# Learning Object-Wise Semantic Representation for Detection in Remote Sensing Imagery

Chengzheng Li[1], Chunyan Xu[1], Zhen Cui[1]*, Dan Wang[2], Zequn Jie[3], Tong Zhang[1], Jian Yang[1]

[1]Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology
[2]Institute of Spacecraft System Engineering (ISSE), China Academy of Space Technology (CAST)
[3]Tencent AI Lab

## Abstract

*With the upgrade of remote sensing technology, object detection in remote sensing imagery becomes a critical but also challenging problem in the field of computer vision. To deal with highly complex background and extreme variation of object scales, we propose to learn a novel object-wise semantic representation for boosting the performance of detection task in remote sensing imagery. An enhanced feature pyramid network is first designed to better extract hierarchical discriminative visual features. To suppress background clutter as well as better estimate proposals, next we specifically introduce a semantic segmentation module to guide horizontal proposals detection. Finally, a ROI module which can fuses multiple-level features is proposed to further promote object detection performance for both horizontal and rotate bounding boxes. With the proposed approach, we achieve* ***79.5%*** *mAP and* ***76.6%*** *mAP in horizontal bounding boxes (HBB) and oriented bounding boxes (OBB) tasks of DOTA-v1.5 dataset, which takes* ***the first and second place*** *in the DOAI2019 challenge[1], respectively.*

## 1. Introduction

In the past few years, the object detection task in remote sensing (RS) imagery, which refers to detecting semantic objects of certain categories, has enabled various high-level applications, such as meteorological observation, urban road construction, surface migration analysis, natural disaster management, etc. However, since the otherness in aspects of posture and altitude when remote sensing satellites are recording the images, objects in aerial images often own some specific characteristics of their own, such as huge variation in scales, arbitrariness of arrangement orientation, high complexity of background information. In view of the above complex situations, object detection task has recently emerged as a fundamental yet challenging problem in the field of remote sensing.

Depend on the powerful feature learning ability, convolutional neural networks (CNNs) have achieved great success in multiple visual tasks among recent years, such as classification [1, 14, 24], segmentation [16, 13, 12], tracking [2, 3, 18], as well as detection [23, 6, 11, 28, 26]. In terms of the detection task, Girshick *et al*. utilize a CNN-based two-stage network structure R-CNN [20] to obtain satisfactory detection results. Following R-CNN, several region-based detectors such as Fast R-CNN [19], Faster R-CNN [23], R-FCN [6] are proposed to further improve the efficiency and performance of detectors. Recently, Lin *et al*. propose Feature Pyramid Networks (FPN) [26] to fuse features from multiple stages so as to improve detection results of multi-scale objects. RetinaNet [27] is subsequently present to deal with the class imbalance of samples during training. Cai *et al*. take the idea of cascade and introduce a multi-stage detector named Cascade R-CNN [35] to produce more accurate bounding boxes. Different from these two-stage detectors, other regression-based methods (*e.g*., SSD [28], YOLO [11, 9, 10]) take detection as a regression problem and predict objects' bounding boxes directly just through a single CNN structure. More recently, some algorithms spring up to complete detection tasks in a keypoint manner, *e.g*., Law *et al*. propose a novel and effective approach CornerNet [5], where detects an object bounding box as the top-left corner and bottom-right corner using a single network. Zhou *et al*. come up with CenterNet [32]

*Corresponding author: Zhen Cui (zhen.cui@njust.edu.cn).
[1]https://captain-whu.github.io/DOAI2019/results.html

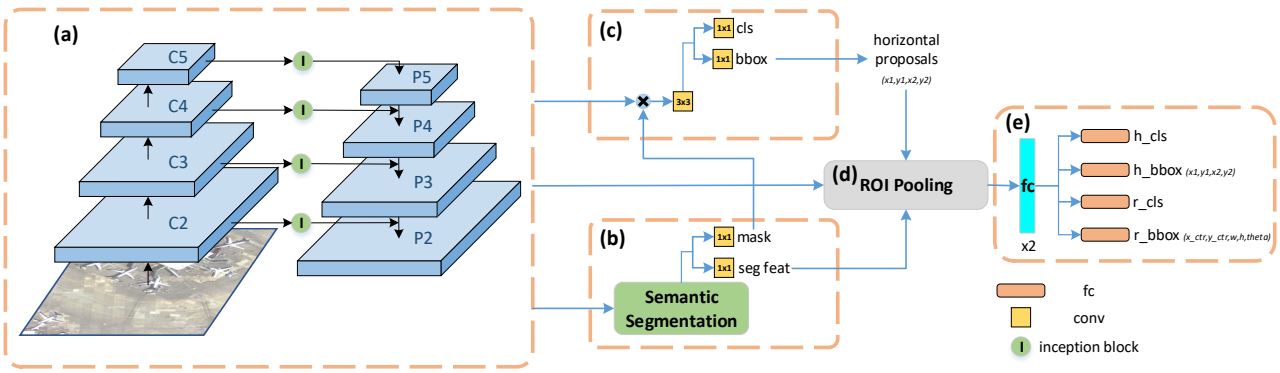


Figure 1. The illustration of our proposed object detection framework. The enhanced FPN (a) is designed to increase the feature discriminability of feature pyramid network by incorporating a sub inception block. Next, we estimate the box-wise mask and learn the semantic feature of the whole image by a semantic segmentation module (b). The produced mask is used to spatially weight the feature of FPN for the prediction of horizontal proposals (c) following the spirit of RPN. To further regress objects accurately, including labels and horizontal/rotate locations (e), we fuse multi-branch features by using ROI Pooling (d), including FPN features, semantic feature and original image. The entire network can be trained and test in an end-to-end way.

to regress size, 3D location, orientation and even pose only based on the center points.

