI Transformer.** For each HRoI, it is passed to a RRoI learner. The RRoI learner in our network is a PS RoI Align followed by a fully connected layer with the dimension of 5 which regresses the offsets of Rotated Ground Truths (RGTs) relative to HRoI. The Box decoder is at the end of RRoI Learner, which takes the HRoI and the offsets as input and outputs the decoded RRoIs. Then the feature map and the RRoI are passed to the RRoI warping for geometry robust feature extraction. The combination of RRoI Learner and RRoI warping form a RoI Transformer. The geometry robust pooled feature from the RoI Transformer is then used for classification and RRoI regression.

$$
\begin{aligned}
t_x^* &= \tfrac{1}{w_r}\big((x^* - x_r)\cos\theta_r + (y^* - y_r)\sin\theta_r\big),\\
t_y^* &= \tfrac{1}{h_r}\big((y^* - y_r)\cos\theta_r - (x^* - x_r)\sin\theta_r\big),\\
t_w^* &= \log\tfrac{w^*}{w_r}, \quad t_h^* = \log\tfrac{h^*}{h_r},\\
t_\theta^* &= \tfrac{1}{2\pi}\big((\theta^* - \theta_r) \mod 2\pi\big),
\end{aligned} \tag{1}
$$

where $(x_r, y_r, w_r, h_r, \theta_r)$ is a stacked vector for representing location, width, height and orientation of a RRoI and $(x^*, y^*, w^*, h^*, \theta^*)$ is the ground truth parameters of an oriented bounding box (OBB). The $\mod$ is used to adjust the angle offset target t_θ^* in $[0, 2\pi)$ for the convenience of computation. Indeed, the target for regression offsets relative to HRoI is a special case of Eq. (1) if $\theta^* = \frac{3\pi}{2}$. The general relative offsets are illustrated in Fig. 3 as an example. To derive the Eq. (1), you need to translate the coordinates of OBB from **global coordinate sysetm** to **local coordinate system** (for example, $x_1O_1y_1$). Mathematically, the fully connected layer output a vector $(t_x, t_y, t_w, t_h, t_\theta)$ for every feature map $\mathcal{F}_i$ by

$$
\boldsymbol{t} = \mathcal{G}(\mathcal{F}; \Theta), \tag{2}
$$

where $\mathcal{G}$ represents the fully connected layer and Θ is the weight parameters of $\mathcal{G}$ and $\mathcal{F}$ is the feature map for every HRoI.

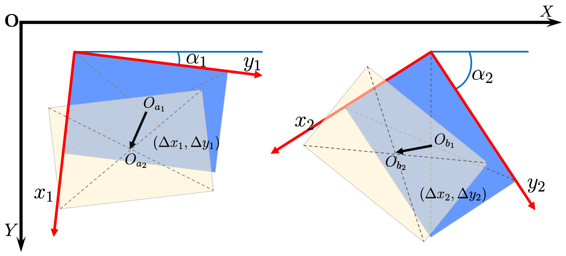


Figure 3. **An example explaining the relative offset.** There are three coordinate systems. The XOY is the **global coordinate system** bound to the image. The $x_1O_1y_1$ and $x_2O_2y_2$ are **local coordinate systems** bound to two RRoIs (blue rectangle) respectively. The $(\Delta x, \Delta y)$ represents the offset between the center of RRoI and RGT. The yellow rectangle represents the Rotated Ground Truth (RGT). The right two rectangles are obtained from the left two rectangles by translation and rotation while keeping the relative position unchanged. The $(\Delta x_1, \Delta y_1)$ is not equal to $(\Delta x_2, \Delta y_2)$ if we observe in the coordinates XOY. They are the same if we observe $(\Delta x_1, \Delta y_1)$ in $x_1O_1y_1$ and $(\Delta x_2, \Delta y_2)$ in $(x_2O_2y_2)$. The α_1 and α_2 denote the angles for two RRoIs respectively.

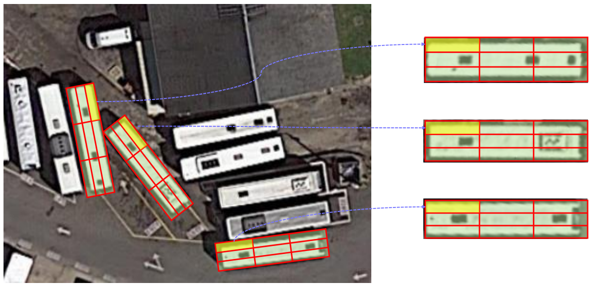


Figure 4. **Rotated RoI warping** The shape of the warped feature is a horizontal rectangle (we use 3 × 3 for example here.) The sampling grid for RoI warping is determined by the RRoI (x_r, y_r, w, h, θ). We employ the image instead of the feature map for a better explanation. After the RRoI warping, the extracted features are geometry robust. (The orientations of all the vehicles are the same).