Although substantial results have been achieved by these detection algorithms mentioned above in the natural scene [25, 17], it still remains some challenges when applying these methods to complete detection task in RS imagery directly, due to the more complex background and extreme variation of scales and postures. To solve these problems, researchers attempt to resort existing successful CNN-based frameworks for feature extraction and further designing new effective architectures [30, 21, 31, 7, 29] for detecting objects in RS imagery.

In the general object detection algorithms, some post-processing methods may be adopted to improve detection precision such as non-maximum suppression (NMS) before outputting the final locations of objects. During NMS processing, bounding boxes whose Intersection-over-Union (IoU) are higher than threshold will be removed. But, it is not suitable for detection task in RS imagery, since these dense rotate objects (*e.g.*, vehicles in the parking lots) will have very high IoU if taking the conventional NMS, and then are going to be over-suppressed, so many objects which could have precise localization will be discarded. While, another alternative way is to detect skew bounding boxes which including one extra rotation angle based on horizontal boxes for RS imagery, using rotate NMS [8] as the post-processing method. Inspired by arbitrary-oriented text detection model RRPN [8], many methods [30, 36, 21] adopt rotate region proposals via skew anchors to better match rotate ground-truth (GT) bounding boxes of RS images. Although these skew anchors can obtain a good coverage with rotate objects, the computational burden will greatly increase since each pixel may generate dozens or even hundreds proposals. Another alternative solution is to regress the coordinates of rotate bounding boxes from coarse horizontal region proposals, like $R^2$CNN [34], which can not only make use of the context information in horizontal regions but also reduce model parameters to a certain extent.

Generally speaking, scales of objects in aerial images are quite inconsistent, for example, scale difference between the vehicle and playground may be dozens or even hundreds of times, so we can't directly do detection task in single level features. A common practice is to utilize FPN [26] to extract multi-scale convolutional features. For example, Azimi *et al*. propose a joint image cascade and feature pyramid network (ICN) [21] to fuse multi-scale semantic features from multi-images. Yang *et al*. extend FPN [26] with dense connections layer by layer to construct dense feature pyramid network (R-DFPN) [30]. Although they improve the performances of results to some extent, pyramidal features may be influenced by noise since the complex background of RS images. Recently, many works have proved that object detection and segmentation are two related tasks, and joint training of these two tasks is effective for both subtasks. For example, Mask R-CNN [13] extends Faster R-CNN [23] by adding a parallel segmentation branch for predicting an object mask. MaskLab [15] predicts instance masks by combining semantic and direction outputs. HTC [12] makes use of the mask information flow and spatial contexts feature to improve the detection and segmentation prediction. All these methods utilize object-wise bounding boxes and mask annotations [25] to improve the learning ability of network. However, in most cases, there are no precise mask annotations of objects in RS images. Which is fortunate, in RS images, since the particular

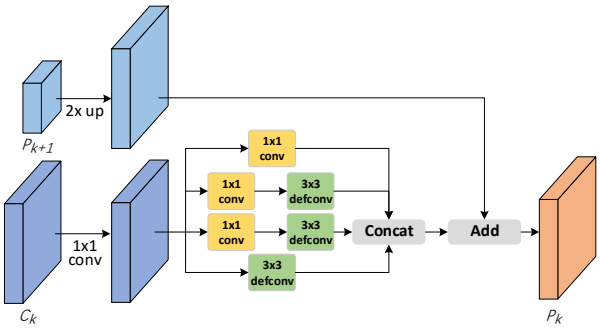


Figure 2. The enhanced feature pyramid network (eFPN).

characteristics of bird's eye view and rotate bounding boxes, there is no or little occlusion between objects. Therefore, box-wise segmentation can also be used to assist detection task in RS imagery.

In this work, we propose a semantic segmentation guided objection detection framework to boost the performance of detection task in remote sensing images. First, we design an enhanced feature pyramid network to better extract hierarchical discriminative visual features. Second, to suppress background information and meantime better estimate proposals, we introduce a semantic segmentation module to guide horizontal proposals generation. The segmentation module produces the object mask and semantic feature for next horizontal proposal estimation and final object detection respectively. In the end, we design a multiple-level fusion based ROI module to predict object labels and corresponding bounding boxes. Based on the proposed framework above, we achieve the performance of **79.5%** mAP and **76.6%** mAP in horizontal bounding boxes (HBB) and oriented bounding boxes (OBB) tasks of DOTA-v1.5 [4] dataset, which takes **the first and second place** in the ODAI2019 challenge, respectively.

## 2. The Proposed Method

### 2.1. Overview

Fig. 1 shows the proposed framework based on object-wise semantic representation. Given a remote sensing image as input, we design an enhanced FPN to better learn these discriminative features of objects with different appearance variations. The detail is introduced in Section 2.2. Next, we introduce the semantic segmentation module (Section 2.3) to obtain the object semantic information, which contains the box-wise mask and semantic feature. The mask is used to guide the generation of horizontal proposals and the semantic feature is used to be fused in ROI pooling for more accurate estimation of bounding boxes. Finally, we design the multiple-level fusion based ROI module (Section 2.4) to learn more distinguishing features of objects. The loss function is introduced in Section 2.5.

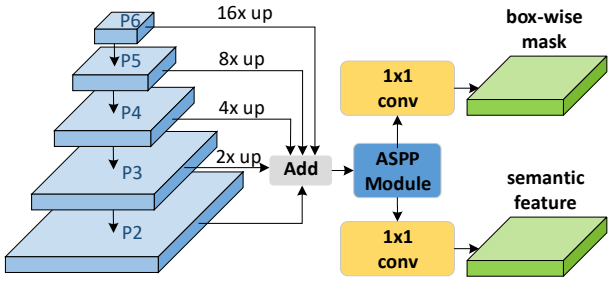


Figure 3. The semantic segmentation module.

### 2.2. Enhanced FPN

Since its simplicity and effectiveness, FPN [26] has became a common approach to fuse multi-scale features. However, objects in RS images often own irregular shape or scale, which is difficult to get robust representations. In order to obtain multi-scale features of different receptive fields, we design an enhanced FPN (eFPN for short) by introducing a sub inception block into the FPN lateral connections to promote the representation ability.

As described in Fig. 2, the encoded and decoded features are integrated into the enhanced feature by one inception block. Concretely, we take the conventional FPN [26] as the infrastructure as shown in Fig. 1. Given the encoded feature map $C_k$ and the decoded map $P_{k+1}$, we take a sub inception block to enhance them and then combine into the new feature map $P_k$ as shown in Fig. 2. The feature map $C_k$ is first transformed into the 256-channel feature by using the $1 \times 1$ convolutional layer as in FPN [26]. Next we take a multi-branch inception process, which transforms the 256-channel feature into four 64-channel features at each branch network. In the sub inception block, we adopt the deformable convolution [33], since we think the deformable convolution can deal well with the geometric particularity of objects in RS images. Also, we add one shortcut in our inception block to reserve the original feature maps. The concrete operations of the multi-branch block can be observed in Fig. 2. After passing four-branch network, the produced features are concatenated into 256-channel feature. Finally, we summarize the concatenate feature from encode layer and the upsampled feature from the decode layer to construct the final feature $P_k$.

### 2.3. Semantic Segmentation-Guided RPN

Some excellent works integrate detection and segmentation tasks into a single network (*e.g.*, Mask R-CNN[13], MaskLab [15], HTC [12]), and considerable results have been achieved on both tasks. Inspired by these works, we introduce the segmentation idea into RPN to build a semantic segmentation-guided RPN(sRPN for short), so that those background clutter can be suppressed as much as possible. Since there are no precise mask annotations of objects in

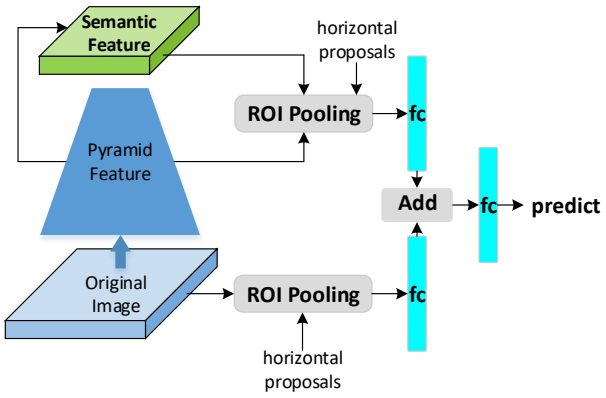


Figure 4. The multiple-level fusion based ROI module (fROI).

RS images, so we just use the masks generated from rotate bounding boxes as substitutes.

The semantic segmentation module is shown in Fig. 3. The multi-level FPN features are first normalized into the same spatial size (*i.e.*, the spatial size of the $P_2$ level) by taking the simple upsample and downsample operations, and then are summarized into a new feature. In [12], the final output level is $P_3$ whose stride is 8, but we find 8 is too big so as to many objects which are too small (like small vehicles) will not be detected, so we choose the $P_2$ level with stride 4, experiments shows that it is a appropriate trade-off between computation and performance. The ensemble feature actually covers the information of different layer feature of objects, and thus can benefit the prediction of region proposals. Next, we take atrous spatial pyramid pooling (ASPP) [16] to generate the semantic representation, which can better encode global context and well boost the performance. For ASPP, we follow the original paper setting [16], which take four parallel atrous convolutions, $1 \times 1$, $3 \times 3$ with rate = 6, $3 \times 3$ with rate = 12 and $3 \times 3$ with rate = 18, we find the original setting has already met our requirements. Feature from ASPP module is used to generate the box-wise mask and the semantic feature by employing the two separate $1 \times 1$ convolution operations with the channel 1 and 256 correspondingly, the sigmoid activation is attached to the first convolutional layer. For supervision, the mask output is upsampled to the same scale as the GT mask.

In the next steps, we downsample the learnt mask to the $P_3 \sim P_6$ level ($P_6$ is produced as in FPN [26]) and multiply them to corresponding RPN head features as shown in Fig. 1, guiding RPN to estimate horizontal region proposals. The semantic feature is sent into the next stage of ROI pooling for more accurate box regression.