During the training, we have to match the input HRoIs and the ground truth of oriented bounding boxes (OBBs). For efficiency, the matching process is made between the HRoI and axis-aligned bounding boxes over original ground truth. Once an HRoI is matched with a ground truth of OBB, we set the t^* directly by definition in Eq. (1). We use the Smooth L1 loss [9] function for the regression loss. For the predicted $\boldsymbol{t}$ in every forward pass, we decode it from offsets to the parameters of RRoI. That is to say, our proposed RRoI learner can learn the parameters of RRoI from the HRoI feature map $\mathcal{F}$.

3.2. RRoI Warping

Once we have the parameters of RRoI, we can extract the rotation-invariant deep features for Oriented Object Detection by RRoI Warping. Here, we propose the module of Rotated Position Sensitive (RPS) RoI Align as the concrete RRoI Warping, since our baseline (more details in Sec. 2.3) is Light-Head R-CNN [17]. Given the input feature map $\mathcal{D}$ with shape of $(H, W, K \times K \times C)$ and a RRoI $(x_r, y_r, w_r, h_r, \theta_r)$, where (x_r, y_r) denotes the center of the RRoI and (w_r, h_r) denotes the width and height of the RRoI. The θ_r gives the orientation of the RRoI. The RPS RoI Align divides the Rotated RoI into $K \times K$ bins and outputs a feature map $\mathcal{Y}$ with the shape of (K, K, C). For the bin with index (i, j) $(0 \leq i, j < K)$ of the output channel c$(0 \leq c < C)$, we have

$$\mathcal{Y}_c(i,j) = \sum_{(x,y)\in bin(i,j)} D_{i,j,c}(\mathcal{T}_\theta(x,y))/n, \tag{3}$$

where the $D_{i,j,c}$ is a feature map out of the $K \times K \times C$ feature maps. The channel mapping from the input to output is the same as the original Position Sensitive RoI pooling [4]. The $n \times n$ is the number of sampling locations in the bin. The $bin_{(i,j)}$ denote the coordinates set $\{i\frac{w_r}{k}+(s_x+0.5)\frac{w_r}{k\times n}; s_x = 0, 1, ...n-1\}\times\{j\frac{h_r}{k}+(s_y+0.5)\frac{h_r}{k\times n}; s_y = 0, 1, ...n-1\}$. And for each $(x, y) \in bin(i, j)$, it is converted to $(x^{'}, y^{'})$ by $\mathcal{T}_\theta$, where

$$\begin{pmatrix} x^{'} \\ y^{'} \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x - w_r/2 \\ y - h_r/2 \end{pmatrix} + \begin{pmatrix} x_r \\ y_r \end{pmatrix} \tag{4}$$

Typically, Eq. (3) is implemented by bilinear interpolation.

3.3. RoI Transformer for Oriented Object Detection

The combination of RRoI Learner, and RRoI Warping forms a RoI Transformer (RT). It can be used to replace the normal RoI warping operation. The pooled feature from RT is rotation-invariant. Moreover, the RRoIs provide better initialization for later regression because the matched RRoI is closer to the RGT compared to the matched HRoI. As mentioned before, a RRoI is a tuple with 5 elements $(x_r, y_r, w_r, h_r, \theta_r)$. In order to eliminate ambiguity, we use h to denote the short side and w the long side of a RRoI. The orientation vertical to h and falls in $[0, \pi]$ is chosen as the final direction of a RRoI. After all these operations, the ambiguity is avoided. Also, the operations are required to reduce rotation variations.

IoU between Polygons When matching between RRoI and RGT, we still use the IoU as the criteria. If a RRoI with any RGT has an IoU over the threshold of 0.5, it is considered to be True Positive (TP). For the calculation of IoU between RRoI and RGT, we use the Eq. (5) as shown below. It has a similar form with the IoU calculation between horizontal bounding boxes. The only difference is that the IoU calculation for RRoIs is performed within polygons. The B_r means the bounding box of a RRoI. The B_{gt} represents the bounding box of a ground truth. The $area$ is a function for calculating the area of an arbitrary polygon.

$$IoU = \frac{area(B_r \cap B_{gt})}{area(B_r \cup B_{gt})} \tag{5}$$

Targets Calculation After RRoI warping, the rotation-invariant feature is obtained. Then we add a 2048 dimension fully connected layer (fc) followed by two sibling fcs for final classification and regression (see in Fig. 2). The classification targets are the same as previous works. However, the regression targets are different. To maintain consistency, the offsets also need to be rotation-invariant. To achieve this goal, we use the relative offsets as explained in Fig. 3. The main idea is to use the coordinate system binding to the RRoI rather than the image for offsets calculation. The Eq. (1) is the derived formulation for relative offsets.