### 2.4. Multiple-level Fusion based ROI Module

Previous work [19, 23, 13, 26] has adopted the ROI pooling (align) [19, 13] operation to produce the proposal-related features with the same output size. Most works [23, 26] only utilize a single level feature map, but due to the extreme variation of object scales in RS images, single level feature map is not enough to satisfy the task. In PANet [22], adaptive feature pooling is proposed to fuse pooled features from all levels of FPN [26] by element-wise max or sum operations, which can improve the segmentation performance to a certain degree. Here, we also design the multiple-level fusion based ROI (fROI for short) module to simultaneously fuse those pyramid features from all multiple scales, original image and the pixel-level segmentation features.

As shown in Fig. 4. First of all, we utilize ROI pooling (align) [19, 13] operation to get pooled features of each proposal on pyramid features of all levels, semantic feature, as well as the original image that has been normalized. For each horizontal region proposal generated by sRPN, both the pooled features from pyramid network and the pixel-level guidance information of semantic segmentation are sent to one shared fully-connected (fc) layer, while features pooled from original image are processed by another fc layer to convert to the eigen-space the same as what are afore-mentioned (1024-channel). Then, we add up these features and utilize one extra fc layer to generate the final proposal-related features. At last, we predict objects' location and classification for both OBB and HBB tasks simultaneously via fc layers as the work [23] does using more distinguishing proposal-related features. The feature aggregation operation can be represented as Eqn. (1).

$$F_{box} = \sum_{k=2}^{5} fc_1(\rho(P_k, B)) + fc_1(\rho(F_{seg}, B)) + fc_2(\rho(I_{norm}, B)) \tag{1}$$

The $fc_1$ and $fc_2$ are two fc layers mentioned above. $P_k$ is the *k-th* level pyramid feature. $F_{seg}$ and $I_{norm}$ represent the semantic feature and normalized original image. $B$ stands for the horizontal region proposals set outputted by sRPN. $\rho$ means the ROI pooling (align) operation.

### 2.5. Loss Functions

As in Faster R-CNN [23], we minimize the multi-task loss:

$$\begin{aligned} L = {} & \lambda_1 \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \\ & + \lambda_2 \frac{1}{N_{reg-r}} \sum_i p_i^* L_{reg-r}(r_i, r_i^*) \\ & + \lambda_3 \frac{1}{N_{reg-h}} \sum_i p_i^* L_{reg-h}(h_i, h_i^*) \\ & + \lambda_4 \frac{1}{N_{seg}} \sum_i \sum_j L_{seg}(s_{i,j}, s_{i,j}^*) \end{aligned} \tag{2}$$

where, $p_i^*$ represents the GT label of objects, $p_i$ stands for the predicted probability of classification, $r_i^*$ and $h_i^*$ repre-

sent the coordinate vectors of GT for OBB and HBB tasks, $r_i$ and $h_i$ represent the predicted coordinate vectors, $s_{i,j}$ means the predicted mask score location and $s^*_{i,j}$ represents the GT labels of segmentation task. The first term $L_{cls}$ in Eqn. (2) is classification loss function which is cross entropy, $L_{reg-r}$ and $L_{reg-h}$ are location loss functions for OBB and HBB tasks respectively and each one is smooth L1 loss defined in [19]. For the segmentation-branch loss $L_{seg}$, we use the focal loss[27]. The hyper-parameters $\lambda_1$, $\lambda_2$, $\lambda_3$ and $\lambda_4$ are balance factors of four loss terms. We set $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1$ for all experiments in this paper.

In addition, for the coordinate vectors mentioned above, we use the following method to perform bounding box regression:

$$\begin{aligned} t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a \\ t_w &= \log(w/w_a), t_h = \log(h/h_a) \\ t_\theta &= \theta - \theta_a \end{aligned} \tag{3}$$

$$\begin{aligned} t^*_x &= (x^* - x_a)/w_a, t^*_y = (y^* - y_a)/h_a \\ t^*_w &= \log(w^*/w_a), t^*_h = \log(h^*/h_a) \\ t^*_\theta &= \theta^* - \theta_a \end{aligned} \tag{4}$$

We regress the center $(x, y)$ and the size $(h, w)$ of bounding box for both tasks, and one extra angle $\theta$ for OBB task, the definition of $\theta$ is the same as in OpenCV which is converted into radian with the range of $\left[-\frac{\pi}{2}, 0\right)$. In Eqn. (3) and (4), variables $x, x_a$ and $x^*$ are for the predicted box, anchor box and GT box respectively (likewise for $y, w, h, \theta$).

## 3. Experiments

In this section, we first introduce the datasets on which we evaluate our proposed object detection framework, then describe the implementation details of our proposed method, and finally show the achieved performance together with some analysis.

### 3.1. Datasets and Settings

To comprehensively evaluate the performance of our proposed detection framework, we conduct experiments on DOTA-v1.0 [2] and DOTA-v1.5 [3] (this challenge) [4], and two tasks named OBB and HBB are involved for testing the performance. The evaluation protocol for both tasks follows the PASCAL VOC benchmark [17], which uses mean Average Precision( mAP) as the primary metric. The IoU calculation on the OBB task takes the intersection over the union area of two polygons.

DOTA-v1.0 is the largest dataset for object detection in aerial images at present, it contains 2806 aerial images ranging in size from 800×800 to 4000×4000 pixels including objects of 15 categories with 188282 instances in total. It is split into training (1/2), validation (1/6) and testing (1/3) sets.

DOTA-v1.5, a upgraded version of DOTA-v1.0, has been employed for performance evaluation in *Detecting Objects in Aerial Images Challenge 2019* (DOAI2019), where the images are mainly collected from the Google Earth, satellite JL-1, and satellite GF-2 of the China Centre for Resources Satellite Data and Application. DOTA-v1.5 contains 0.4 million annotated object instances in total which are labeled into 16 categories. Comparing with DOTA-v1.0, the categories of DOTA-v1.5 are extended by adding the category of container crane. Moreover, many object instances in rather small scales, *e.g.* objects about or below 10 pixels, are additionally annotated in DOTA-v1.5, which makes the detection tasks much more challenging.

We employ the pretraining model ResNet101 [14] (as default) to initialize our network. We train the model for total 12 epochs with batch size 4 on 4 Tesla P40 GPUs (effective minibatch is 16). The learning rate is 0.02 and decrease it by 0.1 after 9 and 11 epochs. Also, the weight decay is 0.0001 and momentum is 0.9. For both training and testing, we split images into the blocks of 1024×1024 with the overlap of 512 pixels using the official development kit. We also employ the multi-scale for both training and testing, we firstly resize all images by 1.5× and 0.5× factors before splitting, then take these split subimages together with the original split ones as expanded training set. During testing, we use the same ratios to resize and split the images, and combine outputs using R-NMS [8] as the final results.

### 3.2. Comparisons with State-of-the-art Methods

We firstly compare our proposed approach on DOTA-v1.5 [4] dataset by the DOAI2019, the compared results of both HBB and OBB detection tasks have been shown in Table 1 and Table 2, where only the first five position scores are provided. The detailed results can be seen in https://captain-whu.github.io/DOAI2019/results.html. We (named "pca_lab") take the first and second places in the HBB and OBB tasks, respectively. We obtain 79.5% mAP of HBB task, which is 0.3% higher than the second place (*i.e.*, 79.5% *vs* USTC 79.2%), and 1.1% better than the third place (*i.e.*, 79.5% *vs* AICyber 78.4%). For example, the mAP performance of HBB task in several classes can significantly outperform other methods: 86.6% *vs* USTC 85.6% for "BD" category, 65.7% *vs* USTC 59.6% for "Brige" category, 84.1% *vs* AICyber 82.9% for "Harbor" category. We further evaluate our method in more challenging OBB task of DOTA-v1.5 dataset, and obtain the 76.6% mAP performance. When compared with all other four state-of-the-art methods, our performance of OBB task is lower 1.7% than the method on the first place, but we can significantly outperform other three methods, such as 0.9%

[2] http://captain.whu.edu.cn/DOTAweb/index.html
[3] https://captain-whu.github.io/DOAI2019/index.html

Table 1. Results comparison of HBB task on DOTA-v1.5 dataset (this challenge). Only the top five methods are provided here.

| Method | mAP(%) | Plane | BD | Bridge | GTF | SV | LV | Ship | TC | BC | ST | SBF | RA | Harbor | SP | HC | CC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ours | **79.5** | 88.3 | **86.6** | **65.7** | 79.8 | 74.6 | 79.4 | 88.1 | **90.9** | 85.4 | 84.2 | **73.9** | **77.4** | **84.1** | 81.1 | 76.1 | **57.1** |
| USTC | 79.2 | **89.3** | 85.6 | 59.6 | **80.9** | 75.2 | 81.1 | **89.6** | 90.8 | **85.9** | 85.7 | 69.5 | 76.3 | 81.7 | 81.8 | **76.5** | 57.1 |
| AICyber | 78.4 | 89.2 | 85.6 | 64.4 | 74.1 | **77.4** | **81.5** | 89.6 | 90.8 | 85.7 | **86.0** | 69.8 | 76.3 | 82.9 | **82.9** | 74.6 | 44 |
| wonderwall | 76.4 | 87.7 | 83.9 | 54.7 | 77.6 | 74.3 | 74.9 | 89.0 | 90.9 | 85.5 | 84.4 | 66.2 | 74.0 | 78.1 | 80.9 | 69.8 | 50.2 |
| czh | 76.2 | 88.0 | 85.0 | 64.4 | 73.5 | 72.7 | 80.3 | 88.4 | 90.8 | 85.4 | 83.6 | 62.9 | 70.0 | 81.1 | 80.6 | 74.1 | 39.1 |

Table 2. Results comparison of OBB task on DOTA-v1.5 dataset (this challenge). Only the top five methods are provided here.