4. Experiments and Analysis

4.1. Datasets

For experiments, we choose two datasets, known as DOTA [37] and HRSC2016 [28], for oriented object detection in aerial images.

- **DOTA [37].** It is the largest dataset for object detection in aerial images with oriented bounding box annotations. It contains 2806 large size images. There 15 categories, including *Baseball diamond (BD), Ground track field (GTF), Small vehicle (SV), Large vehicle (LV), Tennis court (TC), Basketball court (BC), Storage tank (ST), Soccer-ball field (SBF), Roundabout (RA), Swimming pool (SP), and Helicopter (HC)*. The fully annotated DOTA images contain 188, 282 instances. The instances in this data set vary greatly in scale, orientation, and aspect ratio. As shown in [37], the algorithms designed for regular horizontal object detection get modest performance on it. Like PASCAL VOC [7] and COCO [21], the DOTA provides the evaluation server[1].

  We use both the training and validation sets for training, the testing set for testing. We do a limited data augmentation. Specifically, we resize the image at two scales(1.0 and 0.4) for training and (1.0 and 0.5) testing. After image rescaling, we crop a series of 1024×1024 patches from the original images with a stride of 824. For those categories with a small number of samples, we do a rotation augmentation randomly from 4 angles $(0, 90, 180, 270)$ to simply avoid the effect of an imbalance between different categories. With all these processes, we obtain 37373 patches, which are much less than that in the official baseline implements (150, 342 patches) [37]). For testing experiments, the 1024×1024 patches are also employed. None of the other tricks is utilized except the stride for image sampling is set to 512.

- **HRSC2016 [28].** The HRSC2016 [28] is a challenging dataset for ship detection in aerial images. The images are collected from Google Earth. It contains 1061 images and more than 20 categories of ships in various appearances. The image size ranges from 300×300 to 1500×900. The training, validation and test set include 436 images, 181 images, and 444 images, respectively. For data augmentation, we only adopt the horizontal flipping. And the images are resized to $(512, 800)$, where 512 represents the length of the short side and 800 the maximum length of an image.

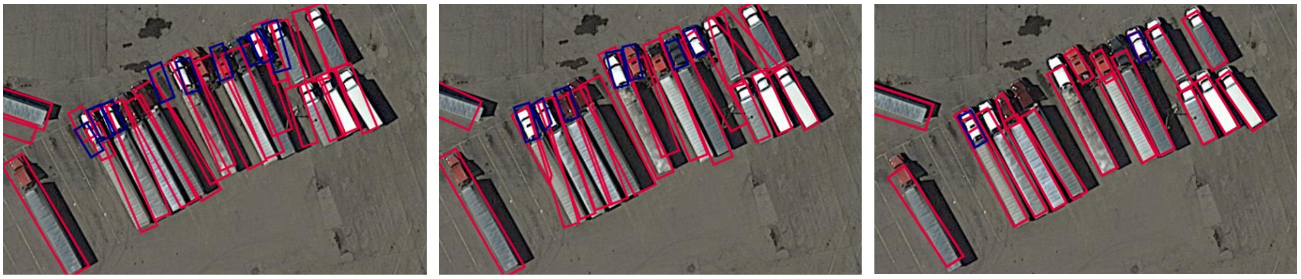
4.2. Implementation details

Baseline Framework. For the experiments, we build the baseline network inspired from Light-Head R-CNN [17] with backbone ResNet101 [11]. Our final detection performance is based on the FPN [22] network, while it is not employed in the ablation experiments for simplicity.

- **Light-Head R-CNN OBB:** We modified the regression of fully-connected layer on the second stage to enable it to predict OBBs, similar to work in DOTA [37]. The only difference is that we replace $((x_i, y_i), i = 1, 2, 3, 4)$ with (x, y, w, h, θ) for the representation of an OBB. Since there is an additional param θ, we do not double the regression loss as the original Light-Head R-CNN [17] does. The hyperparameters of large separable convolutions we set is $k = 15, Cmid = 256, Cout = 490$. And the OHEM [35] is not employed for sampling at the training phase. For RPN, we used 15 anchors same as original Light-Head R-CNN [17]. The batch size of RPN [32] is set to 512. Finally, there are 6000 RoIs from RPN before Non-maximum Suppression (NMS) and 800 RoIs after using NMS. Then 512 RoIs are sampled for the training of R-CNN. The learning rate is set to 0.0005 for the first 14 epochs and then divided by 10 for the last four epochs. For testing, we adopt 6000 RoIs before NMS and 1000 after NMS processing.