| Method | mAP(%) | Plane | BD | Bridge | GTF | SV | LV | Ship | TC | BC | ST | SBF | RA | Harbor | SP | HC | CC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| USTC | **78.3** | **89.2** | 85.3 | 57.3 | **80.9** | **73.9** | **81.3** | **89.5** | 90.8 | 85.9 | **85.6** | 69.5 | 76.7 | 76.3 | **76.0** | **77.8** | **57.3** |
| Ours | 76.6 | 88.2 | **86.4** | **59.4** | 80.0 | 68.1 | 75.6 | 87.2 | **90.9** | 85.3 | 84.1 | **73.8** | **77.5** | 76.4 | 73.7 | 69.5 | 49.6 |
| czh | 75.7 | 89.0 | 83.2 | 54.5 | 73.8 | 72.6 | 80.3 | 89.3 | 90.8 | 84.4 | 85.0 | 68.7 | 75.3 | 74.2 | 74.4 | 73.4 | 42.1 |
| AICyber | 74.7 | 88.4 | 85.4 | 56.7 | 74.4 | 63.9 | 72.7 | 87.9 | 90.9 | **86.3** | 85.0 | 68.9 | 76.0 | 74.1 | 72.9 | 73.4 | 37.9 |
| zzzzzq | 73.5 | 82.1 | 84.9 | 56.0 | 80.0 | 66.1 | 78.1 | 87.8 | 90.8 | 83.1 | 84.2 | 64.8 | 73.7 | **77.5** | 72.4 | 77.4 | 18.0 |

Table 3. Ablation study of components.

| BaseLine | eFPN | sRPN | fROI | Ensemble | mAP@OBB(%) | mAP@HBB(%) |
|---|---|---|---|---|---|---|
| ✓ | – | – | – | – | 72.4 | 75.6 |
| ✓ | ✓ | | | | 72.8 | 75.9 |
| ✓ | ✓ | ✓ | | | 73.2 | 76.3 |
| ✓ | ✓ | | ✓ | | 73.8 | 77.2 |
| ✓ | ✓ | ✓ | ✓ | | 74.9 | 77.9 |
| ✓ | ✓ | ✓ | ✓ | ✓ | **76.6** | **79.5** |

Table 4. Computational complexity analysis.

| Method | Image | Memory (GB) | Training (s/iter) | Testing (FPS) |
|---|---|---|---|---|
| Faster R-CNN [26] | 1333×800 | 9.74 | 0.32 | 4.3 |
| Faster R-CNN | 1024×1024 | 9.50 | 0.29 | 4.5 |
| Baseline | 1024×1024 | 11.48 | 0.39 | 3.7 |
| Our method | 1024×1024 | 13.95 | 0.56 | 3.3 |

over "czh", 1.9% over "AICyber" and 3.1% over "zzzzzq" method. Moreover, for several categories of DOTA-v1.5 dataset, we can also achieve better performance than the method "USTC" on the first place, e.g., 86.4% *vs* 85.3% on "BD" category, 59.4% *vs* 57.3% on "Brige" category, 73.8% *vs* 69.5% on "SBF" category, 77.5% *vs* 76.7% on "RA" category. It demonstrates that the proposed object detection method can perform very well on both the HBB and OBB tasks in the RS imagery, and the proposed object-wise semantic representation method can boost the performance of the location regression and category recognition.

Besides, we also conduct experiments on the original DOTA-v1.0 [4] dataset with both OBB and HBB tasks, where the results are shown in Table 5 and Table 6. All results on DOTA-v1.0 are based on single model (ResNet101 [14] as backbone). For OBB task, our method gets 76.36% mAP which outperforms all published methods in Table 5. For HBB task, our method achieves the best performance among all published methods in Table 5 with 78.79% mAP. For all categories, our method obtains the best performance.

### 3.3. Ablation Analysis

To evaluate each module, we summarize some comparisons in Table 3, where eFPN, sRPN and fROI are respectively corresponded to the enhanced FPN, the semantic segmentation-guided RPN and the multiple-level fusion based ROI module. "ensemble" means that we combine the three results from ResNet101 [14], ResNeXt101 [24] and mdcn-ResNet101 [33], by employing these as the backbone network. Our baseline is Faster R-CNN [23] based on FPN [26] which is extended for rotational regression task.

All results are evaluated in DOTA-v1.5 dataset. From this table, we can have several observations.

**i) Enhanced feature pyramid network.** eFPN slightly improves the performance compared the baseline, for OBB and HBB tasks, we can see the increment are 0.4% and 0.3% respectively. Due to the baseline is already strong so the improvements are slight.

**ii) Semantic segmentation-guided RPN.** In comparison with the baseline, sRPN also has a relative improvements too, which may be attributed to the segmentation information. The relative increase are about 0.4% for both tasks.

**iii) Multiple-level fusion based ROI module.** The performance gain from fROI is relative large due to the integration of multiple-level features, which could compensate some information for objects with different scales, gaining about 1% for these two tasks.

**iv) Ensemble.** The ensemble of different networks can further the performance due to some certain compensation on feature information To a certain extent. In a word, all modules play some roles in boost the final performance of object detection. Besides, some visualization results of horizontal and oriented bounding box of our detector could be found in Fig. 5.

In addition, we also do provide the computational complexity analysis of our method, as shown in Table 4. For analysis, we train the models with batch size 4 on 2 Tesla P40 GPUs (effective minibatch is 8) with learning rate 0.01. In Table 4, the Faster R-CNN [23] is based on FPN [26], and is trained on COCO dataset [25] with the same learning rate strategy, the image is resized such that its shorter side has 800 pixels and 1333 for longer side. For comparison, we also conduct experiments on COCO dataset by resizing images to $1024 \times 1024$ pixels. According to these exper-

Table 5. Results comparison of OBB task on DOTA-v1.0 dataset

| Method | mAP(%) | Plane | BD | Bridge | GTF | SV | LV | Ship | TC | BC | ST | SBF | RA | Harbor | SP | HC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSD[28] | 10.59 | 39.83 | 9.09 | 0.64 | 13.18 | 0.26 | 0.39 | 1.11 | 16.24 | 27.57 | 9.23 | 27.16 | 9.09 | 3.03 | 1.05 | 1.01 |
| YOLOv2[9] | 21.39 | 39.57 | 20.29 | 36.58 | 23.42 | 8.85 | 2.09 | 4.82 | 44.34 | 38.25 | 34.65 | 16.02 | 37.62 | 47.23 | 25.50 | 7.45 |
| R-FCN[6] | 26.79 | 37.80 | 38.21 | 3.64 | 37.26 | 6.74 | 2.60 | 5.59 | 22.85 | 46.93 | 66.04 | 33.37 | 47.15 | 10.60 | 25.19 | 17.96 |
| FR-H[23] | 36.29 | 47.16 | 61.00 | 9.80 | 51.74 | 14.87 | 12.80 | 6.88 | 56.26 | 59.97 | 57.32 | 47.83 | 48.70 | 8.23 | 37.25 | 23.05 |
| FR-O[23] | 52.93 | 79.09 | 69.12 | 17.17 | 63.49 | 34.20 | 37.16 | 36.20 | 89.19 | 69.60 | 58.96 | 49.40 | 52.52 | 46.69 | 44.80 | 46.30 |
| Azimi *et al.*[21] | 68.20 | 81.40 | 74.30 | 47.70 | 70.30 | 64.90 | 67.80 | 70.00 | 90.80 | 79.10 | 78.20 | 53.60 | 62.90 | 67.00 | 64.20 | 50.20 |
| Ding *et al.*[7] | 69.56 | 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 |
| $R^2$CNN++[31] | 71.16 | 89.66 | 81.22 | 45.50 | 75.10 | 68.27 | 60.17 | 66.83 | **90.90** | 80.69 | 86.15 | 64.05 | 63.48 | 65.34 | 68.01 | **62.05** |
| Ours | **76.36** | **90.41** | **85.21** | **55.00** | **78.27** | **76.19** | 72.19 | 82.14 | 90.70 | **87.22** | **86.87** | **66.62** | **68.43** | **75.43** | **72.70** | 57.99 |

Table 6. Results comparison of HBB task on DOTA-v1.0 dataset

| Method | mAP(%) | Plane | BD | Bridge | GTF | SV | LV | Ship | TC | BC | ST | SBF | RA | Harbor | SP | HC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SSD[28] | 10.94 | 44.74 | 11.21 | 6.22 | 6.91 | 2.00 | 10.24 | 11.34 | 15.59 | 12.56 | 17.94 | 14.73 | 4.55 | 4.55 | 0.53 | 1.01 |
| YOLOv2[9] | 39.20 | 76.90 | 33.87 | 22.73 | 34.88 | 38.73 | 32.02 | 52.37 | 61.65 | 48.54 | 33.91 | 29.27 | 36.83 | 36.44 | 38.26 | 11.61 |
| R-FCN[6] | 47.24 | 79.33 | 44.26 | 36.58 | 53.53 | 39.38 | 34.15 | 47.29 | 45.66 | 47.74 | 65.84 | 37.92 | 44.23 | 47.23 | 50.64 | 34.90 |
| FR-H[23] | 60.46 | 80.32 | 77.55 | 32.86 | 68.13 | 53.66 | 52.49 | 50.04 | 90.41 | 75.05 | 59.59 | 57.00 | 49.81 | 61.69 | 56.46 | 41.85 |
| Azimi *et al.*[21] | 72.50 | 90.00 | 77.70 | 53.40 | 73.30 | 73.50 | 65.00 | 78.20 | 90.80 | 79.10 | 84.80 | 57.20 | 62.10 | 73.50 | 70.20 | 58.10 |
| $R^2$CNN++[31] | 75.35 | 90.18 | 81.88 | 55.30 | 73.29 | 72.09 | 77.65 | 78.06 | **90.91** | 82.44 | 86.39 | 64.53 | 63.45 | 75.77 | 78.21 | 60.11 |
| Ours | **78.79** | **90.41** | **85.77** | **61.94** | **78.18** | **77.00** | **79.94** | **84.03** | 90.88 | **87.30** | **86.92** | **67.78** | **68.76** | **82.10** | **80.44** | **60.43** |

Figure 5. Some detected examples on DOTA-v1.5 dataset of our method about both OBB and HBB tasks. The first row shows the results of HBB task while the second row corresponds to OBB task.

iments, we can see that our method would use about 20% more GPU memory than baseline. However, the final results suggest the cost is worth it.

## 4. Conclusion

In this paper, the segmentation guided object detection network was proposed to deal with the object detection tasks in remote sensing imagery. To learn multi-scale features describing objects in various scales, we revised FPN as the enhanced version to generate feature maps of multiple receptive fields. Moreover, considering to guide the rough horizontal region proposals with object-level context information, a semantic segmentation module was specifically designed to generate box-wise masks providing object-level guidance information for the RPN. Finally, a multiple-level fusion based ROI module was proposed for learning object-wise semantic representation base on previously obtained features. Extensive experiments on DOTA-v1.0 and DOTA-v1.5 datasets verified the effectiveness of our proposed object detection in remote sensing imagery.