- **Light-Head R-CNN OBB with FPN:** The Light-Head R-CNN OBB with FPN uses the FPN [22] as a backbone network. Since no source code was publicly available for Light-Head R-CNN based on FPN, our implementation details could be different. We simply added the large separable convolution on the feature of every level P_2, P_3, P_4, P_5. The hyperparameters of large separable convolution we set is $k = 15, Cmid = 64, Cout = 490$. The batch size of RPN is set to be 512. There are 6000 RoIs from RPN before NMS and 600 RoIs after NMS processing. Then 512 RoIs are sampled for the training of R-CNN. The learning rate is set to 0.005 for the first five epochs and divided by a factor of 10 for the last two epochs.

[1] http://captain.whu.edu.cn/DOTAweb/evaluation.html



Light-Head R-CNN Baseline

deformable PS RoI pooling

RoI Transformer

Figure 5. Visualization of detection on the scene where many densely packed instances exist. We select the predicted bounding boxes with scores above 0.1, and a NMS with threshold 0.1 is applied for duplicate removal.

Table 1. Results of ablation studies. We used the *Light-Head R-CNN OBB* detector as our baseline. The leftmost column represents the optional settings for the RoI Transformer. In the right four experiments, we explored the appropriate setting for RoI Transformer.

	Baseline	RoITransformer with different settings			
Light RRoI Learner?			✓	✓	✓
Context region enlarge?				✓	✓
NMS on RRoIS?		✓	✓	✓	
mAP	58.3	63.17	63.39	66.25	**67.74**

Table 2. Comparisons with the state-of-the-art methods on HRSC2016.

method	CP [26]	BL2 [26]	RC1 [26]	RC2 [26]	R^2PN [43]	RRD [20]	RoI Trans.
mAP	55.7	69.6	75.7	75.7	79.6	84.3	**86.2**

4.3. Comparison with Deformable PS RoI Pooling

In order to validate that the performance is not from additional computation, we compared our method with deformable PS RoI pooling (DPSRP), since both of them are a kind of improved RoI Warping operation to model the geometry variations. For experiments, we use the Light-Head R-CNN OBB as our baseline. The deformable PS RoI pooling and RoI Transformer are used to replace the PS RoI Align in the Light-Head R-CNN respectively.

Complexity. Both RoI Transformer and deformable RoI pooling have a light localisation network, which is a standard pooled feature followed by a fully connected layer. In our RoI Transformer, only 5 parameters$(t_x, t_y, t_w, t_h, t_\theta)$ are learned. The deformable PS RoI pooling learns offsets for each bin, where the number of parameters is $7 \times 7 \times 2$. So our module is designed lighter than deformable PS RoI pooling. As can be seen in Tab. 4, our RoI Transformer model uses almost equal memory (273MB compared to 273.2MB) and runs faster at the inference phase (0.17s compared to 0.206s per image). However, RoI Transformer runs slower than deformable PS RoI pooling on training time (0.475s compared to 0.445s) since there is an extra matching process between the RRoIs and RGTs in training.

Detection Accuracy. The comparison results are shown in Tab. 4. The deformable PS RoI pooling outperforms the Light-Head R-CNN OBB Baseline by 5.6 points. While there is only 1.4 points improvement for R-FCN [4] on Pascal VOC [7] as pointed out in [5]. It shows that the geometry modeling is more important for object detection in aerial images. However, the deformable PS RoI pooling is much lower than our RoI Transformer by 3.85 points. We argue that there are two reasons: 1) Our RoI Transformer can better model the geometry variations in aerial images. 2) The regression targets of deformable PS RoI pooling are still relative to the HRoI rather than using the boundary of the offsets. Our regression targets are relative to the RRoI, which gives a better initialization for regression. We visualize some detection results for detecting densely packed instances in Fig. 5. The results show that our proposed method can precisely locate the instances in scenes with densely packed ones. While the Light-Head R-CNN OBB baseline and the deformable RoI pooling show worse performance on classification and localization of instances. Specifically, the head of truck is misclassified to be the small vehicle (the blue bounding box) as shown in Fig. 5. However, our proposed RoI Transformer has the least number of misclassified instances.

4.4. Ablation Studies

We conduct a serial of ablation experiments on DOTA to find the appropriate settings of our proposed RoI Transformer. We use the Light-Head R-CNN OBB as our baseline. Then gradually change the settings. When applying

Table 3. Comparisons with state-of-the-art detectors on DOTA [37]. The short names for each category can be found in Section 4.1. The dresnet101 in ICN [1] means deformable conv. resnet101. The FR-O indicates the *Faster R-CNN OBB* detector, which is the official baseline provided by DOTA [37]. The RRPN indicates the *Rotation Region Proposal Networks*, which used a design of rotated anchor. The R2CNN means *Rotational Region CNN*, which is a HRoI-based method without using the RRoI warping operation. The RDFPN means the *Rotation Dense Feature Pyramid Netowrks*. It also used a design of Rotated anchors and used a variation of FPN. The work in Yang et al. [42] is an extension of R-DFPN.

method	backbone	W/FPN	test scales	Plane	BD	Bridge	GTF	SV	LV	Ship	TC	BC	ST	SBF	RA	Harbor	SP	HC	mAP
FR-O [37]	resnet101		1	79.42	77.13	17.7	64.05	35.3	38.02	37.16	89.41	69.64	59.28	50.3	52.91	47.89	47.4	46.3	54.13
RRPN [30]	resnet101		1	80.94	65.75	35.34	67.44	59.92	50.91	55.81	90.67	66.92	72.39	55.06	52.23	55.14	53.35	48.22	61.01
R2CNN [15]	resnet101		1	88.52	71.2	31.66	59.3	51.85	56.19	57.25	**90.81**	72.84	67.38	56.69	52.84	53.08	51.94	53.58	60.67
R-DFPN [41]	resnet101	✓	1	80.92	65.82	33.77	58.94	55.77	50.94	54.78	90.33	66.34	68.66	48.73	51.76	55.1	51.32	35.88	57.94
Yang et al. [42]	resnet101	✓	1	81.25	71.41	36.53	67.44	61.16	50.91	56.6	90.67	68.09	72.39	55.06	55.6	62.44	53.35	51.47	62.29
ICN [1]	dresnet101	✓	4	81.36	74.3	**47.7**	70.32	64.89	67.82	69.98	90.76	79.06	78.2	53.64	**62.9**	**67.02**	**64.17**	50.23	68.16
Baseline	resnet101		2	81.06	76.81	27.22	69.75	38.99	39.07	38.3	89.97	75.53	65.74	**63.48**	59.37	48.11	56.86	44.46	58.31
DPSRP	resnet101		2	81.18	77.42	35.48	70.41	56.74	50.42	53.56	89.97	**79.68**	76.48	61.99	59.94	53.34	64.04	47.76	63.89
RoITransformer	resnet101		2	88.53	77.91	37.63	74.08	66.53	62.97	66.57	90.5	79.46	76.75	59.04	56.73	62.54	61.29	**55.56**	67.74
Baseline	resnet101	✓	2	88.02	76.99	36.7	72.54	**70.15**	61.79	75.77	90.14	73.81	**85.04**	56.57	62.63	53.3	59.54	41.91	66.95
RoITransformer	resnet101	✓	2	**88.64**	**78.52**	43.44	**75.92**	68.81	**73.68**	**83.59**	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	**69.56**