**Acknowledgments**

This work was supported by the National Natural Science Foundation of China (Grant 61772276, 61602244, U1713208 and 61472187), Tencent AI Lab Rhino-Bird Focused Research Program (No. JR201922), the fundamental research funds for the central universities (No. 30918011321 and 30919011232), the 973 Program No.2014CB349303 and Program for Changjiang Scholars.

## References


[1] A.Krizhevsky, I.Sutskever, and G.Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1

[2] B.Li, J.Yan, W.Wu, Z.Zhu, and X.Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, pages 8971–8980, 2018. 1

[3] B.Li, W.Wu, Q.Wang, F.Zhang, J.Xing, and J.Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. *arXiv preprint arXiv:1812.11703*, 2018. 1

[4] G.Xia, X.Bai, J.Ding, Z.Zhu, S.Belongie, J.Luo, M.Datcu, M.Pelillo, and L.Zhang. Dota: A large-scale dataset for object detection in aerial images. In *CVPR*, pages 3974–3983, 2018. 3, 5, 6

[5] H.Law and J.Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018. 1

[6] J.Dai, Y.Li, K.He, and J.Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016. 1, 7

[7] J.Ding, N.Xue, Y.Long, G.Xia, and Q.Lu. Learning roi transformer for detecting oriented objects in aerial images. *arXiv preprint arXiv:1812.00155*, 2018. 2, 7

[8] J.Ma, W.Shao, H.Ye, L.Wang, H.Wang, Y.Zheng, and X.Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018. 2, 5

[9] J.Redmon and A.Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, pages 7263–7271, 2017. 1, 7

[10] J.Redmon and A.Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[11] J.Redmon, S.Divvala, R.Girshick, and A.Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1

[12] K.Chen, J.Pang, J.Wang, Y.Xiong, X.Li, S.Sun, W.Feng, Z.Liu, J.Shi, and W.Ouyang. Hybrid task cascade for instance segmentation. *arXiv preprint arXiv:1901.07518*, 2019. 1, 2, 3, 4

[13] K.He, G.Gkioxari, P.Dollár, and R.Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1, 2, 3, 4

[14] K.He, X.Zhang, S.Ren, and J.Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 5, 6

[15] L.Chen, A.Hermans, G.Papandreou, F.Schroff, P.Wang, and H.Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. In *CVPR*, pages 4013–4022, 2018. 2, 3

[16] L.Chen, G.Papandreou, F.Schroff, and H.Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 4

[17] M.Everingham, L.Van Gool, C.Williams, J.Winn, and A.Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 2, 5

[18] Q.Wang, L.Zhang, L.Bertinetto, W.Hu, and P.Torr. Fast online object tracking and segmentation: A unifying approach. *arXiv preprint arXiv:1812.05050*, 2018. 1

[19] R.Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 1, 4, 5

[20] R.Girshick, J.Donahue, T.Darrell, and J.Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1

[21] S.Azimi, E.Vig, R.Bahmanyar, M.Körner, and P.Reinartz. Towards multi-class object detection in unconstrained remote sensing imagery. *arXiv preprint arXiv:1807.02700*, 2018. 2, 7

[22] S.Liu, L.Qi, H.Qin, J.Shi, and Jiaya J.Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018. 4

[23] S.Ren, K.He, R.Girshick, and J.Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1, 2, 4, 6, 7

[24] S.Xie, R.Girshick, P.Dollár, Z.Tu, and K.He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 1, 6

[25] T.Lin, M.Maire, S.Belongie, J.Hays, P.Perona, D.Ramanan, P.Dollár, and C.Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 2, 6

[26] T.Lin, P.Dollár, R.Girshick, K.He, B.Hariharan, and S.Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 1, 2, 3, 4, 6

[27] T.Lin, P.Goyal, R.Girshick, K.He, and P.Dollár. Focal loss for dense object detection. In *CVPR*, pages 2980–2988, 2017. 1, 5

[28] W.Liu, D.Anguelov, D.Erhan, C.Szegedy, S.Reed, C.Fu, and A.Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 1, 7

[29] X.Wu, D.Hong, J.Tian, J.Chanussot, W.Li, and R.Tao. Orsim detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features. *IEEE Transactions on Geoscience and Remote Sensing*, 2019. 2

[30] X.Yang, H.Sun, K.Fu, J.Yang, X.Sun, M.Yan, and Z.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. 2

[31] X.Yang, K.Fu, H.Sun, J.Yang, Z.Guo, M.Yan, T.Zhan, and S.Xian. R2cnn++: Multi-dimensional attention based rotation invariant detector with robust anchor strategy. *arXiv preprint arXiv:1811.07126*, 2018. 2, 7

[32] X.Zhou, D.Wang, and P.Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1

[33] X.Zhu, H.Hu, S.Lin, and J.Dai. Deformable convnets v2: More deformable, better results. *arXiv preprint arXiv:1811.11168*, 2018. 3, 6

[34] Y.Jiang, X.Zhu, X.Wang, S.Yang, W.Li, H.Wang, P.Fu, and Z.Luo. R2cnn: Rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017. 2

[35] Z.Cai and N.Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 1

[36] Z.Zhang, W.Guo, S.Zhu, and W.Yu. Toward arbitrary-oriented ship detection with rotated region proposal and discrimination networks. *IEEE Geoscience and Remote Sensing Letters*, (99):1–5, 2018. 2