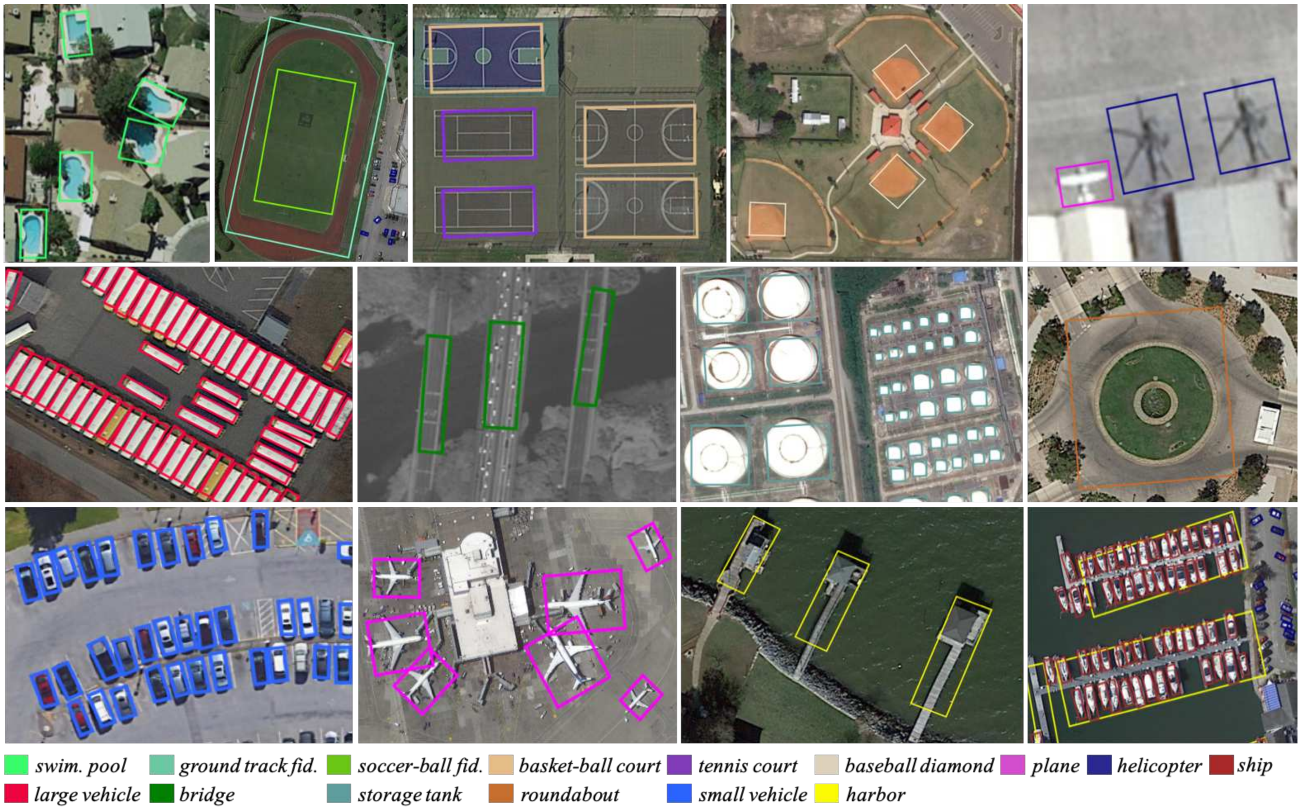


Figure 6. Visualization of results from RoI Transformer in DOTA.

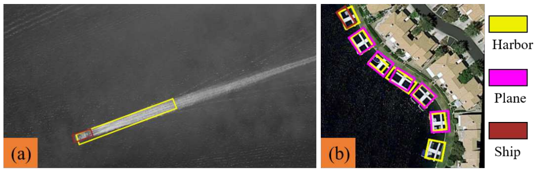


Figure 7. Failure cases. (a) detects the long wake of a ship as a harbor. (b) incorrectly detects harbors as planes.

the RoI Transformer with a simple setting, there is a 4.87 point improvement in mAP. We discuss the other settings in the following.

Table 4. Comparison of our RoI Transformer with deformable PS RoI pooling and Light-Head R-CNN OBB on accuracy, speed and memory. All the speed are tested on images with size of 1024 × 1024 on a single TITAN X (Pascal). The time of post-process (*i.e.* NMS) was not included. The LR-O, DPSRP and RT denote the Light-Head R-CNN OBB, deformable Position Sensitive RoI pooling and RoI Transformer, respectively.

method	mAP	train speed	test speed	param
LR-O	58.3	**0.403 s**	**0.141s**	**273MB**
DPSRP	63.89	0.445s	0.206s	273.2MB
RT	**67.74**	0.475s	0.17s	**273MB**

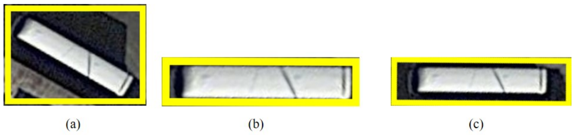


Figure 8. Comparison of three kinds of region for feature extraction. (a) The Horizontal Region. (b) The rectified Region after RRoI Warping. (c) The rectified Region with appropriate context after RRoI warping.

Light RRoI Learner. In order to guarantee the efficiency, we directly apply a fully connected layer with an output dimension of 5 on the pooled features from the HRoI warping. As a comparison, we also tried more fully connected layers for the RRoI learner, as shown at the first and second columns in Tab. 1. We find there is little drop (0.22 point) on mAP when we add one more fully connected layer with an output dimension of 2048 for the RRoI learner. The reason may be that the additional fully connected layer with higher dimensionality requires a longer time for convergence.

Contextual RRoI. As pointed out in [13, 30], appropriate enlargement of the RoI will promote the performance. A horizontal RoI contains much background while a precise RRoI hardly contains redundant background as explained in the Fig. 8. Complete abandon of contextual information will make it difficult to classify and locate the instance even for the human. Therefore, it is necessary to enlarge the region of the feature with an appropriate degree. Here, we enlarge the long side of RRoI by a factor of 1.2 and the short side by 1.4. The enlargement of RRoI improves AP by 2.86 points, as shown in Tab. 1

NMS on RRoIs. Since the obtained RoIs are rotated, there is flexibility for us to decide whether to conduct another NMS on the RRoIs transformed from the HRoIs. This comparison is shown in the last two columns of Tab. 1. We find there is 1.5 points improvement in mAP if we remove the NMS. This is reasonable because there are more RoIs without additional NMS, which could increase the recall.

4.5. Comparisons with the State-of-the-art

We compared the performance of our proposed RoI Transformer with the state-of-the-art algorithms on two datasets DOTA [37] and HRSC2016 [28]. The settings are described in Sec. 4.2, and we just replace the Position Sensitive RoI Align with our proposed RoI Transformer. Our baseline and RoI Transformer results are obtained without using ohem [35] at the training phase.

Results on DOTA. Note the RRPN [30] and R2CNN [15] are originally used for text scene detection. The results are a re-implemented version for DOTA by a third-party[2]. As can be seen in Tab. 3, RoI Transformer without FPN achieved the mAP of 67.74 for DOTA, it outperforms the previous the state-of-the-art without FPN (61.01) by 6.71 points. And there is only 0.42 point lower than the previous state-of-the-art with FPN (68.16). When we add RoI Transformer on the stronger baseline of Light-Head OBB FPN, it still has improvement by 2.6 points in mAP reaching the peak at 69.56. This indicates that the proposed RoI Transformer is valid for different backbones. Besides, there is a significant improvement in densely packed small instances. (e.g., the small vehicles, large vehicles, and ships). For example, the detection performance for the ship category gains an improvement of 13.61 points compared to the previous best result (69.98) achieved by ICN [1]. We give some qualitative results of RoI Transformer on DOTA in Fig. 6. The failure cases are given in Fig. 7. From the failure cases, we can see the model do not learn the context, which is what we do not consider yet.

Results on HRSC2016. The HRSC2016 contains a lot of thin and long ship instances with arbitrary orientation. We use 4 scales $\{64^2, 128^2, 256^2, 512^2\}$ and 5 aspect ratios $\{1/3, 1/2, 1, 2, 3\}$, yielding $k = 20$ anchors for RPN initialization. This is because there is more aspect ratio variations in HRSC, but relatively fewer scale changes. The other settings are the same as those in 4.2. We conduct the experiments without FPN which still achieves the best performance on mAP. Specifically, based on our proposed method, the mAP can reach 86.16, 1.86 higher than that of RRD [20]. The RRD adopt SSD [25] as architecture for oriented object detection. Note it utilizes multi-layers for feature extraction and 13 different aspect ratios of default boxes$\{1, 2, 3, 5, 7, 9, 15, 1/2, 1/3, 1/5, 1/7, 1/9, 1/15\}$. While our proposed framework simply employs the final output features with only five aspect ratios of boxes.

5. Conclusion

In this paper, we proposed a module called RoI Transformer to model the geometry transformation, which can effectively avoid the problem of misalignment between region feature and objects. This design brings significant improvements for oriented object detection on the challenging DOTA and HRSC with a negligible computation cost increase. Furthermore, the comprehensive comparisons with deformable RoI pooling verified that our model is more reasonable when oriented bounding box annotations are available.

Ackowledgement

This work is supported by NSFC-projects under contracts No.61771350 and 61842102. Nan Xue is supported by China Scholarship Council.

References

[1] Seyed Majid Azimi, Eleonora Vig, Reza Bahmanyar, Marco Körner, and Peter Reinartz. Towards multi-class ob-

[2] https://github.com/DetectionTeamUCAS/RRPN_Faster-RCNN_Tensorflow

ject detection in unconstrained remote sensing imagery. *arXiv:1807.02700*, 2018. 4321, 4322, 4327, 4328

[2] Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote Sens.*, 54(12):7405–7415, 2016. 4321

[3] Gong Cheng, Peicheng Zhou, and Junwei Han. Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection. In *CVPR*, pages 2884–2893, 2016. 4321

[4] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016. 4322, 4323, 4324, 4326

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *CoRR, abs/1703.06211*, 1(2):3, 2017. 4322, 4323, 4326

[6] Zhipeng Deng, Hao Sun, Shilin Zhou, Juanping Zhao, and Huanxin Zou. Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks. *J-STARS*, 10(8):3652–3664, 2017. 4321

[7] Mark Everingham, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 4325, 4326

[8] Ross Girshick. Fast r-cnn. In *CVPR*, pages 1440–1448, 2015. 4321, 4322, 4323

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 4321, 4322, 4324

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988. IEEE, 2017. 4322, 4323

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, June 2016. 4325

[12] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. Drone-based object counting by spatially regularized regional proposal network. In *ICCV*, volume 1, 2017. 4321

[13] Peiyun Hu and Deva Ramanan. Finding tiny faces. In *CVPR*, pages 1522–1530. IEEE, 2017. 4328

[14] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 4322, 4323

[15] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: rotational region cnn for orientation robust scene text detection. *arXiv:1706.09579*, 2017. 4322, 4327, 4328

[16] Xiaobin Li and Shengjin Wang. Object detection using convolutional neural networks in a coarse-to-fine manner. *IEEE Geosci. Remote Sensing Lett.*, 14(11):2037–2041, 2017. 4321

[17] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head r-cnn: In defense of two-stage object detector. *arXiv:1711.07264*, 2017. 4323, 4324, 4325

[18] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *CoRR*, abs/1801.02765, 2018. 4321

[19] Minghui Liao, Jian Zhang, Zhaoyi Wan, Fengming Xie, Jiajun Liang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Scene text recognition from two-dimensional perspective. *arXiv:1809.06508*, 2018. 4323

[20] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *CVPR*, pages 5909–5918, 2018. 4326, 4328

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014. 4325

[22] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 3, 2017. 4325

[23] Kang Liu and Gellért Máttyus. Fast multiclass vehicle detection on aerial images. *IEEE Geosci. Remote Sensing Lett.*, 12(9):1938–1942, 2015. 4321

[24] Lei Liu, Zongxu Pan, and Bin Lei. Learning a rotation invariant detector with rotatable bounding box. *arXiv:1711.09405*, 2017. 4322

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016. 4328

[26] Zikun Liu, Jingao Hu, Lubin Weng, and Yiping Yang. Rotated region based cnn for ship detection. In *ICIP*, pages 900–904. IEEE, 2017. 4321, 4322, 4326

[27] Zikun Liu, Hongzhen Wang, Lubin Weng, and Yiping Yang. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geosci. Remote Sensing Lett.*, 13(8):1074–1078, 2016. 4321, 4322

[28] Zikun Liu, Hongzhen Wang, Lubin Weng, and Yiping Yang. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geosci. Remote Sensing Lett.*, 13(8):1074–1078, 2016. 4321, 4322, 4325, 4328

[29] Yang Long, Yiping Gong, Zhifeng Xiao, and Qing Liu. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.*, 55(5):2486–2498, 2017. 4321

[30] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *TMM*, 2018. 4321, 4322, 4327, 4328

[31] Sebastien Razakarivony and Frederic Jurie. Vehicle detection in aerial imagery: A small target detection benchmark. *J Vis. Commun. Image R.*, 34:187–203, 2016. 4321

[32] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(6):1137–1149, 2017. 4321, 4322, 4325

[33] Yun Ren, Changren Zhu, and Shunping Xiao. Deformable faster r-cnn with aggregating multi-layer features for partially occluded object detection in optical remote sensing images. *Remote Sensing*, 10(9):1470, 2018. 4323

[34] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *CVPR*, pages 4168–4176, 2016. 4323

[35] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 4325, 4328

[36] Guoli Wang, Xinchao Wang, Bin Fan, and Chunhong Pan. Feature extraction by rotation-invariant matrix representation for object detection in aerial image. *IEEE Geosci.Remote Sensing Lett.*, 2017. 4321

[37] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proc. CVPR*, 2018. 4321, 4322, 4325, 4327, 4328

[38] Zhifeng Xiao, Yiping Gong, Yang Long, Deren Li, Xiaoying Wang, and Hua Liu. Airport detection based on a multiscale fusion feature for optical remote sensing images. *IEEE Geosci. Remote Sensing Lett.*, 14(9):1469–1473, 2017. 4321

[39] Zhaozhuo Xu, Xin Xu, Lei Wang, Rui Yang, and Fangling Pu. Deformable convnet with aspect ratio constrained nms for object detection in remote sensing imagery. *Remote Sensing*, 9(12):1312, 2017. 4323

[40] Qiangpeng Yang, Mengli Cheng, Wenmeng Zhou, Yan Chen, Minghui Qiu, and Wei Lin. Inceptext: A new inception-text module with deformable psroi pooling for multi-oriented scene text detection. *arXiv:1805.01167*, 2018. 4323

[41] Xue Yang, Hao Sun, Kun Fu, Jirui Yang, Xian Sun, Menglong Yan, and Zhi Guo. Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, 10(1):132, 2018. 4322, 4327

[42] Xue Yang, Hao Sun, Xian Sun, Menglong Yan, Zhi Guo, and Kun Fu. Position detection and direction prediction for arbitrary-oriented ships via multiscale rotation region convolutional neural network. *arXiv:1806.04828*, 2018. 4327

[43] Zenghui Zhang, Weiwei Guo, Shengnan Zhu, and Wenxian Yu. Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks. *IEEE Geosci. Remote Sensing Lett.*, (99):1–5, 2018. 4322, 4326